Business process as a service - BPaaS : securing data and services

Mohamed El Mehdi Bentounsi

Sorbonne Paris Cité - Université Paris Descartes

U.F.R. Mathématiques et Informatique

Rapport de thèse pour l'obtention du grade de

Docteur de l'Université Paris Descartes

Spécialité Informatique

par

BENTOUNSI Mohamed El Mehdi

École Doctorale :	EDITE - Informatique, Télécommunication et Électronique de Paris
Laboratoire :	LIPADE - Laboratoire d'Informatique Paris Descartes
Équipe de Recherche :	diNo - Systèmes Orientés Données Intensives et Connaissances

Les Processus Métiers en tant que Services - BPaaS : Sécurisation des Données et des Services

Soutenue le 14 Septembre 2015

devant le jury composé de :

BENBERNOU Salima	Professeur, LIPADE, Univ. Paris Descartes	Directrice
GODART Claude	Professeur, LORIA, Univ. de Lorraine	Rapporteur
GRIGORI Daniela	Professeur, LAMSADE, Univ. Paris Dauphine	Présidente
HACID Mohand-Said	Professeur, LIRIS, Univ. Lyon 1	Rapporteur
HAINS Gaétan	Professeur, LACL, UPEC et Huawei Technologies	Examinateur
DEME Cheikh Sadibou	PDG, SOMONE France	Invité

Sorbonne Paris Cité - Université Paris Descartes

Department Of Mathematics and Computer Science

Thesis submitted to obtain the degree of

Doctor of Université Paris Descartes

Option Computer Science

by

BENTOUNSI Mohamed El Mehdi

Doctoral School: EDITE - Informatique, Télécommunication et Électronique de Paris
Laboratory: LIPADE - Laboratoire d'Informatique Paris Descartes
Research Team: diNo - Data Intensive and Knowledge Oriented System

Business Process as a Service - BPaaS : Securing Data and Services

presented and defended on September 14, 2015

Thesis Committee :

BENBERNOU Salima	Professor, LIPADE, Univ. Paris Descartes	Supervisor
GODART Claude	Professor, LORIA, Univ. de Lorraine	Reviewer
GRIGORI Daniela	Professor, LAMSADE, Univ. Paris Dauphine	President
HACID Mohand-Said	Professor, LIRIS, Univ. Lyon 1	Reviewer
HAINS Gaétan	Professor, LACL, UPEC et Huawei Technologies	Examiner
DEME Cheikh Sadibou	CEO, SOMONE France	Invited

"If I maintain my silence about my secret it is my prisoner . . . if I let it slip from my tongue, I am ITS prisoner."

Arthur Schopenhauer (1788-1860).

UNIVERSITÉ PARIS DESCARTES

Abstract

Department of Mathematics and Computer Science

Doctor of Philosophy

Business Process as a Service - BPaaS :
Securing Data and Services

by Bentounsi Mohamed El Mehdi

Cloud computing has become one of the fastest growing segments of the IT industry. In such open distributed computing environments, security is of paramount concern. This thesis aims at developing protocols and techniques for private and reliable outsourcing of design and compute-intensive tasks on cloud computing infrastructures. The thesis enables clients with limited processing capabilities to use the dynamic, cost-effective and powerful cloud computing resources, while having guarantees that their confidential data and services, and the results of their computations, will not be compromised by untrusted cloud service providers.

The thesis contributes to the general area of cloud computing security by working in three directions. First, the design by selection is a new capability that permits the design of business processes by reusing some fragments in the cloud. For this purpose, we propose an anonymization-based protocol to secure the design of business processes by hiding the provenance of reused fragments. Second, we study two different cases of fragments' sharing : biometric authentication and complex event processing. For this purpose, we propose techniques where the client would only do work which is linear in the size of its inputs, and the cloud bears all of the super-linear computational burden. Moreover, the cloud computational burden would have the same time complexity as the best known solution to the problem being outsourced. This prevents achieving secure outsourcing by placing a huge additional overhead on the cloud servers.

This thesis has been carried out in Université Paris Descartes (LIPADE - diNo research group) and in collaboration with SOMONE under a Cifre contract. The convergence of the research fields of those teams led to the development of this manuscrit.

Keywords : cloud computing ; security and privacy by design ; business processes ; process reuse ; biometric ; IT monitoring.

Acknowledgements

Contents

List of Figures

List of Tables

Abbreviations

AES	**A**dvanced **E**ncryption **S**tandard
ANRT	**A**ssociation **N**ationale de la **R**echerche et de la **T**echnologie
API	**A**pplication **P**rogramming **I**nterface
ARPA	**A**dvanced **R**esearch **P**rojects **A**gency
ASP	**A**pplication **S**ervice **P**rovider
ATM	**A**utomated **T**eller **M**achine
BP	**B**usiness **P**rocess
BPaaS	**B**usiness **P**rocess **a**s **a** **S**ervice
BPEL	**B**usiness **P**rocess **E**xecution **L**anguage
BPM	**B**usiness **P**rocess **M**anagement
BPMN	**B**usiness **P**rocess **M**odel and **N**otation
BPO	**B**usiness **P**rocess **O**utsourcing
CAPEX	**CAP**ital **EX**penditures
CBC	**C**ipher **B**lock **C**haining
CEO	**C**hief **E**xecutive **O**fficer
CEP	**C**omplex **E**vent **P**rocessing
CERN	**C**onseil **E**uropéen pour la **R**echerche **N**ucléaire
CFB	**C**ipher **F**eed**B**ack
CGT	**C**ombinatorial **G**roup **T**esting
CIFRE	**C**onventions **I**ndustrielles de **F**ormation par la **RE**cherche
CIO	**C**hief **I**nformation **O**fficer
CMOS	**C**omplementary **M**etal **O**xide **S**emiconductor

CTR	**C**oun**T**e**R**
DaaS	**D**atabase **a**s **a** **S**ervice
DBMS	**D**ata**B**ase **M**anagement **S**ystem
DDL	**D**ata**D**efinition **L**anguage
DES	**D**ata **E**ncryption **S**tandard
DGH	**D**omain **G**eneralization **H**ierarchy
DLP	**D**iscrete **L**ogarithm **P**roblem
DML	**D**ata**M**anipulation **L**anguage
ECB	**E**lectronic **C**ode **B**ook
EMS	**E**vent **M**anagement **S**oftwares
EMaaS	**E**vent **M**anagement **a**s **a** **S**ervice
FHE	**F**ully **H**omomorphic **E**ncryption
FPGA	**F**ield-**P**rogrammable **G**ate **A**rray
HaaS	**H**ardware **a**s **a** **S**ervice
HTML	**H**yper**T**ext **M**arkup **L**anguage
HTTP	**H**yper**T**ext **T**ransfer **P**rotocol
HTTPS	**HTTP** over **SSL**
IaaS	**I**nfrastructure **a**s **a** **S**ervice
ICT	**I**nformation and **C**ommunication **T**echnologies
IDS	**I**ntrusion and **D**etection **S**ystems
IFC	**I**nformation **F**low **C**ontrol
IP	**I**nternet **P**rotocol
IPS	**I**ntrusion and **P**revention **S**ystems
IT	**I**nformation **T**echnology
ITU	**I**nternational **T**elecommunications **U**nion
KDD	**K**nowledge **D**iscovery in **D**atabases
MoDem	**Mo**dulator **Dem**odulator
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NPRP	**N**ational **P**riorities **R**esearch **P**rogram
OFB	**O**utput **F**eed**B**ack
OPEX	**OP**erating **EX**pense
OrBAC	**Or**ganization-**B**ased **A**ccess **C**ontrol
PaaS	**P**latform **a**s **a** **S**ervice

PBA	**P**rocess **B**ased **A**pplication
PC	**P**ersonal **C**omputer
PF	**P**rocess **F**ragment
PGT	**P**robabilistic **G**roup **T**esting
PHE	**P**artially **H**omomorphic **E**ncryption
PIN	**P**ersonal **I**dentification **N**umber
PPDP	**P**rivacy-**P**reserving **D**ata **M**ining
PPDP	**P**rivacy-**P**reserving **D**ata **P**ublishing
PPTA	**P**robabilistic **P**olynomial-**T**ime **A**lgorithm
QI	**Q**uasi-**I**dentifier
QoS	**Q**uality **o**f **S**ervice
RDF	**R**esource **D**escription **F**ramework
RIA	**R**ich **I**nternet **A**pplications
RSA	**R**ivest **S**hamir **A**dleman
SaaS	**S**oftware **a**s **a** **S**ervice
SBA	**S**ervice-**B**ased **A**pplication
SCN	**S**ervice **C**omposition **N**etwork
SCSI	**S**oftware and **C**omputing **S**ervices **I**ndustry
SLA	**S**ervice **L**evel **A**greement
SMC	**S**ecure **M**ultiparty **C**omputation
SME	**S**mall **M**edium-sized **E**ntreprises
SOA	**S**ervice **O**riented **A**rchitectures
SOC	**S**ervice **O**riented **C**omputing
SQL	**S**tructured **Q**uery **L**anguage
SSH	**S**ecure **SH**ell
SSL	**S**ecure **S**ockets **L**ayer
SSN	**S**ocial **S**ecurity **N**umber
TCO	**T**otal **C**ost **O**wnership
TCP	**T**ransmission **C**ontrol **P**rotocol
TPS	**T**ransaction **P**rocessing **S**ystem
UI	**U**ser **I**nterface
VGH	**V**alue **G**eneralization **H**ierarchy
VM	**V**irtual **M**achine

VMM	**V**irtual **M**achine **M**onitor
VPN	**V**irtual **P**rivate **N**etwork
W3C	**W**orld **W**ide **W**eb **C**onsortium
WAN	**W**ide **A**rea **N**etwork
WWW	**W**orld **W**ide **W**eb

To my family, for their patience and
their gratifying enthusiasm...

Résumé

Malgré les avantages économiques de l'informatique en nuage (ou cloud computing) pour les entreprises et ses multiples applications envisagées, il subsiste encore des obstacles pour son adoption à grande échelle. La sécurité des données sauvegardées et traitées dans le nuage arrive en tête des préoccupations des décideurs des directions des systèmes d'information. De ce fait, l'objectif principal de nos travaux de recherche lors de cette thèse de doctorat est de poser des bases solides pour une utilisation sûre et sécurisée du nuage.

Dans un premier lieu, l'externalisation des processus métiers vers le nuage permet aux entreprises de réduire les couts d'investissement et de maitriser les coûts d'exploitation de leurs systèmes d'information ; Elle permet aussi de promouvoir la réutilisation des parties (ou fragments) de ses processus métiers en tant que service cloud, éventuellement par des concurrents directs, afin de faciliter le développement de nouvelles applications orientés services 'SOA', ainsi la collaboration à l'échelle du nuage. Néanmoins, le fait de révéler la provenance d'un fragment réutilisé est considérée comme une brèche dans la vie privée et risque d'être dommageable pour l'entreprise propriétaire de ce fragment. Les techniques d'anonymisation des données ont fait leurs preuves dans le domaine des bases de données. Notre principale contribution dans cette partie est la proposition d'un protocole basée sur l'anonymisation des fragments de processus métiers afin de garantir à la fois, la vie privée de leurs propriétaires et la disponibilité de ces fragments pouvant être réutilisés dans le nuage.

Les systèmes d'authentification biométriques permettent une authentification des individus avec une garantit suffisante. Néanmoins, le besoin en ressources informatiques

'calcul et stockage' de ces systèmes et le manque de compétences au sein des organismes freinent considérablement leurs utilisations à grande échelle. Le nuage offre la possibilité d'externaliser à la fois le calcul et le stockage des données biométriques à moindre cout et de proposer une authentification biométrique en tant que service. Aussi, l'élasticité du nuage permet de répondre aux pics des demandes d'authentifications aux heures de pointes. Cependant, des problèmes de sécurité et de confidentialité des données biométriques sensibles se posent, et par conséquent doivent être traité afin de convaincre les institutions et organismes à utiliser des fragments externes d'authentification biometriques dans leurs processus métiers. Notre principale contribution dans cette partie est un protocole léger 'coté client' pour une externalisation (sur un server distant) de la comparaison des données biométriques sans révéler des informations qui faciliteraient une usurpation d'identité par des adversaires. Le protocole utilise une cryptographie légère basée sur des algorithmes de hachage et la méthode de 'groupe de tests combinatoires', permettant une comparaison approximative entre deux données biometriques.

Dans la dernière partie, nous avons proposé un protocole sécurisé permettant la mutualisation d'un Hyperviseur (Outil permettant la corrélation et a gestion des événements issus du SI) hébergé dans le nuage entre plusieurs utilisateurs. La solution proposée utilise à la fois, le chiffrement homomorphique et la réécriture de règles de corrélation afin de garantir la confidentialité les évènements provenant des SI des différents utilisateurs.

Cette thèse a été réalisée à l'Université Paris Descartes (groupe de recherche diNo du LIPADE) avec le soutien de la société SOMONE et l'ANRT dans le cadre d'une convention CIFRE.

Mots clés : informatique en nuage ; sécurité et vie privée ; processus métiers ; réutilisation de processus ; biométrie ; supervision et hypervision informatique.

Intoduction

Preliminaries

HORIZON 2020[1] has included *"protecting freedom and security of Europe and its citizens"* among the seven societal challenges, that reflect the policy priorities of the Europe 2020 strategy and address major concerns shared by citizens in Europe and elsewhere.

Nowadays, with the emergence of computers and especially the Internet, *digital security* is considered as one of the major challenges to the implementation of human rights, e.g., recent controversial debate about PRISM[2] and *right to be forgotten* opposing European Commission to Google Inc. On digital security, this challenge focuses on the improvement of the cyber security ; and ensuring privacy and freedom, including in the Internet, and also enhancing the societal legal and ethical understanding of all areas of security, risk and management according to :

- Article 17 of the International Covenant on Civil and Political Rights, *"No one shall be subjected to arbitrary or unlawful interference with his privacy, family, home or correspondence, nor to unlawful attacks on his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks."*
- Article 8 of the European Convention on Human Rights, *"Right to respect for private and family life : Everyone has the right to respect for his private and*

[1] The European Union Framework Programme for Research and Innovation.

[2] PRISM is a clandestine surveillance program revealed by Edward Snowden. PRISM was launched by United States National Security Agency (NSA) in 2007 to collect internet communications of foreign nationals from major US internet companies.

family life, his home and his correspondence. There shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic wellbeing of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others."

- Article 1 of the French Data Protection Act of January 6th, 1978, and amended in 2004, *"ICT should serve every citizen. Its development must take place within the framework of international cooperation. It must not restrict either human identity, humain rights, humain privacy, individual and public freedoms."*

In such context where, on the one hand, we are witnessing a rapid expanding of digital innovations due to the advent of technologies such as : *Cloud computing*, *social network*, *big data*, *smartphones*, and *Internet of Things* ...etc ; and on the other hand, the will of citizens and governments to retain control over data collected, and also services used through Internet that we have started our reflection [Ben10].

A part of this Ph.D. thesis is a collaborative project (CIFRE[3]) between the diNo[4] research group (Université Paris Descartes), which is interested in data and knowledge management research issues, and SOMONE[5] company, specialized in developing and integrating IT monitoring and event management softwares. The project aims at developing a protocol for secure outsourcing of event management softwares in the cloud.

After the beginning of the thesis, I had the opportunity to participate in two international research projects :

- *"The European Network of Excellence in Software Services and Systems (S-Cube)*[6]*"* comprising several European partners. The network aims at enabling Europe to lead the software services revolution and helping shape the software-service based Internet which is the backbone of our future interactive society, and
- *"Trusted Computation-Intensive Services in Cloud Computing Environments"*, a NPRP[7] project led by Prof. Qutaibah Malluhi (Qatar University) and Prof.

[3]CIFRE 1169-2010 : industriel research contract from 01-01-2011 to 31-12-2013, between Université Paris Descartes and SOMONE, and supported by the French Association Nationale de la Recherche et de la Technologie (ANRT).

[4]Data Intensive and Knowledge Oriented Systems (Laboratoire d'Informatique Paris Descartes).

[5]SOMONE is a French SME specialized in IT monitoring and event management softwares. It was founded in 2006 by Cheikh Sadibou Deme. SOMONE develops the TeeM Software Suite and E-Control.

[6]S-Cube

[7]National Priorities Research Program with a Qatar Foundation Grant.

Mikhail J. Atallah (Purdue University). The project aims at developing techniques and tools for private and reliable outsourcing of compute-intensive tasks on cloud computing infrastructures.

The manuscript summarizes all the work done during my thesis and the results obtained.

Problem Statements

Cloud computing is revolutionizing the computer world by allowing the outsourcing of IT infrastructure to specialized providers, similar to the way companies outsource the production of electricity to power utilities. The key driving forces behind cloud computing are the ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software. The benefits of cloud computing include pay-per-use, reduced power consumption, server consolidation, and more efficient resource utilization. Hence, cloud-service clients will be able to add more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers will increase utilization via multiplexing, and allow for larger investments.

Information security is currently one of the most important issues in information systems, especially with the successful adoption of cloud computing. Security criteria most commonly used are *confidentiality*, i.e., assurance that information is accessible only for authorized persons or organizations, *integrity*, i.e., assurance that the information is authentic and complete, and *availability*, i.e., assurance that the information is accessible when needed, by those who need them [ITU91, Sta10]. Thereby, the principle of *Privacy and Security by Design* should be introduced in the context of cloud computing where privacy and information security should be integrated at the design stage of ICT. More precisely, we investigated security issues in the following contexts :

Design by Selection. The cloud model gives the opportunity for organizations to compose and re-use cloud services from a variety of cloud providers to create what's known as a cloud syndication [YZB11, Pap12, ZZYB13]. Cloud syndications at the SaaS level are termed Business Process as a Service (BPaaS), which, according to business analysts, is the next step forward in the evolution of cloud computing [Bit11].

In a practical manner, BPaaS enables to reuse parts of processes (called process fragments) for the development of new process-based application (PBA) at lower costs, also known as *"design by selection"* [ASKW11]. Therefore, a cloud provider

may decompose business processes to make them more manageable ; and later permits the selection, composition and sharing of process fragments in order to design new process-based application by a third party.

In the literature, several works have adressed business process management (BPM) [BEKM05, BEKM06, BEMP07, MD08, BMS10], configurable process modeling and clones detection [RDtHM11, DDvD+11, DGRU13], business process decomposition and identification [KL06, ICH10], service selection, composition and reuse [NBCT06, RFG10, YB12, ZZYB13, HTTA14] ; but they have not integrated the privacy and information security at the design stage of their approaches. In fact, security has always been seen as an independent layer. Therefore, the security of business processes (i.e., confidentiality of business secrets and availability of applications) was not taken into consideration in the context of design by selection.

Biometric Authentication. When considering business process outsourcing in the cloud, security aspects are regarded as the most critical factors by IT directors. Because companies' digital assets are taken from an intraorganizational to an interorganizational context where cloud providers control the lifecycle management of business processes [BGJ+13]. Thereby, cloud providers assume the responsibility for ensuring the confidentiality, integrity and availability of data and services offered to companies [AFG+09]. For this purpose, they implement a set of security services such as : database duplication, authentication, control access, intrusion detection, software patches, OS updates ... etc.

Conventional password-based authentication is not suitable for use at a large scale in the cloud. Additionally, unlike other security services, authentication is a responsibility shared by the cloud provider and end-users. Consequently, cloud providers should implement new *authentication techniques as a service* that reduce both the risks of users' mistakes and impersonate users. A satisfying solution is to use biometrics-based authentication. Therefore, the security of biometric systems and biometrics data should be taken into account when implementing biometric authentication in the cloud.

Business Process Monitoring in the cloud. Companies have developed Event Management Softwares (EMS) to monitor IT infrastructure and business processes which became critical (e.g., Tivoli Netcool/OMNIbus[8] of IBM, Openview of HP, BMC Event Manager[9] of BMC and interscope of CA). These integrated tools support business and IT users and directors in managing process execution quality by providing several features, such as analysis, prediction, monitoring, control, and

[8] Tivoli Netcool/OMNIbus.
[9] BMC Event Manager.

optimization [GCC+04, Mal11, MHD12]. However, the main obstacle to the broad adoption of such systems remains a high-CAPEX[10] and OPEX[11].

In such context, SOMONE plans to propose an EMS as *Event Management as a Service* (EMaaS) shared between several small and medium-sized entreprises (SME) to *(i)* reduce CAPEX and OPEX, and *(ii)* generalize the use of such EMS. However, transferring and treating IT events in the cloud can be considered, by IT directors, as a breach of security. Indeed, IT events often contain sensitive data about IT infrastructure of companies like : IP adresses, host names, alerts ... etc. To this end, a secure protocol should be implemented to ensure the confidentiality and integrity of IT events in the cloud.

Research Issues

As noticed above, using BPaaS raises various security issues. In particular, we are interested in the following key issues :

1. There are several security risk issues when reusing process fragments in the BPaaS delivery model. First, *how to ensure the end-to-end availability of process-based applications ?*. Existing secure process composition mechanisms assume a fully trusted process provider, which is not always true, and focus on announced Service-Level Agreement (SLA) availability rates of process fragments. However, in reality, a process provider may suspend the outsourcing of a given service including process fragment. Consequently, all business processes that re-use this process fragment will be impacted and abnormalities on their executions will occur.

   A second key problem in outsourcing is that the hosting, the execution and the re-use of process fragments are considered as sensitive that may contain business secrets or provide personal information (e.g., SSN). Consequently, fragments composition may expose process providers' business activities, as well as process consumers and their end-users, to confidentiality issues. Thereby, an adversary may be able to :

   (a) Reveal sensitive information about the process provider activities, such as details of how certain process fragments are composed or the list of process fragments provided by an organization;

[10] CAPEX for Capital expenditures are used by a company to acquire or upgrade physical assets such as equipment, property, or industrial buildings.

[11] An operating expense or OPEX is an ongoing cost for running a product, business, or system.

(b) Infer connections between end-users and a process provider by analyzing intermediate data, like input/output values produced by a process fragment, thus obtain and/or modify confidential and sensitive information by using SQL injection attacks [WMK06].

Both are considered to be unacceptable breaches of confidentiality.

2. The main feature of biometric data is the recognition of persons with a very high probability. Thus, biometric data are considered as personal and private information. Accordingly, biometric systems which manipulate such data in the cloud must integrate efficient security mechanisms to avoid persons impersonating. In addition, we note that biometrics are approximately stable over the time. Solutions exist in the literature to secure remote biometric authentication such as : homomorphic encryption [BG11, YSK+13], biometric cryptosystems [JS02, DKK+12] , biohashing [GL03, JL05, BCRA13] and feature transformation [RCB01, JLKC06]. These solutions are either considerably secure or practical in performance but not both at once. We aim to propose a secure and efficient protocol to permit the use of weak devices in remote biometric authentication in the cloud.

3. Event management, also known as complex event processing (CEP), needs to centralize, at a central point in the cloud, the scattered data in different points of the distributed IT infrastructure such as : servers, hubs, databases, ... etc. IT events are then stored in a remote relational database and correlations between them discovered through standard SQL queries and triggers. The anonymization permits publishing and querying data in a secure manner [Sam01]. However, one must ask about the completeness and accuracy of data due to attributes generalization and tuples suppression in anonymized datasets. On the other hand, encryption aims to modify data mathematically, in order to secure data transfer and storage in the cloud while ensuring accuracy and completeness. However, querying encrypted data remains impractical. Our goal in this project is to provide a protocol for a secure querying of anonymized datasets while ensuring data accuracy and completeness.

Contributions

The thesis contributes to the general area of cloud computing security. We have studied in depth the security issues discussed above, and the main research contributions in this dissertation focus on the following :

1. We study the emergence of the BPaaS delivery model and discuss some research issues.

2. We emphasize cloud computing security towards a *survey*.

3. We formalize the reuse of process fragments in the cloud, and introduced the notion of anonymous process fragments for privacy-preserving business activities of organizations [BBA12, BBDA12].

4. We enrich the proposed approach with a notion of diverse view to guarantee the end-to-end availability of PBAs. Then, to validate the effectiveness and evaluate the performance of the proposed protocol, we applied it to the QWS datasets [AM07, AM08], and studied the impact on the quality of the BPaaS views [BBA16].

5. We give an *overview* of techniques in the literature to secure remote biometric authentication in the cloud.

6. We propose a nonadaptative combinatorial group testing based protocol to permit a secure, approximative, and computationally non demanding remote biometric authentication. A prototype is implemented and its performances discussed.

7. We introduce an encryption-based anonymization approach to secure multi-party complex event processing. The proposed approach is implemented in the context of IT Event Management as a Service [BD15].

Dissertation Structure

This dissertation is structured as follows : In Chapter 1 , we show how successive evolutions of computer systems and the fact that businesses are increasingly embracing the Internet, logically lead to cloud computing and BPaaS, and then introduce some basic concepts related to cloud computing and design by selection. Chapter 2 emphasizes computer security towards a survey and, give an overview on security mechanisms used in privacy and security by design. In Chapter 3, we present the first security issue regarding the design by selection concept. Then, we discuss the solution based on anonymity and diversity of process fragments. Finally, we describe the implementation of our approach and discuss the performance. Chapter 4 focuses on how secure remote biometric authentication system in the cloud. For this purpose, we give an overview of solutions discussed in the literature. Then, we discuss our solution based on combinatorial group testing. In Chapter 5, we present the last security issue regarding the event management

as a service. Then, we discuss the encryption-based anonymization approach. Finally, we provide concluding remarks and discuss directions for future research.

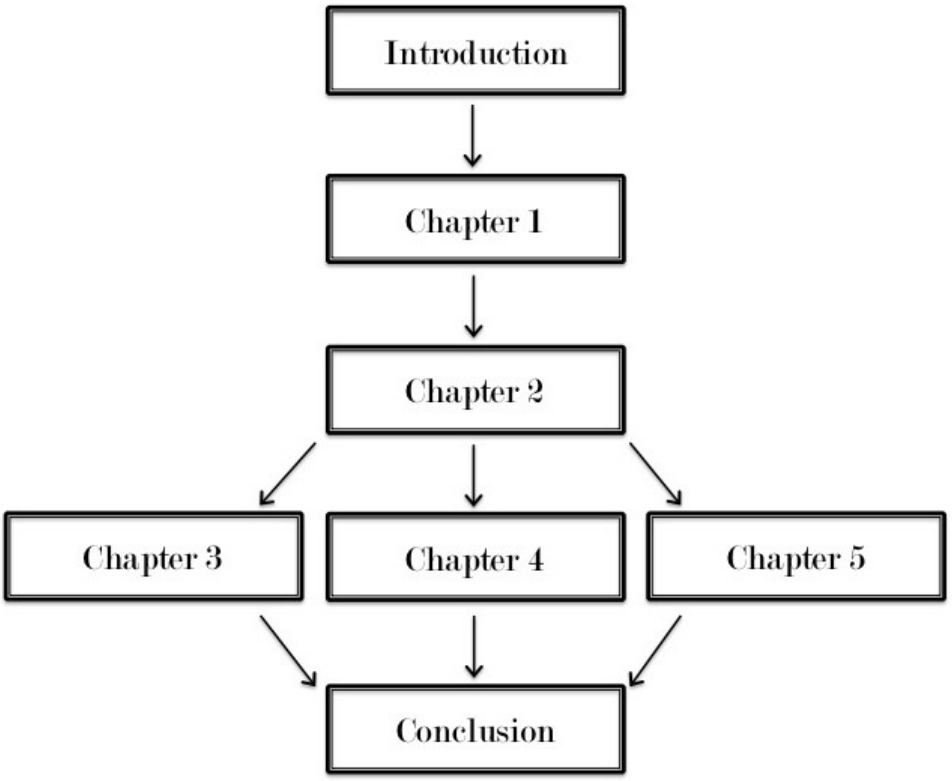

FIGURE 1: Reading Guide.

Figure 1 gives the reading guide of the thesis manuscript. An arrow from one chapter to another indicates that reading the first is necessary to understand the second.

1

Cloud Computing

1.1 Introduction

In this chapter, we present the emergence of cloud computing as a logical step in computer's history. We show how successive evolutions of computer systems and the fact that businesses are increasingly embracing the Internet, logically lead to cloud computing. We also show how in the current context of reducing costs and using mobile devices, cloud computing remains an ideal solution for companies. Finally, we enumerate the cloud computing benefits on users, and principal obstacles to its large adoption.

The remainder of the chapter is structured as follows : We first present IT evolution in Section 1.2. Section 1.3 introduces the key concepts of cloud computing, and the different cloud delivery models. Then, we show the consistency of cloud computing with the trend being followed by companies to outsource their IT resources, business processes, and design new processes by selection. Security risks of cloud computing are also presented and the concept of security and privacy by design introduced permitting to reach a high level of security and privacy. In Section 1.4, we discuss related work on business process outsourcing and present Business Process as a Service as the next major category of IT. Section 1.5 concludes the chapter.

1.2 Towards cloud computing

1.2.1 At the root of cloud computing

To get a complete grasp on cloud computing paradigm, it would be interesting to know where we are today and how we got here. Cloud computing is considered as an evolution of IT with a rich family tree. Indeed, since the emergence of mainframes and the rise of IT in 1960s, computer architecture follows a regular cycle of centralization/decentralization. In this tree, *mainframes* constitute the epitome of centralization and control, because of the centralization of computational logic and data persistence in a *single big* machine.

In the late 1950s, most mainframes had no explicitly interactive interface. They operated in batch mode and accepted sets of punched cards and magnetic/paper tapes to transfer data and programs. By the early 1970s, many mainframes acquired interactive user interfaces based on keyboard/display devices, which did not contain user data, and operated as timesharing computers. This new generation could support hundreds of users

simultaneously along with batch processing. The infrastructure requirements of mainframes were drastically reduced during the mid-1990s when CMOS[1] designs replaced the older bipolar technology.

Mainframes were characterized by a high-CAPEX coupled with a fanatical running. This gave birth in 1980s to *personal computers (PCs)* that each company was able to acquire despite a limited budget, and thus can be seen as ending the tyranny of mainframes [Win11].

PCs allowed the use of commercial programs to process data or perform particular jobs. Such an *autonomous system* had the advantage of allowing the full realization of a job on a *single small* machine without involving other connected systems. Thereby, PCs had served as a launching pad for the software industry ; and with the continued growth of this industry, the cost of IT has dropped drastically. However, on the one hand softwares have brought a powerful automation to anyone having a PC and, on the other hand, companies have developped more and more softwares without consideration for the best practices. Thereby, softwares combined to PCs have posed problems for companies in many areas, especially due to the data persistence issues on PCs and the poor security of softwares.

The *transaction processing systems*, or TPS, has been set up to meet the need of interaction with the same database for a growing number of users. In a TPS model, a single server, generally a mainframe, handles computations and data storages, while client machines are responsible of inputs and outputs. Initially, airline reservation systems[2] had exploited this model where clients have no local storage, and the connection with the server was done by *dedicated networks*.

Comparable to TPS, *client/server architectures* appeared in the early 1990s in order to give a solution to the problem of data persistence in PCs. The innovative idea of the client/server architectures was to split treatment between a server and a PC, which became able to execute some parts of business processes. In most cases, the principal role of servers was to centralize data and manage parts of treatment, while the clients handled the user interface. However, this situation has evolved somewhat rapidly, and PCs allowed to perform important calculations locally in order to improve performances and increase functionalities. Often clients and servers communicate, through a specific software layer called *middleware*, over *computer networks* on separate hardwares or over a *WAN*.

[1] CMOS for Complementary metal–oxide–semiconductor is a technology for constructing integrated circuits.

[2] The first TPS was the American Airlines SABRE system which has been developed in 1953 to automate the way American Airlines booked reservations.

The client/server architecture has been massively used in information systems, and showed its limits due to the lack of any standardized exchange protocol, which made more difficult the flow management. In addition, the non-standardization of front-end clients (hardwares, OS versions) has confronted CIOs and IT Directors with the delicate issue of deployment on user workstations.

1.2.2 The emergence of the Internet and the Web

While the users were forced to interact with computers through punched cards or connected terminals, they experienced a high degree of autonomy by using modems[3], then *Internet* ; and more recently, through broadband networks and wireless. Historically the word Internet was used in 1883 as a verb to refer to interconnected motions. In 1969, the Advanced Research Projects Agency (ARPA) connected the computer systems of Stanford Research Institute, UCLA, UC Santa Barbara, and the University of Utah together, across the United States, in a small network called ARPANET [ACKM04].

ARPANET allowed the connection of *autonomous systems*, which gave rise to the first standards organizations for governing computer networks. By the early 1970s, the term was used as a shorthand form of the technical term *internetwork*, the result of interconnecting computer networks with special gateways or routers.

If the Internet has brought quiet and relatively slow revolution, the *Web* has been a seismic revolution. In the mid-1990s, Web architectures have led to the *re-centralization* of computational logic and data persistence on central servers, bringing the PC to a simple display device through the Web browser. Web architectures allowed the use of applications on the scale of the Internet based on hypertext technology as HTTP[4] and HTML[5] standards. Additionally, they have allowed access to applications without going through software deployment phase on each PC.

Tim Berners-Lee and his team at CERN[6] are credited with inventing the original HTTP along with HTML and the associated technology for a web server and a text-based web browser [Ber88]. In 1989, they proposed the *"WorldWideWeb"* project now known as the *World Wide Web* [BCGP92]. Their initial idea was to create an online encyclopedia. For that, they designed a principle of pages with data sheets, linked by hyperlinks. Later,

[3] A modem (modulator-demodulator) is a device that modulates signals to encode digital information and demodulates signals to decode the transmitted information

[4] HTTP (Hypertext Transfer Protocol) was designed to support hypertext, or the ability to interconnect documents by inserting links between them as part of the document contents.

[5] HTML (HyperText Markup Language) defines a standard set of special textual indicators (markups) that specify how a Web page's words and images should be displayed by the Web browser.

[6] CERN The European Organization for Nuclear Research is a European research organization whose purpose is to operate the world's largest particle physics laboratory

further development was taken over by the World Wide Web Consortium (W3C) with the goal of promoting standards for the Web. When it became more popular and a global platform, the Web was taken up by businesses in order to broadcast commercial wafers at a lower cost. In the late 1990s, the websites became transactional, allowing the emergence of electronic commerce, and have turned into veritable IT applications [Plo09].

1.2.2.1 Application Service Providers

Since the advent of the Internet and Web technologies, the concept of *Application Service Providers* (ASPs) emerged. Indeed, Start'up creators and companies in the software and computing services industry (SCSI) saw great potential for web architectures, and considered a new outsourcing model in the form of ASPs. According to the ASP Industry Consortium, *"An ASP manages and delivers application capabilities to multiple entities from data centers across a wide area network (WAN)"* [Cur00]. As result, SCSI provided to companies a pay-as-you-go pricing model for a variety of applications and business processes hosted in datacenters. This business model allowed them regular incomes through a subscription system. In addition, ASPs have allowed companies to get rid the operating process coupled to a low-CAPEX when integrating new applications.

There are two types of ASP-based applications :

HTTP based Applications. Despite their advantages, HTTP based Applications are subject to a number of limitations. First, complex applications often require that users navigate through a series of Web pages to complete a single job. So, it is very frustrating and confusing to access an application through a HTTP based Web interface. Therefore, HTTP Web interfaces are very limited in terms of capacity of interaction and often provide a *simple* navigation according to a predetermined scenario. This mode of interaction is very limiting for an application frequently used, and for which we would like to have a good productivity [Plo09]. Second, ordinary HTTP does not encrypt data before sending them. If adversaries were to use a network sniffer to intercept messages between clients and a remote HTTP server, they would be able to *read* those messages. A Secure Sockets Layer (SSL) was developed by Netscape to protect data transferred over TCP/IP protocol. HTTPS, also knows as HTTP over SSL, allows the Web server and client to use SSL to authenticate to each other and establish an encrypted connection between themselves [Sto02].

Client/server based Applications. The second alternative to provide ASP based applications is the client/server mode. This mode is much more satisfying in terms of interactivity and ergonomics. However, it needs a deployment phase on

user workstations, which goes against the promise of ASPs to provide hosted applications. Therefore, companies were faced the same issues of software integration of internal applications. In addition, firewalls block outside middleware traffic, which makes the deployment more complicated [Plo09, ACKM04].

The interface issues were the main reason for the failure of the ASPs. In more technical terms, ASP based applications often used an :

- Unique application.
- Unique version of the application.
- Unique database.
- Unique authentication system.

An ASP-based application may be shared by a set of users belonging to different companies. This fact can produce a high volume of data, which is difficult to manage using a single database. Also, it would be interesting to separate authentication systems and data from different companies in order to prevent that an adversary may access to data belonging to a third company.

Furthermore, companies may desire customizing an application to integrate the specificity of their businesses. And the fact to provide a monolithic application can be a locking point for companies to adopt ASP-based applications. Finally, companies may desire keeping the current version of the application, and do not upgrade or integrate new features offered by the ASP providers. Therefore, it is necessary to coexist a multiple versions of the same application.

1.2.2.2 Rich Internet Applications

Traditional Web applications have been extended in several directions to meet the need of new functionalities in Web applications, such as high level of interactivity and effective integration of audio and video. In 2003, Macromedia[7] has introduced several server technologies that enabled advanced user interactivity with shared and dynamic data across networked systems. These technologies, called *Rich Internet Applications (RIAs)*, allow Web designers and developers to create a new breed of Web applications that can connect multiple users simultaneously in live audio, video and text environments.

[7]Macromedia is the software company responsible for the success of the near-ubiquitous Flash Player. The company was acquired in 2005 per its rival Adobe Systems.

According to Macromedia [Duh03], *"RIAs combine the best user interface functionality of desktop software applications with the broad reach and low-cost deployment of Web applications and the best of interactive, multimedia communication. The end result: an application providing a more intuitive, responsive, and effective user experience"*. It means that users are now capable to perform computations, use audio and video in a tightly integrated manner, send and receive data in the background asynchronously from the user's requests, and so forth, independently of the remote server it is connected to. A RIA normally runs inside a Web browser and does not require software installation on the client side to work. Therefore, when using a RIA :

- An interface is deployed on the client side.
- The interface communicates with online services through HTTP. RIA runs as a client/server application, where the client is the RIA interface. During the use of the Web application, the RIA interface remains in the Web browser and disappears when closing the browser.

For all that, RIA can be considered as a resumption of client / server architecture. Or rather, it had ended the choice between Web based application and client/server based application. Indeed, RIA technology has provided a purely Web solution without the delicate issue of software integration, while benefiting a decentralized client / server architecture. However, the major drawback is that RIA does not manage offline mode. Consequently, if the Web browser is closed by mistake, all data will be lost. This issue is now being addressed and Four solutions are given [Plo09] :

1. Stay connected with the spread of wireless networks.
2. Use an extension of Web browser that manages the offline mode (eg., google gears).
3. Use a new generation of Web browser that manages the offline mode.
4. Use a synchronization software (eg., Windows Live Mesh).

Finally and despite the last negative point raised, RIA definitely played a fundamental role in the emergence of cloud computing.

1.2.2.3 Web 2.0

According to Webopedia[8], the *Web 2.0* is defined as a *marketing term given to describe a second generation of the WWW that is focused on the ability for users to collaborate*

[8] Webopedia, the online technical dictionary

and share information online. Web 2.0 basically refers to the transition from static HTML based Web pages to a more dynamic Web applications with new components as Web services, blogs, and wikis...

Web services are considered as the most important component of the Web technology. They have appeared to meet the specific needs of *integration* of several autonomous and heterogeneous information systems, and *automation* of business processes spanning across these systems. Therefore, Web services are the way to expose the functionality of an information system and make it available through standard Web technologies. The use of standard technologies reduces heterogeneity, and in the same time, is the key to facilitating application integration. Furthermore, Web services naturally enable new computing paradigms and architectures. They are specifically geared toward service-oriented computing (SOC) and service oriented architectures (SOA) [ACKM04].

A Web service is seen as an application accessible to other applications over the Web, and is described through its functional (i.e. what it does) and non-functional properties (i.e. the way it is supplied). Non-functional properties of a system include all those which are not directly related to the provided functionality such as quality of service (QoS) as well as cost and adherence to standards and obligations on the user/provider [Bov08, TRF+07].

SOC/SOA propose abstractions, frameworks, and standards to facilitate integrated access to heterogeneous applications and resources, encapsulated in Web services. They allow service compositions through Application Programming Interface (API) in order to master complexity, where complex services are incrementally built out of services at a lower abstraction level. A composite Web service (or composite service for short) can be seen as an umbrella that brings together a set of components to fulfill a complex task (e.g., office tasks, travel, intelligent information gathering, analysis, etc). A composite Web service is itself a Web service and can be accessed using the same protocols [ACKM04].

1.2.3 Virtualization

The *virtualization* means to create a virtual version of a resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments. Virtualization allows applications and users to interact with the virtual resource as if it were a real single logical resource. Tanaka *et al.* [TYI88] are the first who introduced the term of virtualization in the field of databases to provide users with multiple views of single database. They described

the concept, and implementation techniques of schema virtualization in object-oriented databases.

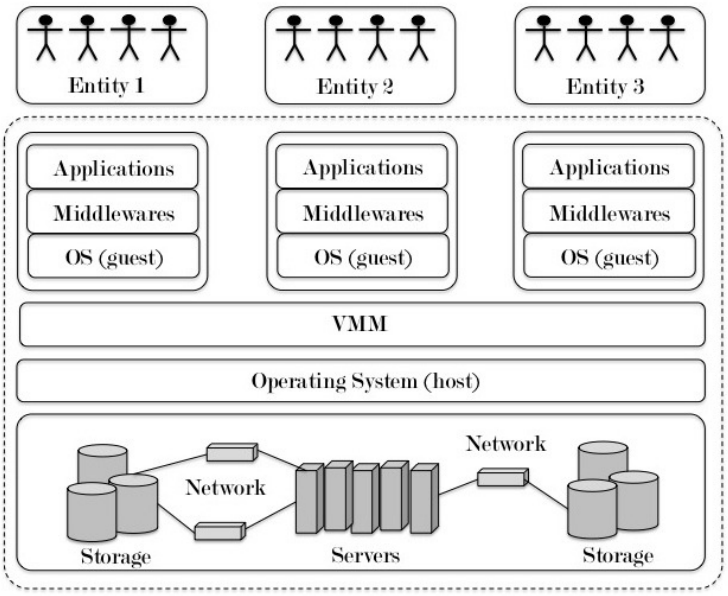

FIGURE 1.1: IT virtualization infrastructure.

Figure 1.1 depicts the architecture for an IT virtualization infrastructure. A virtual machine (VM) is a simple environment that emulates a computer system, generally an operating system, which is created inside another environment. The term *guest VM* refers to the virtual machine, while the environment which hosts virtual machines is called the *host*. A host machine may dynamically create and take into account a set of guests VM on demand. A virtual machine monitor (VMM) or hypervisor intermediates between the host and the guest VM. By isolating individual guest VMs from each other, the VMM enables a host to support multiple guests running different operating systems.

The Virtualization has been a great success with businesses because [Bit11] :

- Enterprises have usually started virtualization as a consolidation effort. Indeed, the focus tended to be on reducing CAPEX (server and hardware), reducing energy costs and perhaps avoiding or delaying a data center build-out or move.
- Entreprises have needed operational improvements, flexibility, speed and managing downtime more efficiently. For this purpose, VMs enabled a foundation that can be used for basic automation tools, rapid provisioning and cloning, server reprovisioning, and rapid restart.
- Once processes were in place to enable broad automation, the enterprises were ready to look at introducing self-service offerings based on the virtualization architecture.

1.2.4 Summary of IT evolution

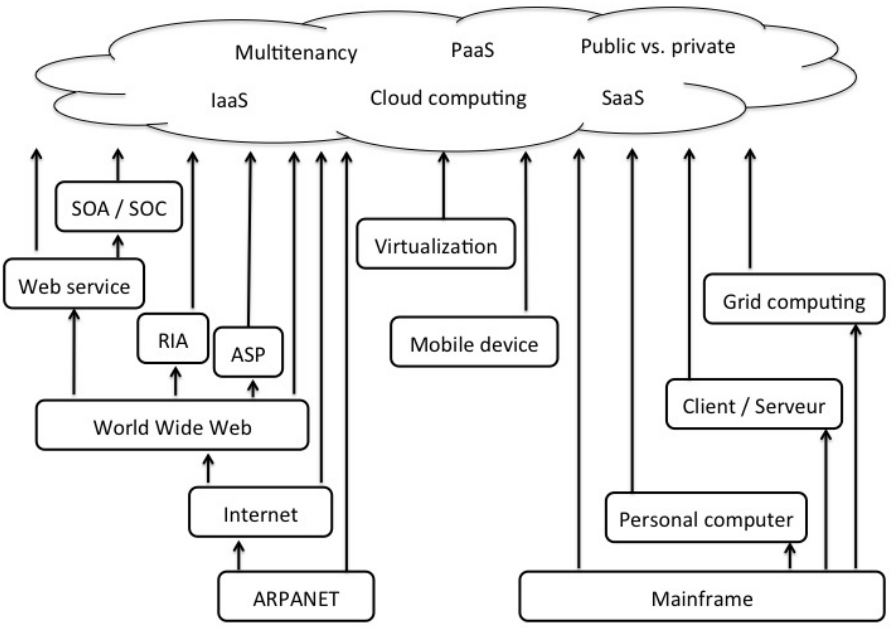

FIGURE 1.2: Summary of IT evolution - Cloud computing family tree.

By offering the hosting of IT applications on platforms available through the Web, cloud computing is a result of all evolutions discussed in the first part of this chapter and summarized in Figure 1.2.

From the computer architecture point of view, and with the rise of mobile devices, the cycle of centralization/ decentralisation started with the mainfraimes seems to be finished due to the need of hosted applications, making inescapable centralized architectures. IT interfaces evolution seems also to be finished with the apparition of RIAs. Indeed, RIA technologies resolve the principal issue of ASP based application, which is the software deployment on user workstations.

Besides,Virtualization has learned from the failure of the HTTP/ASP based applications, especially due to shared resouces, and offered a more suitable architecture for hosted applications. Finally, Cloud computing has integrated the best practice of Web 2.0 such as service mashup and composition through API. In addition, we should note that Web 2.0 has prepared users and businesses to use hosted applications.

1.3 Cloud Computing

One vision of 21st century computing is that users will access Internet services over lightweight portable devices rather than through some descendant of the traditional

desktop PC. Because users won't have (or be interested in) powerful machines, who will supply the computing power ? The answer to this question lies with *cloud computing*.

Cloud computing is a recent trending in IT that moves computing and data away from desktop and PCs into large data centers. The first to give prominence to this term (and maybe to coin it) was Google's CEO Eric Schmidt, in late 2006. It refers to applications delivered as services over the Internet as well as to the actual cloud infrastructure, namely, the hardware and systems software in data centers that provide these services [AFG+09]. The key driving forces behind cloud computing is the ubiquity of broad band and wireless networking, falling storage costs, and progressive improvements in Internet computing software.

Cloud-service clients are able to add more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers will increase utilization via multiplexing, and allow for larger investments. It is facilitating access to an elastic (meaning the available resource pool can expand or contract over time) set of resources, cloud computing has demonstrable applicability to a wide-range of problems in several domains.

1.3.1 Context, social and economic issues

It would be the economic crisis last years, which really put cloud computing on the agenda. In fact, in today's IT world, companies supplying services over the Internet typically need to over provision their servers by as much as a 500 percent to handle peak loads. However, over-provisioning is expensive not only in terms of CAPEX and the cost of the housing of the physical equipment, but also in terms of cooling and supplying electricity mainly to the idle spare machines (OPEX). In fact, it has been estimated that data centers consume 1%-2% of the world's electricity, and this percentage is rapidly growing.

Cloud computing mixes aspects of *grid computing* (,. . . hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [Fos02]), *Internet computing* (,. . . a computing platform geographically distributed across the Internet [MRK+03]), *utility computing* (,. . . a collection of technologies and business practices that enables computing to be delivered seamlessly and reliably across multiple computers, . . . available as needed and billed according to usage, much like water and electricity are today [RW04]), *autonomic computing* (,. . . computing systems that can manage themselves given high-level objectives from administrators [KC03]), *edge computing* (,. . . provides a generic template facility for any type of application to spread its execution across a dedicated grid, balancing the

load ... [DPW04]) and *green computing* (a new frontier of *ethical* computing [Fos05]) starting from the assumption that in next future energy costs will be related to the environment pollution). To the following list we also add *trust computing*, in order to highlight the necessity of mechanisms and techniques for addressing trust and security issues.

The development and the success of cloud computing is due to the maturity reached by both hardware and software virtualization technologies. These factors made realistic the Leonard Kleinrock outlook of computing as the fifth utility, like gas, water, electricity and telephone [Kle05]. In commercial contexts, among the others we highlight :

Amazon Elastic Compute Cloud (Amazon EC2)[9] is a Web service that provides resizable computing capacity in the cloud. In the 2014 Cloud Infrastructure as a Service Magic Quadrant, Gartner placed Amazon Web Services in the *leaders* quadrant and rated AWS as having the furthest completeness of vision and highest ability to execute [LTG+14].

Amazon Relational Database Service (Amazon RDS)[10] is a Web service that makes it easy to set up, operate, and scale a relational database in the cloud.

Google App Engine (App Engine)[11] is a fully-managed Platform as s Service using built-in services to run applications and business processes.

Salesforce.com [12] provides a complete customer relationship management (CRM) technology solution for different areas of companies, starting with sales and extending to other customer-facing areas like marketing and customer service.

Microsoft Azure [13] proposes solutions for Websites hosting, virtual machines, managed relational databases, and cloud-based machine learning and predictive analytics. Recently, Azure Marketplace permits to users to search and deploy thousands of solutions to simplify development and management of applications.

There are also several scientific open activities and projects such as :

RESERVOIR project. RESERVOIR[14] is an open source technologies based Framework that enables the delivery of better services for businesses and eGovernment with energy-efficiency and elasticity by increasing or lowering compute based on demand.

[9] Amazon EC2
[10] Amazon RDS
[11] App Engine
[12] Salesforce.com
[13] Microsoft Azure
[14] RESERVOIR

Future Grid project. The FutureGrid Project[15] supports several clouds, distributed among five sites, in aggregate providing the capacity of over a thousand cores. The FutureGrid clouds are configured with Nimbus, OpenStack and Eucalyptus all of which support interfaces that are roughly compatible with AWS EC2/S3, allowing users to move between clouds relatively easily.

OpenNebula project. OpenNebula[16] provides a simple but feature-rich and flexible solution for the comprehensive management of virtualized data centers to enable private, public and hybrid IaaS clouds. Users use OpenNebula to manage data center virtualization, consolidate servers, and integrate existing IT assets for computing, storage, and networking. They also use OpenNebula to provide a multi-tenant, cloud-like provisioning layer on top of an existing infrastructure management solution (like VMware vCenter).

Nimbus project. Nimbus[17] is an open-source toolkit focused on providing Infrastructure-as-a-Service (IaaS) capabilities to the scientific community.

All of them support and provide an on-demand computing paradigm, in the sense that a user submits a request to the cloud that remotely, and in a distributed fashion, processes them and gives back the results.

1.3.2 Cloud computing ontology

In transitional phase towards cloud computing, new categories of IT services were being created for all kinds of applications, databases and services, providing storage, backups, data replication, data protection, security, etc ; and various classifications of IT cloud services were given. Aymerich *et al.* [AFS08] presented Software as a Service (SaaS), Hardware as a Service (HaaS), Database as a Service (DaaS) and Platform as a Service (PaaS) as the main categories of cloud computing.

As depicted in Figure 1.3, Youseff *et al.* [YBS08] proposed an ontology of cloud computing which demonstrates a dissection of the cloud into Five main layers, with three constituents to the cloud infrastructure layer, and illustrated their inter-relations as well as their inter-dependency on preceding technologies. In order to define the ontology of cloud computing, they opted to use composability as a methodology. Indeed, composability enables the proposed ontology to capture the inter-relations between the different cloud components. They borrowed this method from the principle of composibility in

[15] FutureGrid
[16] OpenNebula
[17] Nimbus

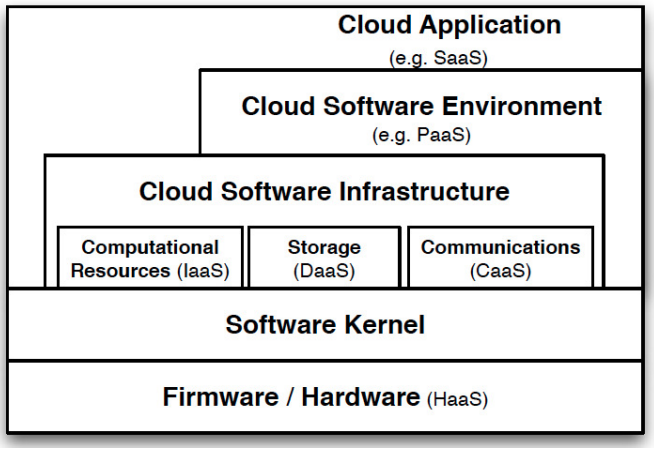

FIGURE 1.3: Cloud computing ontology according to Youseff *et al.* [YBS08].

SOA, they used it here in a limited fashion to refer to the ability to compose one cloud service from one or more other cloud services.

Most research works proposed an ontology consisting of three layers analogous to the technical layers in most cloud realizations : infrastructure, platform as a service, and application [WABS09, ALMS09, YZB11, Pap12, FSG+14]. In the following, we present in detail the different layers of cloud computing and their main vendors.

1.3.2.1 Infrastructures in the cloud

Infrastructure as a service (IaaS) is a type of cloud computing service. IaaS is defined as a standardized and highly automated offering, where locations and hardware infrastructure, complemented by storage and networking capabilities, are owned by a service provider. IaaS is offered as self-service interfaces, including a Web-based UI (User Interface) and an API, to the user on demand [LTG+14], and provides basic security, including perimeter defenses, such as firewalls, Intrusion Prevention Systems (IPS), and Intrusion Detection Systems (IDS) [FSG+14].

Users have to make some decisions regarding the installation of software such as operating system, platform middleware and application. These decisions should comprise security considerations such as blocking out attackers by locking ports, patching the operating system, running an anti-virus software, etc., as well as configuration and enforcement of access control policies [ALMS09].

IaaS constitutes the largest segment of cloud computing market (the broader IaaS market also includes cloud storage and cloud printing). The resources are scalable and elastic in near real time, and metered by use. As shown in Figure 1.4, Weinhardt *et al.* [WABS09] distinguished between two categories of infrastructure business models.

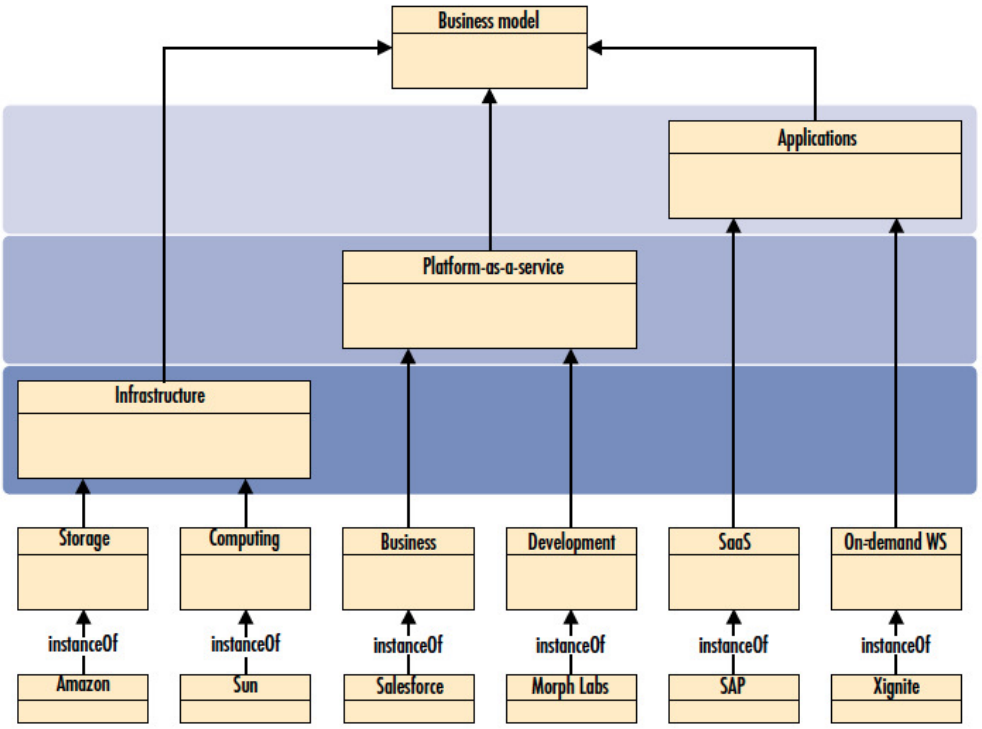

FIGURE 1.4: Cloud computing architecture according to Weinhardt *et al.* [WABS09]. Note that the components' location is significant. Those further to the top facilitate encapsulated functionality from the layers beneath by aggregating and extending service components via composition and mashup technologies.

- Infrastructure supplying computing power as Amazon EC2, and
- Infrastructure providing storage capabilities as Amazon Simple Storage Service (Amazon S3[18]).

Generally, cloud providers organize cloud computing infrastructures in a cluster-like structure to facilitate virtualization technologies. The resources may be single-tenant or multitenant, and hosted by the service provider or on-premises in the user's data center. In [MD11], Mazzuco et Dumas examined the problem of managing a server farm to maximize the revenue earned by cloud providers.

The Magic Quadrant [LTG+14] evaluated IaaS solutions that are delivered in an entirely standardized fashion specifically, public cloud, along with private cloud that uses the same or a highly similar platform. For that, they took into account a set of measuring points to describe each provider :

- Locations: Data center locations by country, languages that the IaaS provider does business in and technical support.

[18] Amazon S3 is used with a simple web services interface to store and retrieve any amount of data from anywhere on the web

- Computing, storage, network and security notes: Notes on the offering, including any missing core functionality or significant features.
- Other notes: including important missing capabilities. They note other cloud-related services, such as cloud storage, and their availability.

FIGURE 1.5: Magic Quadrant [LTG+14] for Cloud Infrastructure as a Service.

Amazon.com and Microsoft confirm that they are leaders of IaaS vendors thanks to their ambitious road map. Indeed, they serve a broad range of use cases, although they do not excel in all areas. In the same time, we have niche players, like Dimension Data, that may be excellent providers for the use cases in which they specialize, but may not serve a broad range of use cases well, or have a broadly ambitious road map. In this Magic Quadrant, there are no challengers, or well-positioned vendors to serve some current market needs. However, there are visionaries, like Google, that have an ambitious vision of the future, and are making significant investments in the development of unique technologies (Figure 1.5).

1.3.2.2 Platforms in the cloud

Platform as a service (PaaS) is a service model which allows customers to build their own applications by delivering services in the form of program development tools. In contrast with the IaaS deployment model, PaaS providers host hardware, operating system and platform middleware such as Business Process Execution Language (BPEL) engines and Database Management Systems (DBMS). PaaS is usually offered as virtual servers (virtualization) on a single physical server.

As depicted in Figure 1.4, the platform layer represents solutions on top of a cloud infrastructure that provide value-added services from both a technical and a business perspective. Weinhardt *et al.* [WABS09] distinguished between development and business platforms.

- Development platforms let developers write their source code and upload it into the cloud where the applications are then served by the upper model. In this case, developers don't have to worry about issues such as system scalability when application usage grows ; and the expenses are considerably lowered to companies, since they do not need to manage the hardware and software required to build applications. For instance, Google App Engine[19] features Software Development Kits (SDKs) for programming in Python, Java, PHP and Go.

- Business platforms such as SalesForce.com, which is a cloud platform that lets companies build and deliver custom apps faster to connect employees, customers, and products. SalesForce.com is named a leader in the Magic Quadrant [NPI+15] for Entreprise Application Platform as a Service (Figure 1.6).

FIGURE 1.6: Magic Quadrant [NPI+15] for Entreprise Application Platform as a Service.

According to the platform delivery model, Merino *et al.* [RVC+12] distinguished two categories of platform providers. In the first category, cloud platforms share the same resources between the users such as an instance of DBMS . Therefore, an effective control access mechanism should be set up to guarantee the security. In the second category,

[19] Google App Engine

providers do not share resources, providing instead pre-packaged VM with the software stack the customer demands. Note that VMs' isolation is not sufficient to guarantee the security.

1.3.2.3 Applications in the cloud

Software as a Service (SaaS) offers us complete and pre-designed softwares, where the users access with authentication protocols and use applications, maintained by providers, via the Internet. It is what most people recognize in cloud computing because it represents the customer's actual interface.

As depicted in Figure 1.4, SaaS represents the top model of cloud solutions. It improves operational efficiency and also reduces costs to customers by streamlining applications maintenance and support to providers. Weinhardt *et al.* [WABS09] distinguished between Web application and Web service. Google Docs is the most prominent example of SaaS. It proposes a broad catalogue of Microsoft Office applications such as Word and Excel as well as easy-to-use email and calendar applications that are entirely accessible through a Web browser or smartphone application.

SaaS is seen as being the showcase of cloud computing. As result, all cloud-based applications will be accessed as SaaS. Therefore, the security risks encompasses all risks regarding lower levels, more risks related to the method of accessing the application, and the terminal used.

1.3.3 Cloud deployment models

Due to the great diversity of cloud computing solutions, customers should take a look to the different cloud deployment models and analyze their advantages, disadvantages, and constraints in terms of security, scalability, elasticity, pricing, and migration.

1.3.3.1 Public vs. Private clouds

When a cloud is made available in a pay-as-you-go manner to the public, we call it a *public cloud* ; and the service being sold is *utility computing*. A public cloud is offered as a service, usually over an Internet connection. Current examples of public utility computing include Amazon Web Services, Google App Engine, and Microsoft Azure.

We use the term *private cloud* to refer to internal datacenters of a business or other organization that are not made available to the public, i.e., behind a firewall. [AFG+09].

Usually, companies use private clouds through Virtual Private Network (VPN) to share a single datacenter among several entities.

Finally, a hybrid cloud environment consisting of multiple internal and/or external providers, and will be typical for most enterprises.

1.3.3.2 Multi-tenants vs. multi-users

Basically, multi-tenants architectures were introduced for databases shared between several tenants. A major consequence of resource sharing is that the performance of one tenant can be adversely affected by resource demands of other colocated tenants [NMS+15]. At first, what is a tenant? *"A tenant is the organizational entity which rents a multi-tenant SaaS solution. Typically, a tenant groups a number of users, which are the stakeholders in the organization."* [BZ10]

With the advent of cloud computing, a new concept of multi-tenancy which refers to resources and applications appeared. Indeed, A multi-tenants application lets tenants share the same hardware resources, by offering them one shared application and database instance, while allowing them to configure the application to fit their needs as if it runs on a dedicated environment.

Multi-tenancy is an architectural pattern in wich a single instance of the software is run on the service provider's infrastructure, and multiple tenants access the same instance. In contrast to the multi-users model, multi-tenancy requires customizing the single instance according to the multifaceted requirements of many tenants. The multi-tenants model also contrasts with the multi-instances model in which each tenant gets his own instance of the application.

1.3.4 The Challenge of Security in the Cloud

The benefits of cloud computing include pay-per-use, reduced power consumption, server consolidation, and more efficient resource utilization. Hence, cloud-service clients will be able to add more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers will increase utilization via multiplexing, and allow for larger investments [BBDA12].

Consequently, tenants' digital assets are taken from an intraorganizational to an interorganizational context. This creates a number of issues, among which security aspects are regarded as the most critical factors when considering cloud computing adoption [BGJ+13]. Armbrust *et al.* [AFG+09] defined a list of three technical obstacles to

the adoption of cloud computing : availability of service, data lock-in and data confidentiality. In the same line, Vouk [Vou08] specified user's security as a research and engineering challenge in future. The principal goal in clouds is then to reach a high level of privacy and data security which allows companies to outsource not only non-strategic applications that also strategic ones.

Additionally, legislation and compliance frameworks raise further challenges on the outsourcing of data, applications, and processes. The high privacy standards in the European Union, e.g., and their legal variations between the continent's countries give rise to specific technical and organizational challenges. For instance, Article 25 and 26 of the EU data protection Directive prohibit transfers of personal data to countries outside of European Economic Area, unless these countries have an adequate level of data protection [20095].

Privacy by design is an approach to systems engineering which takes privacy into account throughout the whole engineering process [Lan01]. Privacy by design can perfectly be used in the context of cloud computing, because it enables a formal definition of security risks, and the design of end-to-end security solutions. Privacy by design is based on several security mechanisms that we will see in detail in the reminder of this manuscript

A great interest on cloud computing security has been manifested from both academic and private research centers, and numerous projects in database and the service community handled the personal data. Some of them are dealing with identity management, and exploit access control method for policy compliance : PRIME[20], PRIMELife[21], SERENITY[22], DISCREET[23], PRiMMA[24], SCALUS[25], and WEBAPPSEC[26].

1.4 Business Process Outsourcing

A *business process* is a collection of related, structured activities or tasks that produce a specific service or product for customers [SF03]. Cloud computing model gives the opportunity to *mashup* and *compose* data and services from a variety of cloud providers to create what's known as a cloud syndication. Cloud syndications are essentially federations of cloud providers whose services are aggregated in a single pool [Pap12]. As depicted in Figure 1.7, cloud syndications at the SaaS level are termed *Business Process as a Service* (BPaaS). It allows creating unique end-to-end business processes that

[20] Privacy for Identity Management in Europe
[21] Privacy and Identity Management in Europe for Life
[22] System Engineering for Security and Dependability
[23] Discreet Service Provision in Smart environment
[24] Privacy Rights Management for Mobile Applications
[25] SCALing by means of Ubiquitous Storage
[26] Web Application Security Consortium

are usually syndicated with other external services (possibly provided by diverse XaaS providers).

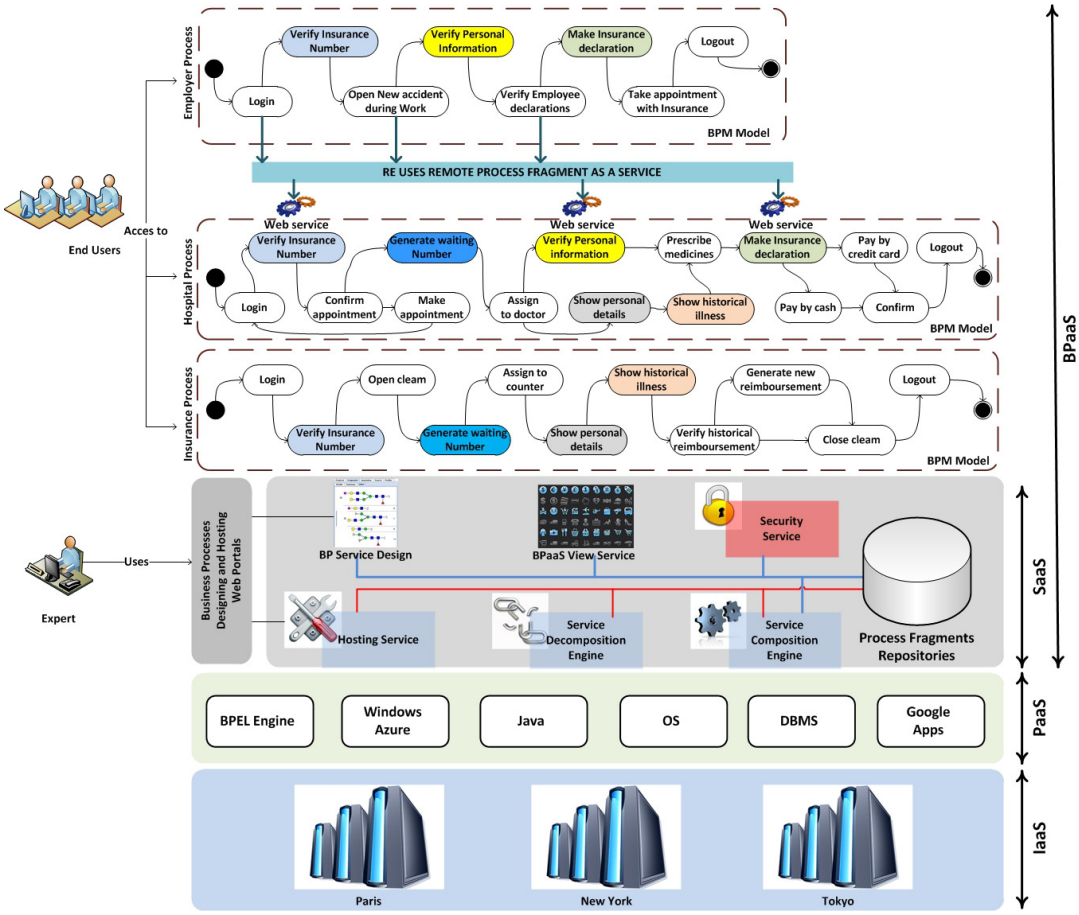

FIGURE 1.7: Multi-tenant BPaaS Platform [BBDA12].

BPaaS is emerging as the next major category of cloud IT. By 2015, 50 percent of new Business Process Outsourcing (BPO) deals will be delivered as BPaaS (i.e., they will be significantly cloud enabled) [McN10]. Forrester research study predicated that BPaaS will grow from 0.8 billion dollars in 2012 to 10 billion dollars in 2020 [RKM10].

We give, in the following, a brief overview of related research works on business process outsourcing in the cloud, in order to overcome the obstacles to greater adoption of BPaaS.

1.4.1 Business Process Management and Modeling

Business Process Management (BPM) aims to *(i)* identify internal business processes of an organization, *(ii)* design new process models, and *(iii)* be able to manage and optimize business process execution by monitoring and reengineering. BPM lifecycle is

an iterative process in which all the BPM aspects are covered. It consists of the following stages :

Design. Business process design consists of identifying existing processes and designing new process models using BPEL[27] or BPMN[28]. The main objective of this step is to ensure that correct and efficient theoretical designs are prepared.

Modeling and Implementation. Processes previously designed are now modeled, then implemented in an executable process language.

Enactment. At this stage, the business processes are deployed and monitored using a Business Process Management System (BPMS).

Evaluation. The business process evaluation encompasses both business process optimization and reengineering.

In the literature, several works have addressed BPM. We can mention Milo *et al.* [BEKM05, BEKM06], whose formalized business processes as business graphs, and proposed *BP-QL* a visual query language for business processes. In [BEMP07, MD08], *BP-Mon* query language was proposed in order to monitor business processes in a distributed environment. Also *BP-Ex* that offers an uniform query-based and user-friendly interface for business processes analysis [BMS10].

A configurable process model captures multiple variants of a business process in a consolidated manner in order to avoid modeling and re-designing processes *from scratch*. In the same line, La Rosa *et al.* [RDtHM11] proposed a configurable process modeling notation, which incorporates features for capturing resources, data and physical objects. Then, the functionality and the architecture of *APROMORE*, an advanced process model repository, were described [RRvdA+11]. Dijkman *et al.* [DDvD+11] presented three similarity metrics to answer queries on process repositories : node matching similarity, structural similarity, and behavioral similarity that compares element labels as well as causal relations captured in the process model. Finally, Duma *et al.* [DGRU13] proposed an indexing structure to support the fast detection of clones in large process model repositories.

Regarding business process monitoring, Grigori *et al.* [GCC+04] presented a set of integrated tools that support business and IT users in managing process execution quality by providing several features, such as analysis, prediction, monitoring, control, and optimization. Recently, Mallick *et al.* [Mal11, MHD12] provided a new modelling approach

[27] Business Process Execution Language
[28] Business Process Model and Notation

to the problem of resource prediction in virtualized systems. Models are based on historical data to forecast shortterm resource usages. Fan *et al.* [FX14, FBXS14] proposed a differential privacy-based technique for privacy-preserving monitoring web browsing.

1.4.2 Business Process Decomposition and Identification

Basically, business process decomposition, i.e., fragmentation, is done to enhance business process execution, and for task distribution over a distributed system. For instance, Khalaf *et al.* [KL06] presented a mechanism for partitioning business processes, where each partition can be enacted by a different entity. The main goal was to disconnect the partitioning itself from the design stage, simplifying the reassignment of activities to different entities.

In the same line, Baresi *et al.* [BMM06] have introduced the idea of distributed orchestrations and have presented a proposal to couple BPEL and distributed execution in mobile settings. The proposed approach transforms a centralized BPEL process into a set of coordinated processes. An explicit meta-model and graph transformation supply the formal grounding to obtain a set of related processes, and to add the communication infrastructure among the newly created processes. We can also mention Caetano *et al.* [CST10], whose used the separation of concerns principle to facilitate the consistent decomposition of a business process and the unambiguous identification of its atomic activities.

One of the key activities to construct a successful SOA is the identification of services with the right level of abstraction. For this purpose, Ma *et al.* [MZZW09] introduced a measurement approach to quantitatively evaluate service identification. Indeed, a set of design metrics were used such as service granularity, coupling, cohesion, and business entity convergence. Ivanovic *et al.* [ICH10] presented an automatic fragment identification approach based on sharing between activities.

1.4.3 Service Selection, Composition, and Reuse

Another research direction is focusing in selection of services based on their Quality of Services (QoS) to reuse them. Awad *et al.* [ASKW11] presented an approach to business process design called *Design by Selection*, which takes advantage of process repositories during design and facilitates reuse of process model. Taher *et al.* [THP+11] presented an approach for achieving service reusability in Service-Based Applications (SBAs). The approach is based on decomposing the reusability requirements into two layers and then into separate views that allow the customization of business policies,

QoS, tasks, and control parameters. Huang *et al.* [HHLZ10] proposed an architecture enabling efficient reuse of process fragment. Indeed, services are organized into a network callled Service Composition Network (SCN), based on their co-occurence in the existing composite services. Process fragments are extracted according to both the structural contraint and the relevance of services. Yu *et al.* [YB12] proposed a multi-attribute optimization approach to tackle the issue of selecting service providers with the best user desired quality. Hung *et al.* [HTTA14] provided an anonymity-based solution to protect shema sharing and reuse against privacy concerns that discourage schema owners from contributing their schemas.

Service and fragment compositions are usually done to permit the reuse of existing services or to master the process complexity. Benatallah *et al.* [NBCT06] discussed the different ways in which the middleware can leverage business protocol descriptions, and focused in particular on the notions of protocol compatibility, equivalence, and replaceability. Rouached *et al.* [RFG10] proposed a semantic framework that provides a foundation for addressing the translation of communication between activities by supporting models of service choreography with multiple interacting Web services compositions.

Zemni *et al.* [ZBC10] applied soft constraints to model SLAs and to decide how to rebuild compositions which may not satisfy all the requirements, in order not to completely stop running systems. Ye *et al.* [YZB11] proposed an extensible QoS model to calculate the QoS of services in the cloud, then a genetic-algorithm-based approach to compose these services. Zheng *et al.* [ZZYB13] proposed a systematic approach to calculate the QoS for composite services with complex structures and taking into consideration of the probability and conditions of each execution path.

1.4.4 Securing Service Composition

Other research works have considered security aspects in Web service composition. Indeed, Carminati *et al.* [CFH06] have proposed a method to allow service requestors and providers to model their security constraints. Then, a brokered architecture were proposed to compose services according to the specified security constraint.

Meziane *et al.* [MBZ+10] adressed the problem of monitoring the compliance of privacy agreement, and proposed a monitoring system for controlling the private data usage in the area of web services. In the same line, Bacon *et al.* [BEE+10, BEP+14] proposed a data tagging schemes and enforcement techniques to have an end-to-end information flow control (IFC). Since IFC security is linked to the data that it protects, both tenants and providers of cloud services can agree on security policy, in a manner that does not require them to understand and rely on the particulars of the cloud software stack in

order to effect enforcement. She *et al.*[SYTB13] developed a three-phase composition protocol integrating flow control to enforce access control in composite services. For that, they considered composition time access control validation.

1.4.5 Data Integration and Mashup

Mashup is an application development approach that allows users to aggregate multiple services, each serving its own purpose, to create a service that serves a new purpose. In constrast with Web services composition where the focus is on the composition of business (process) services only, the Mashup framework goes further in that it allows more functionalities and can compose heterogeneous resources such as data services, UI services, etc. [LHPB09]

Trojer, Fung *et al.* proposed a SOA for privacy-preserving data mashup in the context of financial industry [TFH09, MFWH09] and then, in the context of high-dimensional data, i.e., social networks [FTH+12] when integrating data from multiple data providers. The solution uses an anonymity-based technique.

Elmeleegy *et al.* [EOEA10] implemented the *Hyperion system*, which employs technique based on noise selection and insertion to protect query results, and encryption-based technique to protect the mapping and ensure fairness among peers in Peer-to-peer data integration (i.e., Peer Data Management Systems). In the same line, the PAIRSE project [BBC+13] addressed the challenge of privacy-preserving data integration in peer-to-peer environments. For that purpose, Data Services were modeled as *RDF views* and query resolutions were done by a data services composition. To secure the service execution, a query rewriting based technique was used to integrate security and privacy policies, which are expressed using OrBAC, and to secure the service composition an encryption-based technique were used to encrypt the identifier attribute.

1.4.6 Business Process as a Service

Leymann *et al.* [AKL+09] discussed the outsourcing of company's processes and introduced a general compliance architecture that allows compliance to be monitored and enforced at services deployed in the cloud. Later, they investigated how the cloud delivery models affect the outsourcing of business processes modeled in WS-BPEL [ALMS09]. Cloud service provisionning across multiple cloud providers was studied and architectures were proposed in [HMLZ11, SZG+14].

Cloud blueprinting approach [PvdH11, Pap12, NLPvdH12] allows Service-based Application (SBA) developers to easily design, configure and deploy virtual SBA payloads

on VM. The blueprint concept is proposed as a uniform abstract description for the cloud service offerings that may cross different cloud computing layers. In the same line, Schumm *et al.* [SKK+11] presented advanced application scenarios for using process fragments in development of process-based application in the cloud through fragment library. Also, Taher *et al.* [THNvdH11] proposed T-Shaped platform which aims to develop a cloud based platform that bolsters the public service organizations to develop and deliver public services in efficient and cost-effective manner.

Pacheco and Puttini [PP12] presented an anonymity- based approach to protect cloud consumer's from information disclosure (ID, behavior, location, and data) using anonymity technology. The proposed framework enables anonymous message exchanges, while still allowing for the consumer to contract and have proper access to services and for the provider to authenticate, account, and charge for service usage, on demand. Goettelmann *et al.* [GDG+14] presented an approach for assessing security risks in a cloud context before distributing a business process execution accross multiple clouds.

Current BPaaS offerings can be perceived as monolithic cloud solutions. For this purpose, Taher *et al.* [THvdHF13] proposed a BPaaS engineering techniques which cater fot the tailoring of services to specific business needs using mixture of SaaS, PaaS, and IaaS solutions from various providers. Li *et al.* [LAC+14] developed a mathematical approach of Total Cost of Ownership (TCO) and evaluated a case study of Business Process as a Service.

1.5 Conclusion

At the end of the first chapter, we keep in mind several things. First, the emergence of cloud computing as a logical result of IT innovations that affected computers since their apparitions. Second, the prophecy of Leonard Kleinrock is produced. Nowadays, IT has really become the fifth utility. Finally, cloud computing facilitates the design of new business processes by sharing and reusing services, also known as design by selection. Moreover, it allows collecting and analysing data at large scale, also known as *big data*.

Therefore, our personal data may sometimes be abused without a real control over their use. To avoid this problem, a special care should be taken for privacy-preservation and confidentiality of data transfer, data storage, data treatment, and service sharing in the cloud. This should be translated into reality in the form of algorithms and secure protocols to reach a high level of security and control by using privacy by design.

Next, we present an overview of various security mechanisms that will be used later to secure sensitive data and services in the cloud.

2

Computer Security Background

Contents

2.1 Introduction

Nowadays, an increasing amount of data and services are outsourced for several reasons. This is principally due to the need to make them availables from anywhere to meet user *mobility* ; or simply to share (using *cloud computing* paradigm) information and knowledge in order to take advantage of external expertises. However, outsourcing data and services does not always mean that we wish to disclose them to unauthorized entities. In addition, it may be necessary to be able to ensure their availability and integrity. Thereby, such situations require effective *mechanisms* to guarantee that outsourced data and services are *safely* used and stored in the cloud.

This chapter emphasizes computer security towards a *survey*, and is structured as follows : In Section 2.2, we define what we mean by *computer security*. Section 2.3 introduces the first category of security mechanisms based on cryptography, then details symmetric and asymmetric schemes, gives their characteristics and discusses their security. We conclude the section by an introduction to homomorphic encryption. In Section 2.4, we give an overview about syntactic anonymity and its different variants such as k-anonymity, ℓ-diversity, and t-closeness to privacy-preserving data publishing and data mining. We conclude the section by an introduction to differential privacy. Section 2.5 discusses the use of group testing procedures to ensure the security of data. Section 2.6 concludes the chapter.

2.2 Computer Security

Historically, computer security was studied by networks and systems community [Sta10]. This is due to the fact that security issues started with the emergence of computer networks and the need to transfer data. However, with the emergence of technologies as distributed databases, Web 2.0, cloud computing, big data, and Internet of things ; and the need of outsourced databases and Web Services, databases and services communities are increasingly interested in security issues, that require specific knowledge, in order to use the most adapted security mechanisms to such context. For this purpose, we will use some terms and definitions coming from networks and systems community.

The *NIST*[1] Computer Security Handbook [GR95] defines the term *computer security* as follows : *"The protection afforded to an automated information system in order to attain the applicable objectives of preserving the* ***integrity***, ***availability***, *and* ***confidentiality***

[1]The NIST (National Institute of Standards and Technology) is an U.S. federal agency that deals with measurement science, standards, and technology related to U.S. government use and to the promotion of U.S. private-sector innovation. Despite its national scope, NIST Federal Information Processing Standards (FIPS) and Special Publications (SP) have a worldwide impact.

of information system resources includes hardware, software, firmware, information / data, and telecommunications."

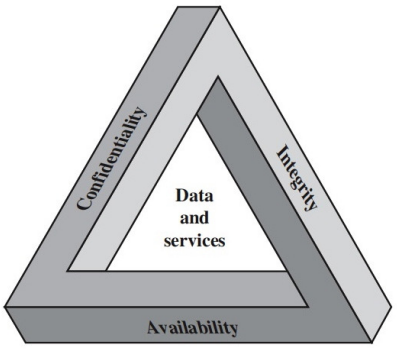

FIGURE 2.1: CIA triad according to [Sta10].

As depicted in Figure 2.1, the NIST definition introduces three key objectives that are at the heart of computer security. These three concepts, which form what is often referred to as the *CIA triad* [Sta10], are discussed in the following :

Confidentiality. The term confidentiality covers two related concepts, data confidentiality and privacy. Data confidentiality assures that *private* and *confidential* information is not made available or disclosed to unauthorized users. However, privacy assures that users *control* or *influence* what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

A *loss of confidentiality* is the unauthorized disclosure of information.

Integrity. The term integrity also covers two related concepts, data integrity and system integrity. Data integrity assures that information and programs are changed only in a specified and authorized manner. However, system integrity assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

A *loss of integrity* is the unauthorized modification or destruction of information.

Availability. Availability assures that systems work promptly and services are not denied to authorized users.

A *loss of availability* is the disruption of access to or use of information or an information system.

After defining what we mean by computer security, we now introduce how to ensure the security of (or simply secure) a computer system. The computer networks and systems community defines several *security services*, which are provided by a layer of communicating open systems, to ensure adequate security of data and services. A clearer

definition of **security service** is given in *RFC 2828* [Shi00] : *"a processing or communication service that is provided by a system to give a specific kind of protection to system resources ; security services implement security policies and are implemented by security mechanisms".*

The *ITU*[2] (International Telecommunications Union) divides security services into Five categories : authentication (i.e., the assurance that the communicating entity is the one that it claims to be), access control (i.e., the prevention of unauthorized use of a resource), confidentiality (i.e., the protection of data from unauthorized disclosure), integrity (i.e., the protection of data from unauthorized modifications), and nonrepudiation (i.e., assurance against denial by one of the entities involved in a communication of having participated in the communication) [ITU91].

In addition, availability (i.e., the assurance that a system or a system resource being accessible and usable upon demand by an authorized system entity), that was treated as a property to be associated with various security services, can perfectly be defined as an *independent* security service [Sta10].

We will present some security mechanisms : *cryptography*, *anonymization*, and *combinatorial group testing* that we consider as our security toolbox. Later, these mechanisms will be used to implement diverse security services. Note that we can use only one security mechanism or a combination of several security mechanisms to address a given security issue in the cloud.

2.3 Cryptographic Basics

First we will settle upon the meaning of *cryptography*[3], which is the study of methods for sending messages in *secret* or *disguised form* to protect several aspects of data, in particular confidentiality and authenticity, against adversary who tries to break the security.

Cryptography has, as its etymology, *kryptos* from the Greek, meanning *hidden*, and *graphein*, meaning *to write*. The original message is called the *plaintext*, and the deguised message is called *ciphertext*[4]. The process of transforming plaintext into ciphertext is called *encipherment* or *encryption*, and the reverse process accomplished by the message's recipient, is called *decipherment* or *decryption*. *Cryptanalysis* is the study of

[2]The ITU is an international organization within the United Nations System in which governments and the private sector coordinate global telecom networks and services

[3]The (English) term cryptography was coined in 1658 by Thomas Browne, a British physician and writer.

[4]The term 'cipher' in English comes from the Arabic word 'sifr'.

methods to break cryptosystems. In contrast with steganography, which conceals the very existence of the message, namely *covert secret writing*, cryptography transforms the data mathematically, generally using a key [Mol07].

Initially, cryptography focused on protecting confidentiality in the context of military and diplomatic communication. Nowadays, with the emergence of high-speed networks, computers, and the replacement of postal mail by electronic communication in such applications as bank transactions, access to worldwide databases as in the WWW, cloud, e-mail, etc. This implies a whole new range of security needs, as previously defined as security services, that need to be adressed, for example : authentication and identification.

2.3.1 Cryptographic Primitives

Cryptography has been studied and used for centuries [AK92, Sac77]. In the first part of the section, we will present some historical cryptosystems to lay the foundation for describing modern cryptography. Then, we will formally define some key terms, which will be used later in this dissertation.

2.3.1.1 Towards Modern Cryptography

It is believed that the oldest known text to contain one of the essential components of cryptography, a *modification of the plaintext*, occurred some 4000 years ago in the Old Kingdom of Egypt. In Mesopotamia, some clay tablets, dated near 1500 BCE, were clearly used to encipher a craftsman's recipe for pottery glaze. Later, the ancient Greeks and Romans employed the *Scytale* transposition and *Caesar* cipher respectively, to protect information of military significance.

Modern cryptography differs from historical cryptography in many aspects. The most important is mathematics which plays a more important role than ever before [Des09a]. In the Codebreakers [Kah96], David Kahn notes that modern cryptography originated among the Arabs. In fact, Al-Kindi, working on ciphers and ciphertexts obtained from the ancient Greeks and Romans, as well as ciphers used at his time sometime around 800, has described (in the greatest treatise entitled *A manuscript on deciphering cryptographic messages*) the first cryptanalysis technique based on frequency of letters in a language.

In the following, we will detail the most significant historical ciphers :

Substitution Ciphers. A *substitution* cipher aims to replace plaintext symbols with other symbols to produce ciphertext. As an example, the plaintext might be

cloud, and the ciphertext might be *FRTGH* when *c, l, o, u, d* are replaced by *F, R, T, G, H* respectively. The cryptographic convention is to use *lower-case* letters for *plaintext* and *UPPER-CASE* letters for *CIPHERTEXT*. Obviously, for the English/French alphabet, there are $26! = 403291461126605635584000000$, roughly 4×10^{26} different combinations.

We discuss now the security of the scheme. Assuming that the adversary knowns the ciphertext and the fact that a substitution cipher was used. Such an attack is called a *ciphertext-only attack*. Shannon's theory of secrecy tells us that an exhaustive key search would roughly take 3.6×10^5 years before finding a sufficiently correct key [Sha49]. However, a faster method, based on redundancy in a language, exists for breaking a substitution cipher [Lew00]. In fact, as depicted in Figure 2.2, the frequency of individual letters, as *e, t, o, a, n, i, r, s, h* , and also diagrams, as *th, er, ed, es, en, ea*, can be used to identify most of letters, where the most frequent letter / diagram in the ciphertext corresponds to *e* / *th* respectively. If mistakes are made, they are easily spotted, and one can recover using backtracking [Des09a].

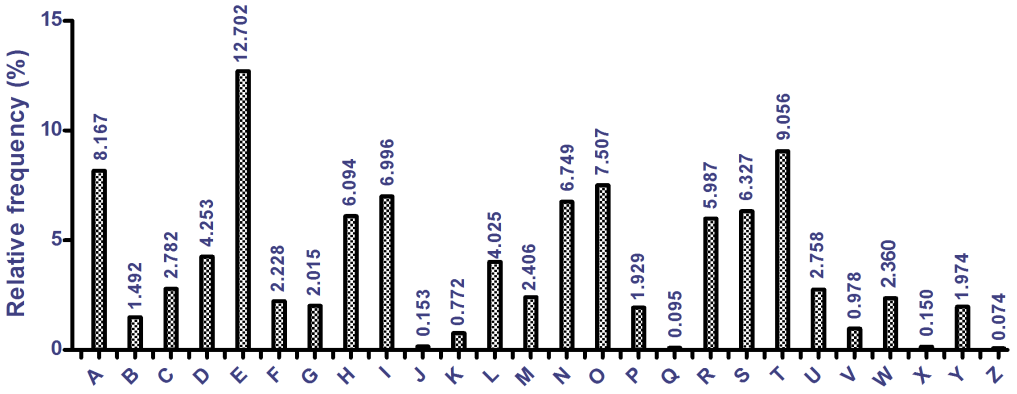

FIGURE 2.2: Relative frequency of letters in English text according to [Lew00].

Transposition Ciphers. In a *transposition* cipher, also known as a permutation cipher, we permute the places where the plaintext letters sit. The plaintext is divided into groups of equal length, and a permutation applied to groups according to the key. As an example, for $d = 5$, we might have (2 3 1 5 4) as a permutation. If the plaintext is *cloud*, the ciphertext will be *LOCDU*. We note that only the frequency of diagrams is affected by encrypting the plaintext [Des09a]. In fact, an adversay can try to restore the distribution of diagrams and trigrams. Sequential application of two or more transpositions will be called compound transposition.

Julius Caesar. One of the oldest cryptosystems is *Caesar cipher*. It consists merely in a shift to the right of three places of each plaintext letter to achieve the ciphertext letters. This is best illustrated by Table 2.1. The plaintext *cloud* for example,

would become *FORXG* in this system. We consider the value +3 as a *enciphering/deciphering key*, that we may regard as a *shared secret* between the sender and the recipient, which *unlocks* the cipher [Mol07].

TABLE 2.1: Caesar cipher table

Plain	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
Cipher	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*
Plain	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
Cipher	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*

The problem with this scheme is an attacker who knows how the ciphertext is encoded can break it (doing a shift to the left of three places of each ciphertext letter). To prevent this, a key $k \in \mathbb{N}$ can be added. For that, we use a more modern variant of the Caesar cipher. Consider Table 2.2 that gives numerical values to the English/French alphabet that simplifies the process.

TABLE 2.2: Numbers-based Caesar cipher table

Plain	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
Cipher	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
Plain	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
Cipher	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*

Each symbol m_i of the plaintext *cloud* is mapped into a number. The numerical equivalent is $2, 11, 14, 20, 3$. To encrypt with this variant of Caesar cipher, we add modulo n the key k to the symbol m_i, represented as an integer between 0 and $n-1$. The corresponding symbol in the ciphertext is $c_i = m_i + k \ mod \ n$.
Regarding the security of the scheme, redundancy in a language, as in the substitution cipher, can be used by attackers to reveal the correct plaintext.

2.3.1.2 Formal Definitions

Definition 2.1. (Cryptosystems/Ciphers) [Mol07, Des09a]
A cryptosystem, also called a cipher or an encryption scheme is composed of an encryption algorithm E and a decryption algorithm D. Consider a plaintext $m \in M$ and a key $k \in K$ as input to E, the output of the encryption is called the ciphertext $c \in C$, where :

$$c = E_k(m) = E(k, m) \tag{2.1}$$

The decryption algorithm D has input a key $k' \in K'$ and a ciphertext $c \in C$, outputs the plaintext $m \in M$, where :

$$m = D_{k'}(c) = D(k', c) \tag{2.2}$$

The keys (k, k') are called a key pair where possibly $k = k'$.

Definition 2.2. (Entity, Channel, and Protocol)
An entity is any person or things, such as computer terminal, which can send, receive, or manipulate information.
A channel is any means of communicating information from one entity to another.
A cryptographic protocol means an algorithm involving two or more entities, using cryptography to achieve a security goal.

Definition 2.3. (One-Wayness) [Poi02]
Consider a cryptosystem with non reversible encryption algorithm E and a plaintext $m \in M$. One-wayness proprety says that it is not possible to find the plaintext m such that $c = E_k(m)$ without knowing the key k. In other words, we can easily compute E, but it is computationally infeasible to compute E^{-1}.

Definition 2.4. (Indistinguishability/Semantic Security) [GM84]
A cryptosystem is semantically secure if any probabilistic, polynomial-time algorithm (PPTA) that is given c the ciphertext of a certain message m, and the message's length, cannot determine any partial information on the plaintext message m.

Definition 2.5. (Non-Malleability) [DDN00]
A cryptosystem is malleable if it is possible for an adversary to transform a ciphertext c into another ciphertext c' which decrypts to a related plaintext m'.

Definition 2.6. (Hash Function) [Mol07]
A hash function is a computationally efficient function that maps bitstrings of arbitrary length to bitstrings of fixed length, called hash values.

Definition 2.7. (One-Way Hash Function) [Mol07]
A one-way hash function H is a hash function where it is computationally easy to compute $c = H(m), \forall m \in M$ and computationally infeasible to find $m \in M$ from a randomly chosen ciphertext c.

Definition 2.8. (Levels of Security) [Des09a]
Given C and C' two ciphers. Different models can be used to define the security of C and C'. We distinguish :

1. *Heuristic security model.* C and C' are heuristically secure as long as no attack has been found. The attacker has a bounded computer power.

2. *As secure as model.* C is as secure as C' if we can prove that a new attack against C' implies an attack against C, and vice versa. The attacker has a bounded computer power.

3. *Proven secure model.* First we formally model what security is, and give some assumptions. After, if we can prove that assumptions are true, then the formal security definition is satisfied for C and C'. The attacker has a bounded computer power.

4. *Unconditionally secure model.* C and C' are unconditionally secure if the attacker has an unbounded computationally power and satisfies the formal definition of security.

5. *Quantum secure model.* A special class of cryptosystem, called *quantum cryptography*, that assumes the correctness of the laws of quantum physics. A more complete description can be found in [BB84].

Definition 2.9. (Attacks) [MVO96, Mol07, Sta10, Des09a]
A security attack is any action that compromises the security of information owned by an organization. An attack on a cryptosystem is any method that starts with some information about the plaintext and the ciphertext enciphered using a secret key, and ends with determining the key and the plaintext. There exists two classes of attacks :

Passive attacks attempt to learn or make use of information from the system but do not affect system resources. Basically, the attacker monitors the communication channel (i.e., eavesdropping message contents and traffic analysis) in order to threaten confidentiality. This class of attacks is very difficult to detect, because it does not involve any alteration of transmitted data. However, measures are available to prevent their success. An adversasy which causes this kind of attacks is *curious*. Typically, passive attacks are classified as follows :

- *Chosen-plaintext.* The attacker chooses plaintext, is then given corresponding ciphertext, then analyzes the data to compute the enciphering key in order to determine plaintexts from ciphertexts.
- *Chosen-ciphertext.* The attacker chooses ciphertext, is then given corresponding plaintext, then analyzes the data to deduce plaintexts from other intercepted ciphertexts.
- *Known-plaintext.* More practical then chosen-plaintext, the attacker has some amount of pairs (plaintext, ciphertext) that may suffice to find the key.
- *Ciphertext-only.* Even more practical then known-plaintext, the attacker has only ciphertext as information to deduce the key and plaintext. Cryptosystems that are vulnerable to ciphertext-only attacks are completely insecure.
- *Adaptive chosen-plaintext.* This attack is a chosen-plaintext attack where the choice of plaintexts depends upon the previously received ones.

- *Adaptive chosen-ciphertext.* This attack is a chosen-ciphertext attack where the choice of ciphertexts depends upon the previously received ones.

There are some passive attacks that in theory break any cryptosystem. However, they are impractical because they require the attacker to do far too much work. For examples :

1. *Brute-Force attacks.* Also called an exhaustive search of the keyspace. In this attack, the adversary tries all possible keys to determine which one is being used to encrypt a plaintext.
2. *Dictionary attacks.* This attack occurs when an adversary takes a list of probable plaintexts, encrypts all the entries on the list, and compares this list with the list of actual ciphertexts in an effort to find a match.
3. *Birthday attacks.* The birhday attack is based on the mathematics exemplified by the birthday paradox. It can be used whenever the issue is finding repeated ciphertexts from some cryptographic technique, e.g., two inputs hashing to the same result.

Active attacks attempt to alter system resources or affect their operations. Basically, the attacker attempts to add, delete, or alter the message in order to threaten not only confidentiality, but also integrity and availability. We note that it is difficult to prevent this class of attacks, because of the wide variety of potential physical, software, and network vulnerabilities. The principal goal will be to detect active attacks and to recover from any disruption or delays caused by them. An adversasy which causes this kind of attacks is *malicious*. In the following, we give some examples of active attacks :

1. *Masquerade or Spoofing.* The attacker pretends to be a different entity. Masquerade attack attempts to utilize an alternate identity while threatening a system and almost always uses other forms of attack in conjunction with this method.
2. *Replay.* The attacker captures information and later attempts to reuse, replay, that information in order to gain access to protected data.
3. *Modification (substitution, insertion, and destruction).* In this attack, some parts of the legitimate messages are altered or deleted, or fake messages being processed between two or more entities are generated.
4. *Denial of service.* In this attack, the normal use of the system is prevented or inhibited (e.g., a server is flooded by fake requests so that it cannot reply normal requests).

2.3.2 Conventional / Symmetric Ciphers

We distinguish two kinds of cryptosystems : *symmetric* cryptosystems and *asymmetric* cryptosystems. A cryptosystem is called conventional or symmetric if it is easy to compute k' from k, where the pair (k, k') is the encryption and decryption key respectively. Caesar cipher, previously defined, is an example of a symmetric cryptosystem. However, given a key k, if it is hard to compute k' then k can be made public and the cryptosystem is called a public key or asymmetric cryptosystem [Des09a].

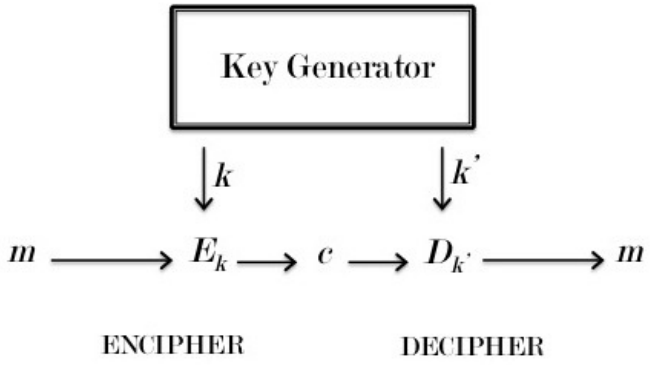

FIGURE 2.3: Model of symmetric ciphers.

The symmetric cryptosystem depicted in Figure 2.3 is formally defined as follows :

Definition 2.10. (Symmetric-Key Ciphers) [Mol07]
A cryptosystem is called symmetric-key, single-key, one-key, or conventional, if for each key pair (k, k'), the key k' is *computationally easy* to determine knowing only k and similarly to determine k knowing only k'. A computationally easy problem can be solved in expected polynomial time.
A symmetric-key cryptosystem is semantically secure. Thus, an adversary must not be able to compute any information about a plaintext from its ciphertext.

In symmetric ciphers, the encryption algorithm E uses secret keys k_i to perform various substitutions and transformations on the plaintext m. As result, different outputs may be produced depending on the specific key k_i being used at the time. In fact, the exact substitutions and transformations performed by the algorithm E depend on a secret key k, that is independent of the plaintext m and of the algorithm E. It is essential that the algorithm used for encryption is at least resistant to ciphertext-only attack.
When using symmetric ciphers, we must be sure that the sender and the receiver have obtained copies of the secret key in a secure fashion. Moreover, we must keep secret the shared key, because if an attacker discovers it, all communications using this key are readable [Sta10].

Stream and block ciphers are the two major classes of symmetric cryptosystems. The distinctions between them are more readily seen in practice than in theory. Usually, stream ciphers are used to encrypt small strings (one bit, byte or word) using a transformation cipher which varies over time. However, in a block cipher, the same encryption algorithm is applied to different strings derivatives of plaintext message, which are more consistent size, typically 64, 128 or 256 bits [MVO96].

Intuitively, study of these two classes will be useful. Indeed, databases contain variable size columns, which can be as small as a one bit or containing several hundred bits. This encourages us to consider both, public and secret key ciphers, as interest to achieve our goals. Before that, we give some examples of symmetric ciphers.

Monoalphabetic and Polyalphabetic Ciphers. A *homophone* is a ciphertext symbol that always represents the same plaintext symbol. *Monoalphabetic* cipher means that only one cipher alphabet is used. In the Caesar cipher, the letter D is always the ciphertext of the plaintext a, so D is a homophone in the *monoalphabetic* Caesar cipher. Note that monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.

A countermeasure is to provide multiple substitutes for a single plaintext letter. For example, the plaintext a could be assigned a number of different cipher symbols, such as D, K, V, and X, with each homophone assigned to a letter in rotation or randomly. We use the term *polyphone* to refer to a ciphertext symbol that always represents the same set of plaintext symbols, typically a set consisting of at most three plaintext symbols [Mol07].

A cipher is called *polyalphabetic* or *periodic substitution* if it has more than one cipher alphabet. In this type of cipher, the relationship between the ciphertext substitution for plaintext symbol is variable (not fixed as in monoalphabetic ciphers). Practically, the plaintext m is split into blocks of equal length, called the period d. We use d monoalphabetic substitution ciphers by encrypting the i^{th} symbol ($1 \leq i \leq d$) in a block using the i^{th} substituion cipher [Des09a].

Polyalphabetic ciphers have the following features in common [Sta10] :

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

The cryptanalysis is similar to the simple monoalphabetic ciphers once the period d has been found. To find the exact period, the Kasiski method [Kas63] analyzes repetition in the ciphertext ; and Friedman [Fre20] uses index of coincidence.

The Vigenère Auto-key Cipher. An example of polyalphabetic ciphers is due to Vigenère, whom has exploited an idea, of using the plaintext as its own key, that others had invented. Morever, Vigenère added something new, called a *priming key*, which is a single secret letter that is used to encipher the first plaintext letter, which would, in turn, be used to encipher the second plaintext, and so on. In fact, the periodic nature of the keyword is eliminated by using a nonrepeating keyword that is as long as the message itself. Vigenère proposed what is referred to as an auto-key system, in which the keyword is concatenated with the plaintext itself to provide a running key [Mol07].

The Vigenère cipher was considered, up to the middle nineteenth century, to be unbreakable. However, in 1863 Kasiki [Kas63] found a method for cryptanalyzing it. The method is based on the observation of repeated portions of plaintext enciphered with the same part of a key must result in identical ciphertext patterns. The key and the plaintext share the same frequency distribution of letters and a statistical technique can be applied. For example, the letter e enciphered by e can be expected to occur with a frequency of $0.127^2 \simeq 0.016$. These regularities can be exploited to achieve successful cryptanalysis [Sta10].

2.3.2.1 Stream Ciphers

We start by formally defining stream ciphers, and their three subclasses, synchronous, self-synchronizing, and nonsynchronous. Then, we will discuss the process of generating the encryption key, which remains a key point in stream ciphers.

Definition 2.11. (Keystreams, Seeds, and Generators) [Mol07]
If K is a keyspace for a set of enciphering transformations, then a sequence $k_1k_2k_3 \ldots \in K$ is called a keystream.
A keystream is either randomly chosen or generated by an algorithm, called a keystream generator, which generates the keystream from an initial small input keystream called a seed.
Keystream generators that eventually repeat their output are called periodic.

Definition 2.12. (Stream Ciphers) [Mol07]
Let K be a keyspace for a cyptosystem and let $k_1k_2k_3 \ldots \in K$ be a keystream. The cryptosystem is called a stream cipher if encryption upon plaintext strings $m_1m_2m_3 \ldots$ is achieved by repeated application of the enciphering transformation on plaintext message units as :

$$E_{k_i}(m_i) = c_i \tag{2.3}$$

If k'_i is the inverse of k_i, then deciphering occurs as :

$$D_{k'_i}(c_i) = m_i, \ for \ i \geq 1 \tag{2.4}$$

If there exists an $l \in \mathbb{N}$ such that $k_{i+l} = k_i$ for all $i \in \mathbb{N}$, then we say that the stream cipher is periodic with period l.

Definition 2.13. (Synchronous and Asynchronous Stream Ciphers) [Mol07] Given a stream cipher C, and $k_1k_2k_3\ldots \in K$ a keystream. C is said to be :

Synchronous Cipher if the keystream is independant of both, plaintext and ciphertext.

Self-synchronous Cipher if the keystream is generated as a function of the key and a fixed number of previous ciphertext units.

Nonsynchronous Ciphers if the keystream is generated as a function of the plaintext.

In the following, we will limit the use of the term stream ciphers to synchronous ciphers. This is motivated by the fact that asynchronous ciphers are became obsolete. In fact, the implementation of synchronous stream ciphers can guarantee that a single bit error will result in only a single bit of corrupted plaintext. Thus, synchronous stream ciphers would be useful where lack of error propagation is critical. However, use of asynchronizing stream ciphers can result in error propagation.

In stream ciphers, an encryption algorithm consists in combining the plaintext with a binary sequence having the same length, called the key. Let $k = (k_i)_{i\geq 0}$ to be this sequence generated by an algorithm, called keystream generator. The role of the keystream generator is to generate at every moment i, a m-bit block, k_i, which is a function of its internal state x_i.

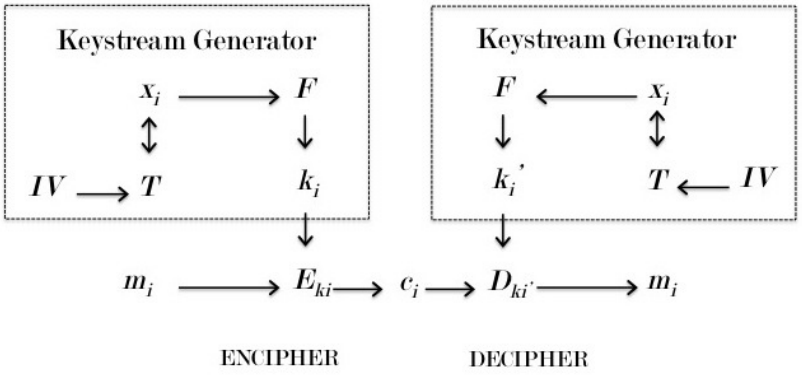

FIGURE 2.4: Synchronous stream ciphers.

We now explain how a keystream generator operates. Basically, a keystream generator comprises three functions, as described in Figure 2.4 [Jac12] :

An Initialization Function. Using a seed (i.e., a secret key) and a public initialization vector (IV), an initial state x_0 of the generator is calculated. Sometimes, this step may be divided into two phases :

1. A seed loading phase, which consists in computing a value depending only on the secret key.
2. An IV injection or resynchronization phase, which determines the initial state x_0 of the generator from the IV and the value obtained in the previous phase.

This permits to save time when only the IV is changed (without changing the secret key), which is common. For example, when using a databases where the data may be very small and the IV varies from a column to another and from an update to another.

A Transition Function. denoted T, it consists of modifying the internal state of generator from x_i to x_{i+1} corresponding respectively to the instant i and $i+1$. Usually, this function is fixed, but it may vary depending on the key, IV, and even over time.

A Filtering Function. denoted F, which returns the key k_i from the current internal state x_i. For simplicity and space for hardware implementations, the filter function is generally fixed as the transition function.

Stream ciphers are faster than block ciphers from the perspective of hardware. The reason is that stream ciphers encrypt individual plaintext message of one binary digit at a time. A small size allows reducing both, time and memory space, needed to store the ciphertext before obtaining a full block. Moreover, stream ciphers do not need to make a padding, which is highly appreciable when bandwidth is low or the communication protocol requires the use of short packets. However, stream ciphers are not suitable for software implementation since the manipulation of a small block is time consuming.

One-Time Pad

We previously saw that Vigenère cipher was cryptanalysed by using frequency distribution of letters. The ultimate defense against such a cryptanalysis is to use a cipher where the key has as many symbols as the plaintext itself, and the key is truly randomly generated (with no statistical relationship to plaintext) and never used more than once.

The one-time pad, introduced by an AT&T engineer named Gilbert Vernam in 1918, and Shannon's analysis of its security are considered as the most important discoveries in modern cryptography [Des09a].

In this stream cipher, both the key and plaintext are written in binary, namely the alphabet of definition is $A = \{0, 1\}$. A binary operation is defined, for example, the exor (exclusive-or) $\oplus$. The system can be expressed succinctly as :

$$c_i = m_i \oplus k_i \tag{2.5}$$

where c_i, m_i and k_i are the i^{th} binary digit of ciphertext, plaintext and key respectively. To decrypt, we compute :

$$m_i = c_i \oplus k_i^{-1} \tag{2.6}$$

The key is used only once. This implies that if a new message needs to be encrypted, a new key is chosen, which explains the terminology one-time pad.

We discuss now the security of the scheme. One-time pad is unbreakable, because the ciphertext contains no information whatsoever about the plaintext (ciphertext has an uniform distribution), there is simply no way to break the code. In practice, two fundamental difficulties exist :

1. Making a large quantities of random keys.
2. Key distribution and protection.

Because of these difficulties, the one-time pad is of limited utility and is useful primarily for low-bandwidth channels requiring very high security.
Shannon [Sha49] defined an encryption system to be perfect when, for a cryptanalyst not knowing the secret key, the plaintext m is independant of the ciphertext c. Then, he proved that the one-time pad is perfect and the length of the key must be at least the entropy of the message. More recent work has demonstrated that the length of the key must be at least the length of the plaintext [BDSV95].

2.3.2.2 Block Ciphers

Block ciphers are the most prominent and important elements in modern cryptographic systems. In fact, many encryption schemes are polygram substitution ciphers, which are a substitution of many symbols at once. The problem to obtain a practical scheme was principally the number of possible keys that needs to be reduced. Shannon proposed

a method based on using substitution and transposition at the same time, and Feistel [Fei73, FNS75] adapted it. Formally,

Definition 2.14. (Block Ciphers)
A block cipher is a cryptosystem that separates the plaintext message into strings, called blocks, of fixed length $k \in \{64, 128, 160, 256\}$ bits, called the blocklength. Then, a mode maps the n-bit plaintext blocks to n-bit ciphertext blocks at a time.
Block ciphers are devided into two types, substitution and transposition based ciphers and Feistel scheme based ciphers.

Modes of operation

Symmetric block ciphers have five *modes of operation* recommended by NIST. These modes are meant to address every conceivable application for cryptology to which block ciphers can be applied. In the following, we give an overview of block cipher modes [Mol07, Des09a] :

Electronic Code Book (ECB). Each n-bit plaintext block is enciphered with the same key, albeit independently. ECB uses substitution ciphers and is vulnerable to text redundancy based attacks. If the same key is used for too long a time, most parts of the plaintext can be recovered. This mode is not recommended and only used to send small amount of data such as a symmetric key.

Cipher Block Chaining (CBC). The input is the addition, modulo 2, (EXOR) of the previous n-bit ciphertext with the succeeding n-bit plaintext. In addition, the initial vector IV used, should be unpredictable. Normally, this mode is used as a general-purpose block-transport mechanism but also may be employed for authentication purposes.

Cipher Feedback (CFB). This mode employs a chaining mechanism similar to CBC. The ciphertext block $c_i = m_i \oplus Select_n(k, d_i)$ where $Select_n$ selects the n most significant bits of d_i, and d_i is the input to the cipher. d_i is constructed from the least significant bits of d_{i-1} (the previous input), shifted to the left by n positions, and concatenated with c_{i-1} the n-bit ciphertext. The initial vector IV used, should be unpredictable. This mode is employed as a stream-cipher-oriented means for general-purpose messaging since it processes $n \in \mathbb{N}$ at a time.

Output Feedback (OFB). This is comparable with CFB mode with the exception that its input is the prior block cipher's output. This mode is employed for stream-cipher-oriented communications, especially those requiring message authentication.

Counter (CTR). Similar to the OFB mode, in the sense that both form stream cipher. However, the input to the encryption algorithm is the output of a counter. This mode is remarkably easy to use and is typically utilized for high-speed transmission.

Feistel Cipher.

The Figure 2.5 depicts the structure proposed by *Feistel* [Fei73, FNS75]. A Feistel cipher is a block cipher that inputs a plaintext pair $m = (L_0, R_0)$, where both halves L_0 and R_0 have bitlength b, where : $b = \frac{blocklength}{2} \wedge b \in \mathbb{N}$, and outputs a ciphertext pair (L_1, R_1), where L_1 and R_1 have the same bitlength $b \in \mathbb{N}$, according to an iterative process F, making it what is called an *iterated block cipher*.

The key k is input and subkeys k_j for $j = 1, 2, 3, \ldots, r$ are generated from it via a specified key schedule. Generally, $k_j \neq k_i$ for $j \neq i$, and $k \neq k_j$ for any j. Formally,

$$(L_1, R_1) = (R_0, L_0 \oplus F(k_1, R_0)) \tag{2.7}$$

We note that such a system is easily reversible and the decryption is easily deduced :

$$(L_0, R_0) = (R_1 \oplus F(k_1, L_1), L_1) \tag{2.8}$$

This allows us to use the same circuit for encrypting and decrypting. The function F, called a *round function* iterated over r rounds, all of which have the same construction, acts on plaintext pairs.

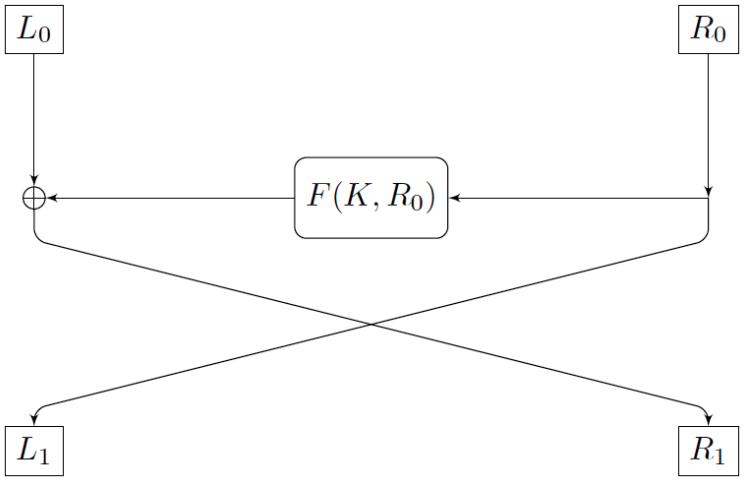

FIGURE 2.5: Feistel scheme.

We now look at some design features of Feistel ciphers :

Block size. Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm. A 64-bit blocklength having been common, but *blocklength* $\geq$ 128 bits or more, are becoming standard due to modern demands stemming from increased cryptanalytic developments.

Keylength. Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion. Keylength of 64 bits or less are now widely considered to be inadequate. Typically, 128-bit keylengths are becoming standard.

Rounds and round functions. The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is sixteen rounds. A round function with increased complexity adds to the security.

Subkeys. Generation of subkeys from an input key k during the operation of the algorithm aids in thwarting cryptanalysis.

After introducing block ciphers, we now present DES and AES encryption schemes.

Data Encryption Standard (DES)

The most widely used encryption scheme is based on the *Data Encryption Standard (DES)* adopted in 1977 by the NIST [FIP77]. DES is basically a block cipher combining fundamental cryptographic techniques, *confusion* and *diffusion*. Confusion obscures the relationship between the plaintext and the ciphertext, which thwarts a cryptanalyst's attemps to study the ciphertext by looking for redundancies and statistical patterns. It is necessary to have a deeply complex substitution algorithm in order to cause confusion. Diffusion dissipates the redundancy of the plaintext by spreading it over the ciphertext, which frustrates a cryptanalyst's attemps to search for redundancies in the plaintext through observations of the ciphertext. To cause diffusion, we repeatedly perform permutations on data [Mol07].

In DES, data are encrypted in 64-bit blocks using a 56-bit key. To encrypt a message longer than 64 bits, a mode is used. Since DES algorithm is outdated, we discuss it briefly. As described by the NIST [FIP77], the DES algorithm consists of three fundamental phases :

- The enciphering computation which follows a typical Feistel approach is described in Figure 2.6.

- The calculation of $F(R_{i-1}, k_i)$.
- The key schedule calculation.

The decryption algorithm is identical to the encryption operation, except that it uses subkeys with reverse order $k_{16}k_{15}k_{14}\ldots$.

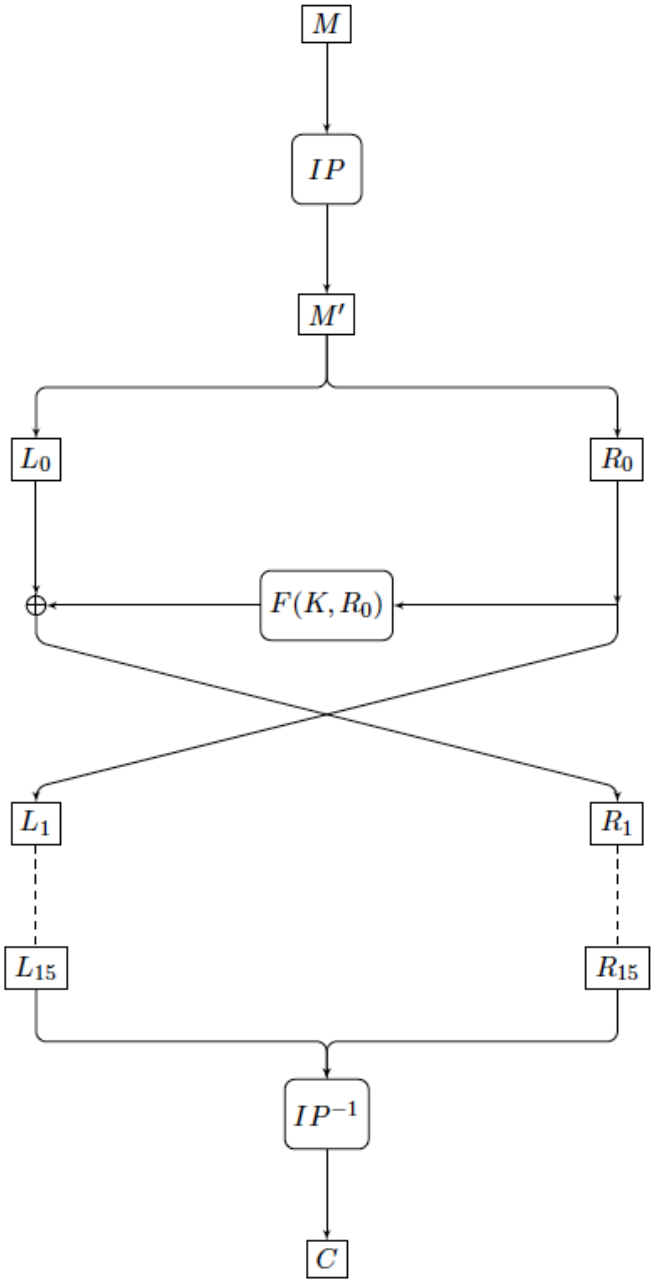

FIGURE 2.6: DES block diagram of the enciphering computation.

We now discuss the security of DES. There exist many types of attacks that can break DES algorithm without testing all possible keys. In 1991, Eli Biham and Adi

Shamir [BS90] used *differential cryptanalysis*. Thereby, based on chosen-plaintext attack, they used 2^{47} pairs of (plaintext, ciphertext) to find the encryption key. Later, Mitsuru Matsui [Mat93], based on *linear cryptanalysis*, has improved the number of pairs used to 2^{43}. Other attacks exist, in practice, the most effective remains the brute-force attack. In fact, a dedicated machine was produced in 1998, by the Electronic Frontier Foundation, which found an encryption key in 3 days [EFF98]. A FPGA-based machine has improved this time to less than one day.
To avoid this weakness, double and triple encryption are used [Des09a]. Both use a 112 bit key. Double encryption DES is obtained by running :

$$DES_{k_1} \circ DES_{k_2} \tag{2.9}$$

Triple encryption uses DES as encryption and as decryption, denoted as DES^{-1} giving :

$$DES_{k_1} \circ DES_{k_2}^{-1} \circ DES_{k_1} \tag{2.10}$$

However, the blocklength of the double and triple variants is too short for high security. This has resulted the withdrawal of DES standard by NIST in 2005.

Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a new encryption standard since November 26, 2001 [FIP01]. It was proposed by Rijndael [DR02] and selected by the NIST as an unclassified and publicly disclosed (open) encryption algorithm, to replace DES for protecting sensitive data.
Rijndael cipher uses substitutions and transpositions, and is based upon the 128-bit block cipher, called *square*, which Rijmen and Daemen originally designed with a concentration on resistance against linear cryptanalysis.

The standard AES is a symmetric block cipher that takes a plaintext/ciphertext block size of 128 bits. The keylength can have 128, 192, or 256 bits, and the algorithm is referred to as $AES-128$, $AES-192$, or $AES-256$ respectively. The number of rounds varies depending on the keylength, that is, 10, 12, or 14 rounds, and all operations are performed on 8-bit bytes. The first and last round differ slightly from the other rounds [DR02, Mol07, Des09a].

In order to give even a brief description of AES, we need to describe its essential components :

The state. The *State*, is the intermediate cipher resulting from application of the round function. It can be depicted as a $4 \times Nb$ matrix, where Nb is the blocklength

diveded by 32. In Table 2.3, we show a State for an input block of length 128 bits, it would have 16 bytes as a 4×4 matrix ($Nb = \frac{128}{32} = 4$). Note that the input

TABLE 2.3: AES State

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

block is put into the State by column and in the exection of the cipher the bytes are taken in the same order.

The cipher key. As with the State, the *cipher key* can be depicted using a $4 \times Nk$ matrix, where Nk is the keylength diveded by 32. For example, for an key of length 192 bits, it would have 24 bytes as a 4×6 matrix ($Nk = \frac{192}{32} = 6$).
A *round key* is derived from the cipher key by means of the following *key schedule* :

1. The total number of round key bits equals $B \times (Nr + 1)$ where B is the blocklength and Nr is the number of round.
2. The cipher key is expanded. The *expanded key* is a array of 4−byte words, where the first Nk words contain the cipher key.
3. Round keys are extracted from the expanded key, where the i^{th} round key consists of the $i^{th}Nb$ words.

The Round Function. The *round function* consists of four steps :

1. *Byte sub*, which are fixed byte substitution. In contrast with DES, only one s-box is used and the substitution is no linear. Thus, for instance the State matrix :
$$(a_{i,j}) = (8i + j - 9), \ for \ 1 \leq i \leq 32 \ \wedge \ 1 \leq j \leq 8 \tag{2.11}$$
It consists of an inverse operation in a finite field $GF(2^8)$, followed by an affine (invertible) transformation over $GF(2)$.
2. *Shift row*, which is a permutation of the bytes. The row j for $j = 2, 3, 4$ of the State matrix are shifted respectively $x_j = 1, 2, 3$ units to the right. Shift row introduces high diffusion over multiple rounds and interact with the next step.
3. *Mix column*, which are fixed linear combinations. Each linear combination over $GF(2^8)$ acts on 4 bytes and outputs 4 bytes.
4. *Round key addition*, which performs an exor with the round key.

We discuss now the security of AES. The S-Box is nearly perfect for resistance to differential cryptanalysis. Thereby, Rijndael design is sufficient to withstand differential and linear attacks. Moreover, the design of Rijndael practically eliminates the possibility of weak or semi-weak keys, that exist for DES, and the key schedule eliminates the possibility of equivalent keys.
For seven of more rounds, no attacks faster than brute-force attack has been found due to the diffusion and non-linearity of Rijndael's Key Schedule and the complicated construction of the S-Box [Mol07].

We conclude symmetric ciphers with some rules of block cipher utilization. Untill now, we have seen how to encrypt a block with a given size. However, in practice, there is no guarantee that :

1. The plaintext length is a multiple of the blocklength, and
2. The last block will be completly full.

For, we must use padding to complete the last block. Note that several criteria can influence the padding. The first is to ensure its reversibility. For example, it will be easy to decipher if we add a 1, then many 0, that simply many 0 to fill the block. In addition, to check the integrity, it may be necessary to add the length of the message at the end of the plaintext to avoid collision. Finally, in the case of stream ciphers, padding may be used to hide the plaintext length [Jac12].

2.3.3 Public-Key / Asymmetric Ciphers

The concept of *public-key* or *asymmetric encryption* was previously introduced. Based on number theory, it was invented by Whitfield Diffie and Martin Hellman [DH76] and independently by Merkle [Dif88]. Public-key ciphers were presented with a novel property that publicly revealing the encryption key k does not thereby reveal the corresponding decryption key k' (i.e., computationally infeasible to determine k' from k). Thereby, the method of enciphering is a *one-way function* that cannot be reversed, and where the recipient needs additional information, called *trapdoor*, to decrypt the ciphertext.

The public-key cryptosystem depicted in Figure 2.7 is formally defined as follows :

Definition 2.15. (Public-Key Ciphers) [Mol07]
A cryptosystem is called asymmetric, or public key, if for each key pair (k, k'), the enciphering key k, called the public key, is made publicly available, whereas the deciphering key k', called the private key, is kept secret. In addition, the cipher must satisfy the one-wayness property.

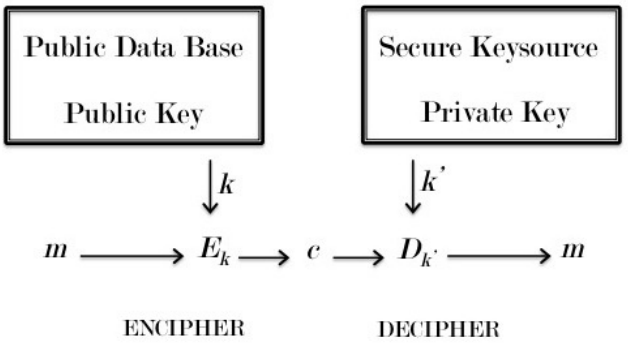

FIGURE 2.7: A generic public-key cipher.

Public-key ciphers can be used to encrypt plaintext or to verify a digital signature[RSA78] ; whereas the private key is used for the opposite operations, to decrypt ciphertext or to create a digital signature. Up to the time of this idea, all ciphers, including DES, were looking for mechanisms to securely distribute secret key. Now, with the introduction of the *Diffies-Hellman Key-Exchange*, entities could exchange keys in an open and ensure confidentiality.

TABLE 2.4: Public-key and secret-key ciphers - A comparison

	Public-key cipher	Secret-key cipher
Security	The private key needs to be kept secret by only one entity. The public key may be distributed. No cipher has been proven secure.	The secret key must be securely shared between entities. No cipher has been proven secure.
longevity	$[1, 2]$ years according to the NIST [oST12].	≤ 2 years according to the NIST [oST12].
Key exchange	No key exchange is required	A risky key exchange is required.
Key management	For a large network of $n \in \mathbb{N}$ entities, n key pairs are required.	For a large network of $n \in \mathbb{N}$ entities, $n \times \frac{(n-1)}{2}$ key pairs are required.
Efficiency	Slow.	Fast.
Keylength	≥ 1024 bits [LV01].	≥ 128 bits [LV01].
Nonrepudation	Ensured using digital signatures.	A trusted third party is needed. Ensure only confidentiality.

A quick comparison between public-key and secret-key ciphers is given in the Table 2.4. The legitimate question that we might ask is why we should use a public-key cipher to securely exchange secret keys rather then directly use a public key cipher to encrypt plaintext ? The principal reason has to do with efficiency [Mol07] ! As shown in Table 2.4,

public-key ciphers are extremely slow compared with symmetric-key ciphers (thousand times slower than). Thus, public-key ciphers are not meant to replace symmetric-key ciphers but rather to supplement them for achieving maximum security and efficiency.

Both, public-key and secret-key ciphers, come to be used, in concert, to create *hybrid* ciphers or *digital envelopes*. In such ciphers, the public-key is used only to exchange *session keys*, which are symmectric-keys generated for each new session and used to encrypt plaintext. In the following, we will explain some asymmetric encryption schemes.

2.3.3.1 Discrete Logarithm

The security of a cryptosystem depends upon the difficulty of solving mathematical problems. A *discrete logarithm problem* (DLP) or simply *discrete log* deals with finding k from $c = \langle m \rangle = m^k | k \in \mathbb{N}$, in the cyclic group m, and is denoted as $k = \log_m(c)$. It must be hard to find this k. Formally,

Definition 2.16. (Discrete Logarithm Problem)
Given a prime p, a generator m of $\mathbb{F}_p^*$, and an element $c \in \mathbb{F}_p^*$, find the unique integer k with $0 \leq k \leq p-2$ such that $c \equiv m^k (mod\ p)$.

Here, $k \equiv \log_m(c) (mod\ p-1)$ and if p is properly chosen, then it is a very difficult problem to solve. In fact, the complexity to find k when p has n digits is roughly the same as factoring an n-digit number. When n is very large, no efficient, non-quantum algorithm is known for the *integer factoring problem*. In 2009, an effort by several researchers concluded that factoring a 232-digit number (RSA-768), using hundreds of machines, takes over two years [KAF+10]. Hence, ciphers based upon the DLP are assumed to be secure. Here, we discuss the ElGamal encryption scheme.

ElGamal Encryption Scheme

In 1984, ElGamal announced a public-key scheme based on DLP [Gam84, Gam85]. The global elements of ElGamal scheme are the prime number p and a, which is a primitive root of p. For simplicity, we assume that p and a are public. When an entity A wants to generate its public and private keys, it chooses a uniform random $X_A \in \mathbb{Z}_p$, such that $1 < X_A < p-1$, then computes $Y_A = a^{X_A} mod\ p$, and makes it public.

Encryption. To encrypt a plaintext m, an entity B, that has access to the public key (Y_A, p, a), chooses a uniform random $k \in \mathbb{Z}_p$, such that $1 < k < p-1$. A one-time key K is computed as :

$$K = (Y_A)^k mod\ p \tag{2.12}$$

Finally, the ciphertext c is computed as a pair (c_1, c_2), where :

$$c = (c_1, c_2) = (a^k mod\ p, m \times K\ mod\ p) \tag{2.13}$$

Decryption. To decrypt the ciphertext $c = (c_1, c_2)$, the entity A, knowing the secret key X_A, compute the key :

$$K = (c_1)^{X_A} mod\ p \tag{2.14}$$

then the plaintext as :

$$m' = (c_2 \times K^{-1}) mod\ p \tag{2.15}$$

We discuss now the security of ElGamal. As explained before, ElGamal is based on the difficulty of computing discrete logarithm. To recover the private key of the entity A, an adversary would have to compute $X_A = \log_{a,p}(Y_A)$. Alternatively, to recover the one-time key K, an adversary would have to determine the random number k, and this would require computing the discrete logarithm $k = \log_{a,p}(c_1)$.
In [Sti95], it is pointed out that these calculations are regarded as infeasible if p is at least 300 decimal digits and $p - 1$ has at least one *large* prime factor.

2.3.3.2 RSA

RSA is an acronym for the inventors of the scheme : Rivest, Shamir and Adleman [RSA78]. Basically, RSA is a heuristic cryptosystem that applies the Euler-Fermat theorem [Des09b]. The scheme is a block cipher in which the plaintext m and ciphertext c are integers between 0 and $n - 1$ for some $n = 1024$ bits or 309 decimal digits. The RSA algorithm involves three steps : key generation, encryption and decryption.

RSA Key Generation

- An entity A generates two large, random primes $p \neq q$, then computes two integers $n = p \times q$, called the RSA modulus, and $\varnothing(n) = (p - 1) \times (q - 1)$.
- The entity A selects a random $e \in \mathbb{N}$, called the RSA enciphering exponent, such that $1 < e < \varnothing(n)$, where $gcd(e, \varnothing(n)) = 1$.
- The entity A computes the unique $d \in \mathbb{N}$, called the RSA deciphering exponent, such that $1 < d < \varnothing(n)$, where $d \equiv e^{-1} (mod\ \varnothing(n))$.
- The entity A publishes (n, e) as RSA public key and keep d as RSA secret private key. Note that p, q, and $\varnothing(n)$ need to remain secret.

RSA Enciphering

- To encrypt a message m in numerical form with $m < n$, an entity B obtains the RSA public key of the entity A, which we called (n, e).
- The entity B enciphers m by computing $c \equiv m^e (mod\ n)$.

RSA Deciphering

- To decrypt a ciphertext c, the legitimate receiver, let us say entity A, knowing the secret key d.
- The entity A deciphers c by computing $m' \equiv c^d (mod\ n)$.

We cannot encipher a plaintext message if it is a numerical value $m \geq n$. In this case, we must subdivide the plaintext message into blocks of equal size. This process is called *message blocking*. The plaintext message is writing as blocks of l-digits, where $N^l < n$ and N is the base, then are enciphered separately.

Nowadays, an RSA modulus of 1024 to 4096 bits would be considered secure [Mol07]. To speed up encryption, it has been suggested to choose $e = 3$ or a small e. When m is chosen as a uniformly random element, and p, q are large enough, no attacks are known for finding m. It has been argued that this is as hard as factoring n (without proof) [Des09b].

2.3.4 Introduction to Homomorphic Encryption

Rivest *et al.* [RAD] were the first to pose the problem of making operations on encrypted data. Indeed, one of the basic limitations of encryption is that an information system working with encrypted data can at most store or retrieve the data for the user ; any more complicated operations seem to require that the data be decrypted before being operated on.

There exists a third kind of encryption schemes, refered as *homomorphic encryption*, that allows us to bypass this limitation and compute on ciphertexts, generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. We distinguish two types of homomorphic encryption schemes : *fully* and *partially* homomorphic encryption schemes. All of them are malleables by default (Definition 2.5).

Partially homomorphic encryption (PHE) schemes usually allow only one type of homomorphic operation. Table 2.5 shows several PHE schemes and homomorphic operations allowed.

TABLE 2.5: Partially homomorphic encryption schemes

Schemes	Operations allowed
Unpadded RSA	multiplication of two messages modulo n.
ElGamal	multiplication of two messages.
Boneh-Goh-Nissim	a random number of additions and a single multiplication.
Naccache-Stern (generalization of Benaloh scheme [Hen08])	addition and the multiplication by a constant.
Damgard-Jurik	addition and multiplication by a constant.
Paillier (a special case of Damgard-Jurik)	addition and multiplication by a constant.

Fully homomorphic encryption (FHE) scheme, introduced by Craig Gentry [Gen09a], allows us to compute *arbitrary* functions over encrypted data without the decryption key, i.e., given ciphertexts $c_1, c_2, \ldots, c_n$ of plaintexts $m_1, m_2, \ldots, m_n$, one can efficiently compute a compact ciphertext that encrypts $f(m_1, m_2, \ldots, m_n)$ for any efficiently computable function f.

Basically, Gentry' scheme ε has four polynomial algorithms :

Key generation $\text{KeyGen}_\varepsilon(\lambda)$ outputs a key-pair (sk, pk) using a security parameter λ.

Message encryption $\text{Encrypt}_\varepsilon$ takes pk and a plaintext $m \in \mathcal{M}$ as input, and outputs a ciphertext $c \in \mathcal{C}$.

Message decryption $\text{Decrypt}_\varepsilon$ takes sk and a ciphertext c as input, and outputs the plaintext m.

Homomorphic evaluation $\text{Evaluate}_\varepsilon$, that takes as input the public key pk, a circuit (function) $F \in \mathcal{F}_\varepsilon$ fom a permitted set of circuits (functions) and a tuple of ciphertexts $\Psi = \langle c_1, c_2, \ldots, c_t \rangle$ for the input wires of F ; it outputs a ciphertext c. Informally, if c_i encrypts m_i under pk, then $\text{Evaluate}_\varepsilon(pk, F, \Psi) \rightarrow c$ encrypts $F(m_1, m_2, \ldots, m_t)$ under pk, where $F(m_1, m_2, \ldots, m_t)$ is the output of F on inputs $m_1, m_2, \ldots, m_t$.

There are different ways of formalizing the functionality $\text{Encrypt}_\varepsilon(m_1, m_2, \ldots, m_t)$. A minimal requirement is *correctness* given in Definition 2.13.

Definition 2.17. (Correctness of Homomorphic Encryption) [Gen09a]
We say that a homomorphic encryption scheme ε is correct for circuits in $\mathcal{F}_\varepsilon$ if, for any key-pair (sk, pk) outputs by $\text{KeyGen}_\varepsilon(\lambda)$, any circuit $F \in \mathcal{F}_\varepsilon$, any plaintexts $m_1, m_2, \ldots, m_t$, and any ciphertexts $\Psi = \langle c_1, c_2, \ldots, c_t \rangle$ with $\text{Encrypt}_\varepsilon(pk, m_i) \rightarrow c_i$, it is the case that :

$$if\, c \leftarrow \text{Evaluate}_\varepsilon(pk, F, \Psi), then \text{Decrypt}_\varepsilon(sk, c) \rightarrow F(m_1, m_2, \ldots, m_t) \tag{2.16}$$

The first Gentry' scheme, presented in [Gen09b], was able to evaluate an arbitrary number of additions and multiplications on encrypted data using lattice-based cryptography. This scheme was rather theoretical than implementable. A new version, presented in [vDGHV10], used integers instead of lattices. Despite improvements proposed recently, all algorithms of this kind of scheme are computationally demanding and remain too slow to be used in the context of large databases or data stream.

2.4 Theory of Anonymity

Historically, *anonymity* focused on protecting author's privacy in the context of philosophical and political publications. Technological advances of the last few decades allow data to be easily collected, stored and analyzed by organizations in ways that were impossible in the past. Thereby, huge data collections can be analyzed using powerful data mining techniques to discover new knowledges [AW89, Klö95, CdVFS08]. In the same time, sophisticated algorithms have made possible *linking attacks*, combining data available through different sources to infer sensitive information. In this context, anonymity was introduced as a security mechanism to adress privacy concerns.

First we will settle upon the meaning of anonymity, which is a state or quality of being *anonymous*. Anonymous has, as its etymology, *an* from the Greek, meanning *not*, *onym*, meaning *name*, and *-ous*, meaning *possessing*, and means not named or unsigned. In IT, anonymity is the study of methods and algorithms for publishing and treating information in an anonymous form. The original data, generally saved in a relational table or database, is called the *entire or private dataset* and the modified data released is called the *anonymized or released dataset*. The process of transforming original dataset into anonymized dataset is called *anonymizing*. Like hash function, anonymizing process is one-way and does not have reverse process. The main objective for anonymizing dataset is to protect privacy aspects against adversary who tries to disclosure private data, such as medical data collected during hospital treatment.

2.4.1 Towards Anonymity

Anonymous publications were common in the seventeenth century. In [Cat82], it is noted that only one author of the six sets of objections initially published with the Meditations [5] was named ; and one of those anonymous authors, Thomas Hobbes, had at about that time circulated anonymously his Elements of Laws.

More examples of this practice among writers, scholars, theologians, philosophers, scientists, and politicians could easily be multiplied. Generally, authors published anonymous works to avoid censorships, polemical contests or political crisis. For instance, in June 8, 1637, René Descartes published anonymously his famous *Discours de la méthode*, written a few years after the trial of Galileo in June 1633, which had been condemned by the Church.

Within today's global infrastructure, entities and users interact with remote servers and databases for retrieving data or for using online services. In the same time, safeguarding privacy and human identity is a right established by national laws (French Data Protection Act[6]), and international treaties (European Convention[7] and United Nations Resolution[8]). In such a context, Ciriani *et al.* [CdVFS09] noted that privacy involves three different but related concepts, as following :

Privacy of the user. It concerns protecting the identity of entities, that communicate through networks, to avoid possible attacks regarding the relationships between them. Anonymizing the communication layer is thus a necessary measure to protect the privacy of users, and computer systems against traffic analysis. Anonymous communications were firstly established in 1981 by David Chaum, and implemented using different methods. We can mention, *Mix networks* [Cha81], *Onion routing* [SGR97], *Tor* [DMS04], and *Crowds* [RR99]. With the exception of the mix networks, all others methods are based on encryption schemes.

Privacy of the communication. It concerns protecting the confidentiality of information sent through a network ; and the content of requests. Regarding the first issue, i.e., protecting the confidentiality of information, we have previously discussed (in Section 2.3) some encryption schemes for this purpose. As regards the

[5] Meditations on First Philosophy is a philosophical treatise, first published in 1641, that consists of the presentation of René Descartes' metaphysical system.

[6] Article 1 - Loi 78-17 du 6 janvier 1978 modifiée.

[7] Article 8 (Right to respect for private and family life) - European Convention on Human Rights entered into force on 3 September 1953.

[8] Article 17 - International Covenant on Civil and Political Rights, adopted and opened for signature, ratification and accession by General Assembly resolution 2200A (XXI) of 16 December 1966.

latter issue of protecting the request content, known as *Private Information Retrieval problem* (PIR), it consists in safely querying remote databases. A naive solution is to completely downloading the remote database.

In [CKGS98], the authors prove that if we have only one copy of the database in the server side, then there is no solution that is better than the naive one. However, if we have m copies of the database, we can submit m independent requests, and the m results are then combined to have the final result [Amb97, CKGS98, IK99]. Solutions based on ciphers [CG97, KO97, CMS99], or *Secure Multiparty Computation* (SMC) [DA01] were also proposed.

Privacy of the information. It is related to the development of methods for ensuring proper data protection and anonymity of persons and entities. Anonymity implies that released information be *nonidentifiable*. Given, for instance, a set of personal information about a hospital patient p, such as, *social security number (SSN), name, gender, date of birth, ZIP code*, and *disease*. The identity of p is protected if the value allowing its identification is kept private (here the attribute *name*). We call this process *de-identification*. Note that a subset of personal information, such as gender, date of birth and ZIP code, can be linked with external information to identify p [Gol06]. Additionally, p can be identified by his SSN. However, with the absence of information that associate the SSN to a name, p is still anonymous. Thereby, de-identification is not sufficient to guarantee the *identity disclosure protection*.

Now, imagine that the identity of p is well protected. Moreover, p belongs to a group or table that could have the same sensitive information (e.g., disease). The identity disclosure protection of p alone will not guarantee the protection of his sensitive information. For that, we should provide additional mechanisms to guarantee the *attribute disclosure protection*.

In the remainder of this section, we will describe security mechanisms provided to protect identity and attribute disclosure.

2.4.2 k-anonymity

Knowledge discovery in databases (KDD) is the search for patterns that exist in datasets, but are hidden among the volumes of data [Klö95]. KDD becomes nowadays the centerpiece in business management, public institutions and government (e.g., insurance, finance, and health). Knowledge discovery process involves different but related stages,

such as *data collection and extraction*, *data preparation and transformation*, *data cleaning*, *data mining*, and *reporting*. Statistics and reports are usually *disseminated* and *shared* within the organization collecting it and with other organizations.

Data released could be to satisfy legal requirements or as part of some business process. Indeed, security risks and protection regulations are relevant, and should be adressed. An important issue, regarding current laws, is the protection of the privacy of individuals or entities (i.e., *respondents*) to whom the data refer [FW72, DL86]. For that purpose, specific *data protection norms* and appropriate *safeguards* must be applied before releasing information. These appropriate safeguards depend on the method in which data are released [CdVFS09]. We distinguish :

Macrodata , which are statistics on users or entities presented as statistical databases or two-dimentional tables.

Microdata , which are data containing structured information on individuals like persons, entities, and transactions.

In the past, information was principally released as macrodata (tabular and statistical form). Security-control methods for macrodata are generally based on selective obfuscation of sensitive cells. Adam *et al.* [AW89] classified them into four general categories : *conceptual*, *query restriction*, *data perturbation*, and *output perturbation* [Den83, DS83].

Nowadays, many situations call for the release of microdata. In fact, in contrast to macrodata that report precomputed statistics, microdata provide the convenience of allowing the final recipient to perform analysis as needed. Then, the protection of microdata against improper disclosure is therefore an issue that has become increasingly important and will continue to be so [Sam01, Iye02].

2.4.2.1 Problem Statement

Table 2.6 depicts an example a de-identified (medical) microdata over a set of attributes : *SSN*, *name*, *date of birth*, *gender*, *ZIP code*, *marital status*, and *disease*. Microdata was de-identified using a naive approach. It consists in simply deleting values corresponding to both attributes *name* and *SSN*, to not explicitly disclose the identities of respondents in the table. Therefore, the attributes (or columns) of the Table 2.6 can be classified as follows :

Identifier attribute (or unique identifier) is any attribute that uniquely identifies a respondent. Unique identifiers are typically removed entirely from released microdata (e.g., the attributes *name* and *SSN* in Table 2.6).

TABLE 2.6: De-identified (Medical) private microdata

SSN	Name	Date of birth	Gender	ZIP	Marital status	Disease
		02/04/1978	M	77430	divorced	hypertension
		03/09/1978	M	77420	divorced	obesity
		05/04/1978	M	77410	married	chest pain
		03/03/1977	F	77410	married	obesity
		08/03/1977	F	77410	married	short breath
		17/07/1984	M	77400	single	short breath
		17/07/1984	M	77410	single	obesity
		17/07/1984	M	77410	single	chest pain
		17/07/1984	M	77420	widower	short breath

Quasi-identifier , denoted QI, is a minimal set of attributes that can be linked with external datasets to reduce the uncertainty over respondents' identities. For instance, consider the public Voter List illustrated in Table 2.7. The attributes *date of birth*, *gender*, *ZIP code*, and *marital status* of PT can be linked to the Voter List to reveal sensitive information (e.g., disease) that refer to M. Durant. We assume that QI is recognized based on knowledge of the domain.

Confidential attribute contain sensitive information, such as *disease*. An adversary should not be able to uniquely associate its value with a unique identifier.

Nonconfidential attribute does not fall into any of the categories above.

TABLE 2.7: Non de-identified (Voter List) public microdata

Name	Gender	DoB	Adress	City	ZIP	Status
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
Durant M.	M	02/04/1978	Ch. de Samois	Champagne/Seine	77430	divorced
...	...	...	...	...	...	...
...	...	...	...	...	...	...

One approach to reduce re-identification risk is to perturb the microdata using techniques like *adding noise* and *swapping values* while ensuring that some statistical properties of the entire table are maintained [KW95]. The tradeoff between information loss, called *data quality*, and the re-identification risk using such methods is being actively researched [YWC02].

An alternative approach is to transform the dataset by using generalizations and suppressions. Several works have explored this approach (e.g., [Sam01, Iye02, Swe02, JA05, LDR05, FWY05, GTK+05, AFK+05, MW04]), and will be discussed later. An example of a transformation by generalization is to replace the exact *date of birth* in Table 2.6

by only the year of the birth. Suppressions can be seen as ultimate generalizations since no information is released.

The principal challenge is to find the *right* tradeoff between the amount of privacy and loss of information content (i.e., data quality) due to adding noise, swapping values, or data transformation. In such context, *k-anonymity* has been therefore proposed as an approach to protect respondents' identities while releasing **truthful** information [Sam01].

2.4.2.2 Formal Definitions

We formally define some key terms, which will be used in the dissertation.

Definition 2.18. (Relational table) [Sam01]
Let $\mathbb{A}$ a set of attribute, $\mathbb{D}$ be a set of domains, and $dom : \mathbb{A} \rightarrow \mathbb{D}$ be a function associating each attribute with its domain.
A relational table T over a finite set $\{A_1, \ldots, A_p\} \subseteq \mathbb{A}$ of attributes, denoted $T(A_1, \ldots, A_p)$, is a set of tuples over the set $\{A_1, \ldots, A_p\}$ of attributes, where :

- $dom(A, T)$ denotes the domain of attribute A in T.
- $\mid T \mid$ denotes the number of tuples in T.
- $t\,[A]$ represents the value v associated with A in t.
- $t\,[A_1, \ldots, A_p]$ represents the subtuple of t containing the values of attributes $\{A_1, \ldots, A_p\}$.
- $T\,[A_1, \ldots, A_p]$ represents the subtuples of T containing the values of attributes $\{A_1, \ldots, A_p\}$ (i.e., the projection of T over $\{A_1, \ldots, A_p\}$).

Definition 2.19. (k-anonymity requirement) [Sam01]
Each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k respondents.

The k-anonymity requirement is quite simple. It assumes that the data owner knows how many respondents each released tuple matches. For that purpose, linking released data with external datasets is necessary. However, data owner usually ignores external information. Consequently, k-anonymity requirement stipulates that no individuals record should be uniquely identifiable from a group of k on the basis of its QI values. Thus, the k-anonymity definition requires each respondent to be *indistinguishable* with respect to at least other $k-1$ respondents in the released table. Formally,

Definition 2.20. (k-anonymity) [Sam01]
Let $T(A_1, \ldots, A_p)$ be a table, and QI be a quasi-identifier associated with it. T is said to satisfy k-anonymity with respect to QI iff each sequence of values in $T[QI]$ appears at least with k occurences in $T[QI]$.

The second problem to which may face a data owner is identifying the set of quasi-identifiers QI. For instance, Table 2.6 is 1-anonymous w.r.t $QI = \{ZIP\}$ and $QI = \{ZIP, gender, dateofbirth\}$. In the same time, the table is 2-anonymous w.r.t $QI = \{gender\}$. Consequently, to correctly enforce k-anonymity, it is necessary to clearly identify QI [CdVFS09].

2.4.2.3 Generalization and Suppression

Basically, each attribute A_i in the Table T is associated with a *groud domain* $D = dom(A_i, T)$. An attribute generalization (AG for short) consists in a substitution of all values v of the attribute A_i : $A_i \in QI$, with a more general value $\acute{v} \in \acute{D}$, where $\acute{D}$ is a generalized domain for D, denoted $D \leq_D \acute{D}$, and $\acute{v}$ is a generalized value for v, denoted $v \leq_D \acute{v}$. Note that cell generalization (CG for short) is performed on individual cells.

As depicted in Figure 2.8, the generalization relationship implies the existence, for each domain D, :

- A totally ordered hierarchy, called *domain generalization hierarchy*, denoted DGH_D, and represented using a simple path ; and
- A *value generalization hierarchy*, denoted VGH_D, and represented using a tree

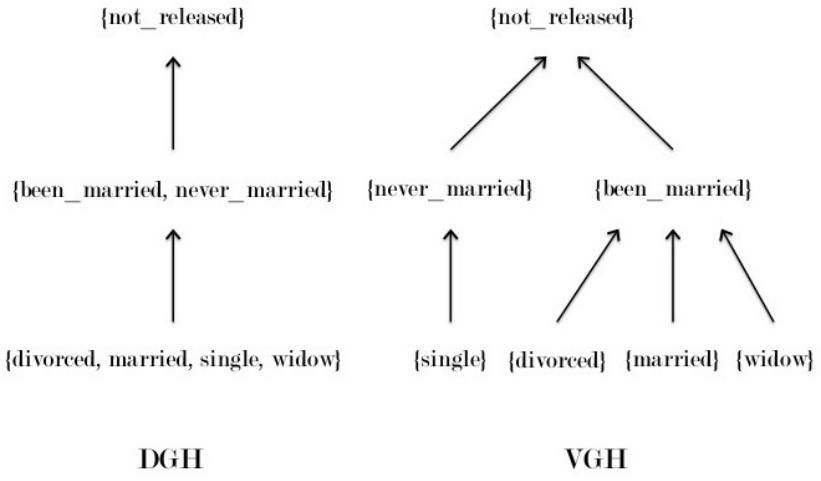

FIGURE 2.8: Generalization hierarchy for the marital status.

Generalizations may be performed using two approachs : *Hierarchy-based generalization* and *Recoding-based generalization* [CdVFS09]. A comparison between them is given in Table 2.8.

TABLE 2.8: Hierarchy & Recoding-based generalization - A comparison

Hierarchy-based generalization	Recoding-based generalization
Based on the definition of a generalization hierarchy.	Based on the recoding into intervals.
For each attribute in QI, the most general value is at the root and the leaves correspond to the most specific values.	For each attribute in QI, the ground domain is partitioned into possibly disjoint (labeled) intervals.
Values are mapped with one of their ancestor vertices.	Values are mapped with the intervals they belong to.
Hierarchy must be predefined.	Intervals are computed at the runtime.

Suppression consists in removing from the private table a cell (CS for short), an attribute (AS for short), or a tuple (TS for short). Suppressions are combined to generalizations to reduce the amount of generalization necessary to satisfy k-anonymity requirement. Formally,

Definition 2.21. (Generalized table with suppression) [Sam01]
Let T_i and T_j be two tables defined on the same set of attributes. Table T_j is said to be a generalization (with tuple suppression) of table T_i, denoted $T_i \preceq T_j$, if :

1. $\mid T_j \mid \leq \mid T_i \mid$;
2. the domain $dom(A, T_j)$ of each attribute A in T_j is equal to, or a generalization of, the domain $dom(A, T_i)$ of attribute A in T_i.
3. it is possible to define an injective function associating each tuple t_j in T_j with a tuple t_i in T_i, such that the value of each attribute in t_i is equal to, or a generalization of, the value of the corresponding attribute in t_i.

The distance vector of T_j from T_i is the vector $DV_{i,j} = [d_1, \ldots d_n]$, where each $d_z, z = 1, \ldots, n$ is the length of the unique path between $dom(A_z, T_i)$ and $dom(A_z, T_j)$ in DGH_{D_z}.

Using generalization and suppression may produce one or more anonymized tables, which are *more general* (less precises) and *less complete* (due to tuples suppression). Therefore, the main objective is to maintain as much information as possible (i.e., the minimality of the solution should be guaranteed) under the k-anonymity constraint. For that, we should be able to quantify genaralizations and limit suppressions. The concept of *k-minimal generalization* was introduced in [Sam01]. k-minimal generalization

uses *distance vector* between two tables and *hierarchy of distance vectors* to quantify generalization ; and a threshold, denoted $MaxSup$, specifying the maximum number of tuples that can be deleted.

2.4.2.4 Algorithms for k-anonymity

k-anonymizing private tables by exploiting generalization and suppression has been widely studied and a number of approaches have been proposed. Le Fevre *et al.* [LDR05] have described the first taxonomy for classifying k-anonymity approaches, where suppression and generalization are applied at the cell and attribute levels. Later, Ciriani *et al.* [CdVFS07] refined and completed it. In Ciriani *et al.* taxonomy, generalization and suppression are applied at different granularity levels (i.e., AG, CG, CS, AS, and TS).

TABLE 2.9: Classification of k-anonymity techniques

Generalization	**Suppression**			
	Tuple	Attribute	Cell	None
Attribute	AG_TS	$AG\_AS \equiv AG\_$	AG_CS	$AG\_ \equiv AG\_AS$
Cell	N/A	N/A	$CG\_CS \equiv CG\_$	$CG\_ \equiv CG\_CS$
None	$\_TS$	$\_AS$	$\_CS$	

Table 2.9 summarizes different combinations at all possible granularity levels according to [CdVFS07]. We refer to each model with a pair separated by _ . The first element describes the level of generalization (AG, CG, or none) and the second element describes the level of suppression(TS, AS, CS, or none).

In Table 2.10, we describe some algorithms investigated in the literature. The majority of them are based on AG_TS , i.e., Generalization of attribute (column) and suppression of tuple (row). This is due to the assumption considered in the original model proposed in [Sam01]. The k-anonymity problem is NP-hard for $k \geqslant 3$. Then, subsequent approaches provide efficient algorithms for solving the k-anonymity problem to enhance data quality and to reduce computational complexity.

All exact algorithms have computational time exponential in the number of the attributes composing the quasi-identifier QI. In fact, Sweeney' algorithm [Swe02] exhaustively examines all potential generalizations for identifying a minimal one satisfying the k-anonymity requirement, which is clearly impractical for large datasets. Samarati' algorithm [Sam01] exploits a binary search on the DGH to avoid an exhaustive visit of the whole generalization space. Bayardo and Agrawal algorithm [JA05] exploits ad-hoc pruning techniques to specialize a fully generalized table (with all tuples equal) into a minimal k-anonymous table. Finally, Le Fevre *et al.* algorithm (Incognito) [LDR05] uses a bottom-up technique and a priori computation.

TABLE 2.10: Algorithms for k-anonymity

Model	Equiv	Class	Algorithm	Type	Complexity
AG_TS		NP-hard	Samarati[Sam01]	Exact	$e^{\|QI\|}$
			Sweeney[Swe02]	Exact	$e^{\|QI\|}$
			Bayardo *et al.*[JA05]	Exact	$e^{\|QI\|}$
			LeFevre *et al.*[LDR05]	Exact	$e^{\|QI\|}$
			Iyengar[Iye02]	Heuristic	limit. itera.
			Winkler[Win02]	Heuristic	limit. itera.
AG_AS	AG_	NP-hard	No investigated		
AG_CS			No investigated		
AG_	AG_AS	NP-hard	Fung *et al.*[FWY05]	Heuristic	limit. itera.
CG_CS	CG_	NP-hard	No investigated		
CG_	CG_CS	NP-hard	Aggarwal *et al.*[GTK+05]	$O(k)$-approx	$O(kn^2)$
_TS	AG_TS	polynomial	No investigated		
_AS	AG_	NP-hard[MW04]	No investigated		
_CS	AG_	NP-hard[GTK+05]	Aggarwal *et al.*[AFK+05]	$O(k)$-appro	$O(kn^2)$
			Meyerson *et al.*[MW04]	$O(klogk)$-approx	$O(n^{2k})$

Heuristic algorithms were also explored. Iyengar' algorithm [Iye02] used genetic algorithms to solve the k-anonymity problem using an incomplete stochastic search method. Fung *et al.* [FWY05] presented a top-down heuristic and Winkler [Win04] proposed a method based on simulated annealing for finding locally minimal solutions.

2.4.3 ℓ-diversity

We previously stated that k-anonymity has been provided as a security mechanism to the problem of identity disclosure. Since it protects individuals and reduces uncertainty about their identities by making each record indistinguishable from at least $k-1$ other records, k-anonymity does not protect from attribute disclosure. We now describe some attacks against k-anonymity, then present a new security mechanism, i.e., *ℓ-diversity*, that completes the k-anonymity concept and permits protecting from attribute disclosure.

2.4.3.1 Attacks on k-anonymity

Table 2.11 shows a 3-*anonymous* medical microdata from a fictitious parisian hospital. Note that identifier attributes *SSN* and *Name* were deleted, and quasi-identifier attributes *ZIP Code*, *Marital status*, and *Gender* were generalized, to protect patients' identities. 3-anonymous table means each tuple has the same values for the *QI* attributes as at least two other tuples in the table. The confidential attribute values, i.e., diseases, must not be discovered by an adversary for any individual in the microdata.

TABLE 2.11: 3-anonymous (Medical) microdata

	Identifier		Quasi-Identifier			Confidential
	SSN	Name	ZIP Code	Marital status	Gender	Disease
1			7500∗	been_married	F	hypertension
2			7500∗	been_married	F	hypertension
3			7500∗	never_married	M	obesity
4			7500∗	never_married	M	cancer
5			7500∗	never_married	M	obesity
6			7500∗	been_married	F	hypertension
7			7501∗	been_married	M	obesity
8			7501∗	been_married	M	cancer
9			7501∗	been_married	M	cancer

Machanavajjhala *et al.* [MGKV06] defined two attacks against k-anonymity : *homogeneity attacks* and *background knowledge attacks*.

Homogeneity attacks. Note that tuples of Table 2.11 comprise three distinct groups : $(1, 2, 6)$, $(3, 4, 5)$, and $(7, 8, 9)$. If the tuples in a given group, which share a specific value for the QI, have the same confidential attribute value, then an adversary can infer which is value for this confidential attribute (here *disease*) for the known respondent.

For instance, *Eve* knows that her friend, named *Alice*, is divroced and living in Paris with ZIP code 75006. Therefore, *Eve* kowns that *Alice*' tuple number is 1, 2, or 6, and can infer that *Alice* suffers from *hypertension*. So, k-anonymity can create groups that leak information due to lack of *diversity* in the confidential (sensitive) attribute.

Background knowledge attacks. Background knowledge attack is based on a priori knowledge of the adversary of some additional external information. For instance, *Alice*' neighbor, named *Bob*, got sick and was taken by ambulance to the same hospital. *Alice* knows that *Bob* is single, male, and lives in her area. Additionally, *Alice* knows that *Bob* is *thin*. Consequently, *Alice* kowns that *Bob*' tuple number is 3, 4, or 5, and based on her a priori knowledge *"Bob is thin"*, *Alice* can infer that *Bob* suffers from *cancer*. So, k-anonymity does not protect against attacks based on background knowledge.

2.4.3.2 ℓ-diversity Principle

Before defining ℓ-diversity, Machanavajjhala *et al.* [MGKV06] have modelized the background knowledge of an adversary as a probability distribution over the attributes. For

that, they started by quantifying adversary' *prior* and *posterior believes* α and β, respectively. Then, they introduced the notions of *positive* and *negative disclosure*. Formally, given $\delta > 0$:

- There is a positive disclosure if $\beta_{(q,c,T)} > 1 - \delta$ and $\exists t \in T \ : t\,[QI] = q \wedge t\,[C] = c$. For instance, in the homogeneity attack where *Eve* infer that *Alice* suffers from hypertension is a positive disclosure.
- There is a negative disclosure if $\beta_{(q,c,T)} < \delta$ and $\exists t \in T \ : t\,[QI] = q \wedge t\,[C] \neq c$. For instance, in the background attack where *Alice* can deduce that *Bob* does not have hypertension is a negative disclosure.

Finally, they gave a first principle of privacy : *"The published table should provide the adversary with little additional information beyond the background knowledge"*.
Due to the difficulties for probabilistically modeling the knowledge, and the ignorance of the degree of knowledge of an adversary, a second principle was introduced to face lack of diversity of anonymized tables, and strong background knowledge of adversaries. Given q-block a set of tuples in T having the same value for QI : *"a q-block is ℓ-diverse if contains at least ℓ "well-represented" values for the sensitive attribute C*. Therefore ℓ-diversity can be defined as follows :

Definition 2.22. (ℓ-diversity) [MGKV06]
Let $T(A_1, \ldots A_n, C)$ be a table, $QI = \{A_1, \ldots A_n\}$ be its quasi-identifier, and C a confidential attribute. Let ℓ be a threshold defined by a user. T is said to be ℓ-diverse if all its q-blocks are ℓ-diverse.

After the introduction of ℓ-diversity, some variants have been studied in [MGKV06, TV06, WLFW06]. *Recursive* (c, ℓ)*-diversity* is a less conservatrice instantiation of ℓ-diversity, which has been developed in the case of one value of the confidential attribute is very common. Thus for $\ell > 2$, a q-block satisfies (c, ℓ)-diversity if we can delete one possible confidential value in the q-block and still have a $(c, \ell - 1)$-diversity block [MGKV06]. *Multi-attribute ℓ-diversity* treats the case where more attributes are confidential [MGKV06]. As multi-attribute ℓ-diversity, Truta *et al.* [TV06] proposed *p-sensitive k-anonymity* that considers microdata with more than one sensitive attribute. Finally, Wong *et al.* [WLFW06] supposed that not all the values in the domain of a confidential attribute are equally sensitive. For that, they have proposed (α, k)*-anonymity*, which considers a threshold α for the relative frequency of values considered as sensitive.

2.4.3.3 Implementing ℓ-diversity

Machanavajjhala *et al.* [MGKV06] prove that ℓ-diversity satisfies the monotonicity property with respect to DGH. This means if T_i guarantees ℓ-diversity, then any T_j such that $T_i \preceq T_j$ satisfies ℓ-diversity. Therefore, generalization based algorithms introduced in Table 2.10 for k-anonymity can also be used to achieve ℓ-diversity by checking property evry time a table is tested for k-anonymity. Since ℓ-diversity is a property that is local to each q-block ; and since all ℓ-diversity tests are solely based on the counts of the sensitive values, the test can be performed very efficiently.

TABLE 2.12: The anatomized tables

TABLE 2.12.A The quasi-identifier table (QIT)

	ZIP Code	Marital status	Gender	Group-ID
1	75001	divorced	F	1
2	75002	married	F	1
3	75003	single	M	2
4 (Bob)	75006	single	M	2
5	75005	single	M	2
6 (Alice)	75006	divorced	F	1
7	75017	married	M	3
8	75018	widow	M	3
9	75019	divorced	M	3

TABLE 2.12.B The confidential table

Group-ID	Disease	Count
1	hypertension	3
2	obesity	2
2	cancer	1
3	obesity	1
3	cancer	2

Xiao and Tao [XT06] introduced a technique, called *anatomy*, which releases all the quasi-identifier and confidential attributes directly in two separate tables : quasi-identifier table (QIT) and confidential table. Figure 2.12 depicts an anatomized version of Table 2.11. Anatomy protects privacy and allows more effective aggregate analysis in the microdata than generalization. Furthermore, Nergiz *et al.* [NC11, NCM13] introduced operations to safely querying and updating anatomized tables.

2.4.4 t-closeness

Up to now, two kinds of information disclosure have been studied, and two security mechanisms, k-anonymity and ℓ-diversity, have been introduced. In this section, we

describe attacks against ℓ-diversity from attribute disclosure, then we present a security mechanism, i.e., *t-closeness*, that completes the ℓ-diversity and permits a better protecting against attacks from attribute disclosure.

2.4.4.1 Attacks on ℓ-diversity

Table 2.13 shows a 3-anonymous and 2-diverse medical microdata from a fictitious parisian hospital. Note that identifier attributes *SSN* and *Name* were deleted, and quasi-identifier attributes *ZIP Code*, *Marital status*, and *Gender* were generalized, to protect patients' identities. 3-anonymous table means each tuple has the same values for the QI attributes as at least two other tuples in the table. Confidential attribute C values, i.e., diabetes and salary, must not be discovered by an adversary for any individual in the microdata. 2-diverse table means each q-block has at least two different values for the C attributes.

TABLE 2.13: 3-anonymous and 2-diverse (Medical) microdata

	Identifier		Quasi-Identifier			Confidential	
	SSN	Name	ZIP Code	Marital status	Gender	Diabetes	Salary
1			7500∗	been_married	F	N	25.000
2			7500∗	been_married	F	N	26.000
3			7500∗	never_married	M	Y	60.000
4			7500∗	never_married	M	Y	35.000
5			7500∗	never_married	M	N	100.000
6			7500∗	been_married	F	N	25.500
7			7501∗	been_married	M	N	32.000
8			7501∗	been_married	M	Y	31.000
9			7501∗	been_married	M	N	30.000

Li *et al.* [LLV07] defined two possible attacks against ℓ-diversity : *Skewness attacks* and *Similarity attacks*.

Skewness attacks. In Table 2.13, the q-block $(3, 4, 5)$ having two out of three tuples with a positive value of the attribute *Diabetes* and only one tuple with a negative value. This presents a serious privacy risk, because anyone in the q-block would be considered to have 67% possibility of being positive, as compared with the 30% of the overall population.

Consider now a second q-block that has 49 positive value and only one negative value. This satisfies 2-diversity, and anyone would be considered to have 98% possibility of being positive, as compared with the 30% of the overall population.

Therefore, the two q-blocks have exactly the same diversity and present a very different levels of privacy risks So, ℓ-diversity can create groups that leak information when the overall distribution is *skewed*.

Similarity attacks. When the confidential attribute values in a q-block are distinct but semantically similar, an adversary can learn important information. For instance, it is easy to infer that single males living in the $7501*$ area have the salary value $\in [30.000, 32.000]$, since the tuples in the q-block $(3, 4, 5)$ have these values for the considered confidential attribute. So, ℓ-diversity does not protect against attacks based on semantical closeness.

2.4.4.2 t-closeness Principle

Li *et al.* [LLV07] introduced the notion of *indermediate believes*. Indeed, given α the prior belief and β the posterior belief. α can be influenced by D_T the distribution of the confidential attribute value in the whole population, i.e., table. This belief, before discovering the released table, is defined as an intermediate belief denoted δ. Since ℓ-diversity aims to limit the difference between α and β, t-closeness chooses to limit the difference between δ and β, i.e., D_T is considered as a public information and the knowledge gain between α and δ is about the whole population. For this purpose, t-closeness limits the distance between D_T and D_q, where D_q is the distribution of the confidential attribute value in the q-block. Then, a q-block is said to have t-closeness if the distance between the distribution of a confidential attribute in this q-block and the distribution of the attribute in the whole table is no more than a threshold t. Therefore t-closeness is formally defined as follows :

Definition 2.23. (t-closeness) [LLV07]
Let $T(A_1, \dots A_n, C)$ be a table, C a sensitive attribute, and t a threshold defined by a user. A table is said to have t-closeness if all q-blocks in T have t-closeness.

t-closeness completes ℓ-diversity, and helps to protect from attribute disclosure, by using both skewness and similarity attacks. For that, t-closeness guarantees that the distribution of confidential value in q-blocks (D_q) is similar to the distribution of confidential value of the whole population D_T. Thus, all q-blocks will have approximately the same D_q.

2.4.4.3 Implementing t-closeness

t-closeness is a difficult property to achieve. It requires the measurement of the distance between two probabilistic distributions, either numerical and categorical. Li *et*

al. [LLV07] proposed to adopt Earth Mover's Distance (EMD), which is based on the minimal amount of work needed to transform one distribution to another by moving distribution mass between each other. Then, they prove that t-closeness with EMD satisfies generalization and subset properties, which implies monotonicity, and can be easly integrated with the Incognito algorithm [LDR05].

Other types of attacks exist in the literature. We can mention the brute force attack, based on the knowledge about the generalization algorithm itself, introduced by Wong *et al.* [WFWP07]. Liu *et al.*[LWZ10] proposed k-jump strategy that penalizes cases where recursion is required to compute the disclosure set.

The state of practice is based on standards for generalization of certain types of information, e.g., any disclosed geographic unit must contain at least 10.000 or 100.000 respondents. Legislation and compliance frameworks detail the types and specificity of data generalization and suppression that are deemed to make data safe for releasing. The main problem of this approach is that new domain requires new rules and the proliferation of domains where data are collected makes this approach impractical [CT13]. In the middle of the previous decade, the research community began exploring new privacy notions that are not based on *syntactic* definition of privacy, most prominent which is *differential privacy* [Dwo06].

2.4.5 Introduction to Differential Privacy

In contrast with privacy-preserving data publishing (PPDP), Privacy-preserving data mining or analysis (PPDM) consists in releasing statistical facts about the population studied without compromising the privacy of respondents. We distinguish two different manners to release statistics, *iterative* and *noniterative*. In the noniterative setting, statistics are computed and published, then data are not used further, i.e., simply destroyed. However in the iterative setting, data can not be destroyed. Therefore, queries and the responses to these queries are simply modified in order to protect the privacy of individuals in dataset. In such context differential privacy has been introduced to ensure that the removal or addition of a single dataset item does not (substantially) affect the outcome of any analysis. Formally,

Definition 2.24. (Differential Privacy) [Dwo06, Dwo08]
Let be T_1 and T_2 two datasets. A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all datasets T_1 and T_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$, where :

$$Pr\left[\mathcal{K}(T_1) \in S\right] \leq exp(\epsilon) \times Pr\left[\mathcal{K}(T_2) \in S\right] \tag{2.17}$$

Note that the parameter ϵ in Defintion 2.26 is public. It varies from 0.01 to $\ln 3$, and its choice is a social question. For instance, if the probability that some bad event will occur is very small then it might be tolerable to increase it by such factors as 2 or 3. Also, differential privacy suffers from a data utility issue due to the inherent uncertainty and the fact that errors may be significant with high probability.

Clifton and Tassa [CT13] examined in details the two types of privacy models : syntactic models of anonymity and differentiel privacy. They concluded that syntactic models of anonymity (k-anonymity, ℓ-diversity, t-closeness, ...) are designed for PPDP while differentiel privacy is typically applicable for PPDM. Hence, one approach cannot replace the other. However, they are not necessarily exclusive. A key point adressed in [CLLS10] is that k-anonymity must introduce some random variability in the anonymizing process. Indeed, the generalization function must be developed using ϵ-differentially private mechanism.

2.5 Combinatorial Group Testing

2.5.1 Problem Statement

The identification of *bad* or *defective* members of a large population is an expensive and tedious process. This problem dates back to *World War II*, where the objective was to determine, in a population P of n members, which individuals are infected with syphilis.

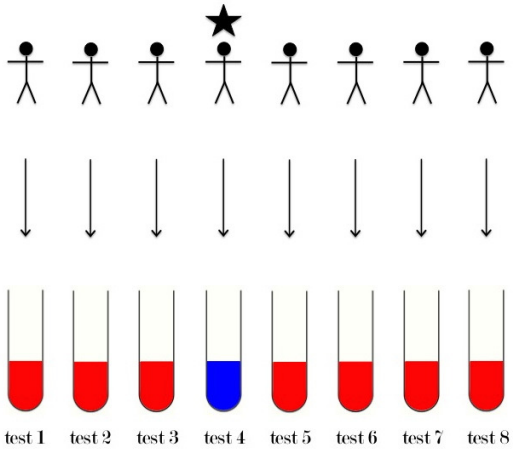

FIGURE 2.9: Individual tests.

Figure 2.9 depicts a classical approach consisting into two parts :

1. Samples of blood are drawn from individuals,

2. Each blood sample is subjected to a laboratory analysis which reveals the presence or absence of syphilitic antigen.

For instance, the presence of syphilitic antigen in the *test* 4 is a good indication of infection. Note that when we utilize this approach, n chemical analyses are required in order to detect all infected members of a population of size n.

In 1943, Dorfman [Dor43] formulated a new approach using *group tests* to reduce the total number of tests. A group test consists of selecting a set of samples $T \subset P$, extracting a few drops from each sample in T, pooling them together, and performing a single experiment to determine whether or not T contains infected individuals. Therefore, the outcome of a group test is *"contains at least one infected person"* or *"contains no infected person"*.

Other applications that fit this framework include [EGH07] :

Screening vaccines for contamination. In this case, individuals are vaccines and tests are cultures done on mixtures of samples taken from selected vaccines.

Clone libraries for a DNA sequence. Here, the individuals are DNA subsequences (called clones) and tests are done on pools of clones to determine which clones contain a particular DNA sequence (called a probe) [MSES97].

Pattern matching algorithm. Searching for a pattern in a text with a bounded number of mismatches [CEPR07].

Data forensics. In this case, individuals are documents and the tests are applications of one-way hash functions with known expected values applied to selected collections of documents. The differences from the expected values are then used to identify which, if any, of the documents have been altered [GAT05].

2.5.2 Group Testing

We distinguish two classes of group testing scenarios : *combinatorial* and *probabilistic*. In combinatorial group testing (CGT) scenarios, the number of bad members is either fixed or had an upper bound d where $1 \leq d \leq n$, while in probabilistic group testing (PGT) scenarios, defectives occur with some probability. We also distinguish between *adaptive* and *nonadaptive* group testing. A testing scheme that makes all its tests in a single round, with all test sets determined in advance, is said to be nonadaptative. In adaptive group testing, we specify these tests one at a time, using the outcome of the previous tests [ZK00]. Formally,

Definition 2.25. (Positive and negative tests) [ZK00]
Let be P a finite set of binary n-vectors (or integers from 0 to $2^n - 1$), and $T \subseteq P$ a (group) test.
Given a set S : If $T \cap S \neq \emptyset$ then T is positive with respect to S, else ($T \cap S = \emptyset$) T is negative with respect to S.

Definition 2.26. (Syndrome, d-separable, and d-disjunct) [ZK00]
Let be $\mathcal{T} = (T_0, T_1, \ldots, T_{m-1})$ a testing schema, and S a set with a cardinality $|S|$.
If $Q \subseteq \mathcal{T}$ is a set of positive tests in $\mathcal{T}$ with respect to S, then Q is the syndrome of S with respect to the testing schema $\mathcal{T}$.
$\mathcal{T}$ is d-separable if the syndrome Q of each set S where $|S| \leq d$ is distinct.
$\mathcal{T}$ is weakly d-separable if the syndrome Q of each set S where $|S| = d$ is distinct.
$\mathcal{T}$ is d-disjunct if for each singleton $\{x_i\}$ with a syndrome Q_i and for each set S not containing x_i where $|S| \leq d$, Q_i is not contained in Q the syndrome of S.

2.5.3 The Special Case of 1 out of n

We are interested here in nonadptive CGT, in which all the subsets to be tested have to be decided ahead of time, i.e., before any subset is tested. We can design a scheme where we experiment only $1 + \log n$ groups to determine the sample infected in Figure 2.9.

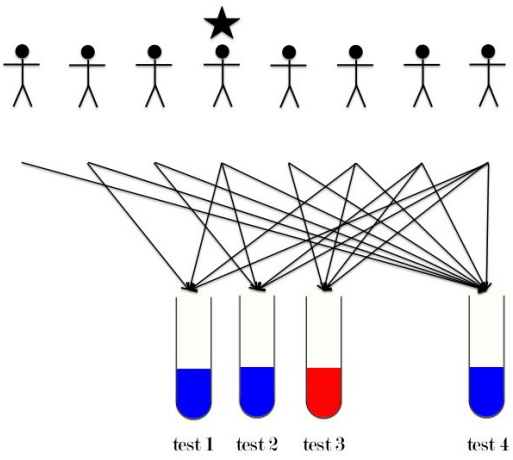

FIGURE 2.10: Group tests.

We explain which samples are selected and how these samples are pooling together to determine which individual is infected. For simplicity, assume $n = 8$ is a power of 2, and the samples are numbered from s_0 to s_7 (left-to-right).

As depicted in Figure 2.10, one of the tests (i.e., test 4) is used for the pooling of all samples. This serves to confirm that there is an infected individual in our population. The remaining $\log n = \log 8 = 3$ experiments are for determining which individual is

infected, and are as follows. For $j = 1, \ldots, 3$, the j^{th} experiment is for the pooling of those s_i for which the integer i has a 1 in the j^{th} least significant bits of its binary representation ; i.e., a sample s_i is in the j^{th} test if, in the binary representation of the integer i, the j^{th} least significant bit is a 1. Table 2.14 summarizes the pooling of samples in the three tests.

TABLE 2.14: Summary of samples pooling

Tests	Samples			
$1^{st}(j=1)$	s_1 $(001\underline{1})$	s_3 $(011\underline{1})$	s_5 $(101\underline{1})$	s_7 $(111\underline{1})$
$2^{nd}(j=2)$	s_2 $(01\underline{1}0)$	s_3 $(01\underline{1}1)$	s_6 $(11\underline{1}0)$	s_7 $(11\underline{1}1)$
$3^{rd}(j=3)$	s_4 $(\underline{1}00)$	s_5 $(\underline{1}01)$	s_6 $(\underline{1}10)$	s_7 $(\underline{1}11)$

To determine which sample s_i is infected, the binary representation of integer i is constructed one bit at a time, as follows : For $j = 1, \ldots, \log n$ in turn, if the j^{th} test is positive then the j^{th} bit of i is 1, and if the test is negative then the bit is 0. For instance, in Figure 2.10 the 1^{st} and 2^{nd} tests are positives and the 3^{rd} test is negative. This implies that the infected sample s_i has a 1 in bit positions 1 and 2 of the 3-bit binary representation of i ; and a 0 in bit position 3, i.e., $s_i = s_{011} = s_3$.

We saw that for the case $d = 1$, it is straightforward to design a nonadaptive scheme using $O(\log n)$ tests. For the general case, $d \geq 2$, designing efficient general testing schemes is more chalenging. The best known general-purpose adaptive schemes use $O(d \log(\frac{n}{d}))$ tests, whereas the number of tests used by the best known general-purpose nonadaptive schemes is $O(d^2 \log n)$ [ZK00].

2.6 Conclusion

In this chapter, we have defined what we mean by computer security, then we provided a non exhaustive overview of several security mechanisms that are complementary and, in the same time, the basis of our solutions and constitute our security toolbox. Other security mechanisms still exist in the literature but have not been addressed because not explored in our research. We can mention for example secret sharing and garbled circuit.

The goal of our work is not to propose new competing solution to what already exists, but rather to adapt existing ones to secure, and privacy-preserving data and services when using BPaaS. We will study later in this manuscript, the use of these mechanisms to address three different security issues by integrating the solution at the design stage of the service.

Next, we present our solution to preserve business secret of companies when using BPaaS to develop business processes by selection.

3

Security-Aware Business Process as a Service

Contents

3.1 Introduction

Cloud services have been extensively studied in recent years and two categories were proposed: application services and utility computing services [AFG+09]. Application

services, i.e., Software as a Service (SaaS), offer complete and pre-designed services, where end-users access with authentication protocols and use services maintained by cloud providers. Utility computing services, i.e., Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), provide fundamental computing resources that are used to develop, test, deploy and monitor process-based application (PBA). Therefore, *hosting* business processes in specialized cloud providers may lead to lower costs, by sharing hardware and software resources, as well as administrative staff, and enables pay-as-you-go pricing model [CLN12].

The cloud model also gives the opportunity for organizations to compose and re-use cloud services from a variety of cloud providers to create what's known as cloud syndication [YZB11, Pap12, ZZYB13]. Cloud syndications at the SaaS level are termed Business Process as a Service (BPaaS), which, according to business analysts, is the next step forward in the evolution of cloud computing [Bit11]. The BPaaS model considers a *multi-party* cloud system, which consists of multiple cloud platforms and cloud's users. Thus, we define each cloud platform as being a *process curator* that hosts a set of business processes and maintains them long-term such that they are available for execution.

Currently, organizations outsource more and more business processes to process curators in order to take benefits from the cloud business model, and also to share data and services [RKM10]. Each *complex* business process deployed can be broken down into smaller (and more manageable) process fragments suitable for re-use to accelerate future process modeling [BMM06, KL06, CST10, HHLZ10, ICH10, MDKL11]. Indeed, a process fragment represents a self-contained and functionally complete artifact for process design and execution. These organizations are therefore defined as *process providers*.

As a result, process curators built over time and maintain large repositories of process fragments [RRvdA+11]. Such repositories may contain hundreds or even thousands of process fragments (e.g., Amazon.com, schema.org, etc.). These process fragments can be extracted, published and shared through libraries, allowing the design of new PBAs by selection [SKK+11, ASKW11, THvdHF13, SSY14]. The development of new PBAs supports to reduce not only the cost of designing new business processes but also to enhance homogeneity between them. For instance, Amazon.com[1] provides an application catalog (as of June 2015, there were more than 900 processes), that can be provisioned and re-used on the fly. In this chapter, we use the term *process consumer* to refer to such third organization that re-uses process fragments provided by process curators in the cloud.

[1] http://www.aws-partner-directory.com

The main problem that cloud computing paradigm implicitly contains is that secure outsourcing of sensitive as well as business-critical data and processes [BGJ+13]. In fact, there are several security risk issues when reusing process fragments in the BPaaS delivery model. The first issue is *how to ensure the end-to-end availability of PBAs ?*. Existing secure process composition mechanisms assume a fully trusted process provider, which is not always true, and focus on announced Service-Level Agreement (SLA) availability rates of process fragments.

However, in reality, a process provider may suspend the outsourcing of a given service including process fragment. Consequently, all PBAs that re-use this cloud service will be impacted and abnormalities on their executions will occur. One possible solution consists in keeping a copy of each process fragment by the process curator as long as it is needed. However, this solution requires that the process provider should let available its own process fragments after unsubscribing. In some cases that may well be true, but very often that is not the case.

A second key problem in outsourcing is that the hosting, the execution and the re-use of process fragments are considered as sensitive that may contain business secrets or provide personal information (e.g., SSN). Consequently, fragment's compositions may expose process providers' business activities, as well as process consumers and their end-users to confidentiality issues. Thereby, an adversary may be able to:

1. Reveal sensitive information about the process provider activities, such as details of how certain process fragments are composed or the list of process fragments provided by an organization;

2. Infer connections between end-users and a process provider by analyzing intermediate data, like input/output values produced by a process fragment, thus obtain and/or modify confidential and sensitive information by using SQL injection attacks [WMK06].

Both are considered to be unacceptable breaches of confidentiality.

Existing solutions characterize security as a set of attributes, where process providers and process consumers define their security constraints in terms of these attributes (e.g., Goettelmann *et al.* [GDG+14]). Thus, PBA's security is ensured if the security constraints of each fragment reused satisfy security constraints of the process consumer. But as the first issue, these mechanisms assume a fully trusted process provider and consumer, and are used to prevent only external attacks. In the case where an attacker is one of parts of cloud system, these mechanisms are not efficient.

In the same line, Benbernou *et al.* [BMLH07, MB10] proposed a privacy agreement model that spells out a set of requirements related to consumer's privacy rights in terms of how Web Service provider must handle privacy information as a bilateral SLA. Moreover, they provided a private data usage flow model to monitor at run time the compliance of requirements defined in the privacy agreement [BMH07, MBZ+10]. However, such approaches are not handling privacy preservation and do not deal with the availability of Web Services involved in a fragment of a business process and in a setting of the cloud. There have been some works on security-aware compositions [CFH06, DKM+11, SYTB13]. Unfortunately, these works do not consider service provenance and focus on access control, data integration and provenance.

The results presented in this chapter has been published in [BBA16]; and are an extension of our earlier works [BBA12, BBDA12] in which we formalized the reuse of process fragments in the cloud, and introduced the notion of anonymous process fragments for privacy-preserving business activities of organizations. In this chapter, we investigate how much we can secure PBAs while multi-organizations share a BPaaS in a multi-party cloud system and we provide a positive answer to the above questions. For that purpose, we propose an anonymization-based approach providing anonymous views on BPaaS to preserve the confidentiality of multi-tenant fragments, and to reduce the cost associated with the approach. At the same time, we enrich the approach with a notion of diverse view to guarantee the end-to-end availability of PBAs, and to reduce the cost associated with the approach. We make the following contributions:

Anonymous and diverse views. In order to hide the activity of a process provider sharing some of its process fragments with other organizations, we define a new notion of views on BPaaS handling the instances of shared and reused process fragments. Moreover, to ensure the availability of process fragments for building new PBAs, we also introduce the notion of diverse views handling the diversity of process fragment provenances.

Confidentiality and availability costs. To quantify the proposed framework's security, we use two types of cost: one for confidentiality, and another for the availability of process fragments in the BPaaS.

Secure Business Process as a Service. To take into account the aforementioned goals, the proposed secure framework is based on a multi-objective optimization approach.

Evaluation on real datasets. To validate the effectiveness and evaluate the performance of the proposed protocol, we have applied it to the QWS datasets [AM07,

AM08], then studied the impact on the quality of the BPaaS views. Experiments permitted us to set parameter values of the protocol.

The remainder of the chapter is structured as follows: Section 3.2 describes the problem statement through motivating examples. Section 3.3 gives some preliminaries on BPaaS and process fragment provenance for faster and easier design of process-based applications. After defining the security model for the BPaaS in Section 3.4, Section 3.5 presents the details of our protocol, including the anonymous and diverse views on BPaaS model for securing process fragment reuse. Experiment results of the proposed protocol and an optimization are presented in Section 3.6. Section 3.7 discusses related work and Section 5.6 concludes the chapter.

3.2 Motivating Examples

We start by setting out examples that motivate the research presented in the chapter. We present scenarios for reusing process fragments, that cannot resist several possible attacks. These scenarios infer availability and confidentiality issues.

3.2.1 Availability issue

In the first scenario, we allow for the possibility of an adversary using the BPaaS to outsource new business processes as process provider. Accordingly, an adversary may enrich the repository with new process fragments that can be reused by other organizations. We also allow for the possibility of an adversary to remove its own process fragments previously deployed on the BPaaS. Thereby, the availability of the adversary's process fragments will not be assured. The following example illustrates the availability issue.

Example 3.1. *Let us consider an employer business process EBP used by a human resources department (HRD) to manage employee accidents at work. EBP is a simple sequential pattern, it means an activity is enabled after the completion of another one. So, EBP can be represented as a business graph with a set of activities as depicted in Figure 3.1. Activities are listed in the following :*

1. *Check insurance number (CIN).*
2. *Create new accident declaration (CNA).*
3. *Check personal information (CPI).*
4. *Validate employee declaration (VED).*

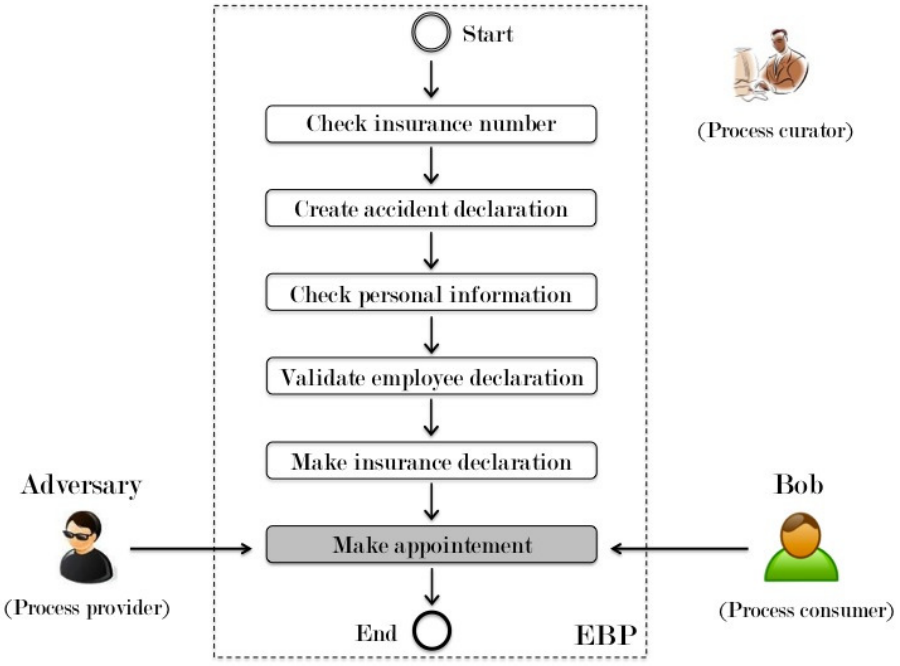

FIGURE 3.1: Process-based applications' availability issue.

5. *Make insurance declaration (MID).*

6. *Make appointment with insurance (MAI).*

Note that compositions in the application level (SaaS) are similar to the Web service compositions in SOC. Thus, CIN, MID and MAI are considered as cross-organization activities and require service invocations and data exchanges with insurance company through application programming interface (API).

The main problem in this scenario is, an adversary may ***provide*** *a set of process fragments in the BPaaS as a process provider. Suppose MAI is one of these process fragments. As depicted in Figure 3.2, MAI is split up into two roles: the sender (entity A) and the receiver (entity B). Sometime later, Bob, the process designer of HRD, uses the BPaaS for a faster design of EBP by selecting MAI. So, the end-to-end availability of EBP requires the availability of all reused process fragments including MAI. Thus, if the process curator or the (malicious) adversary chooses to remove MAI from the BPaaS repository, then EBP will become unavailable. This example perfectly illustrates the availability issue when reusing process fragments provided by a malicious process provider.*

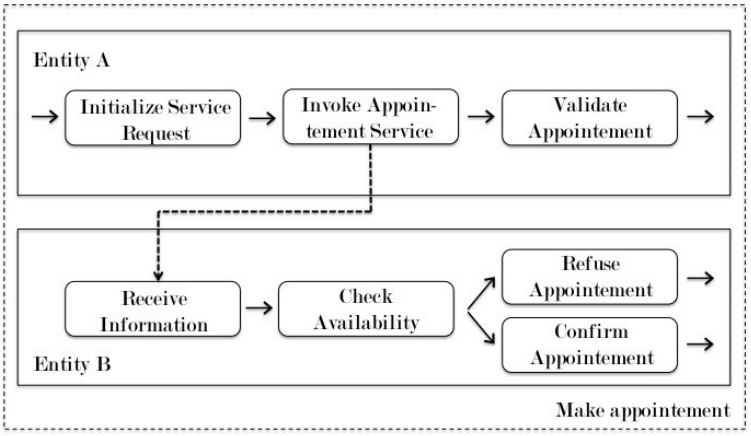

FIGURE 3.2: Process fragment for make insurance appointement.

3.2.2 Confidentiality issue

In a multi-party cloud system, an adversary can use the BPaaS as a process consumer to design new PBAs by selection. Therefore, the adversary will have access to all process fragments available in the BPaaS' repository. Figure 3.3 depicts the confidentiality issue when reusing process fragments.

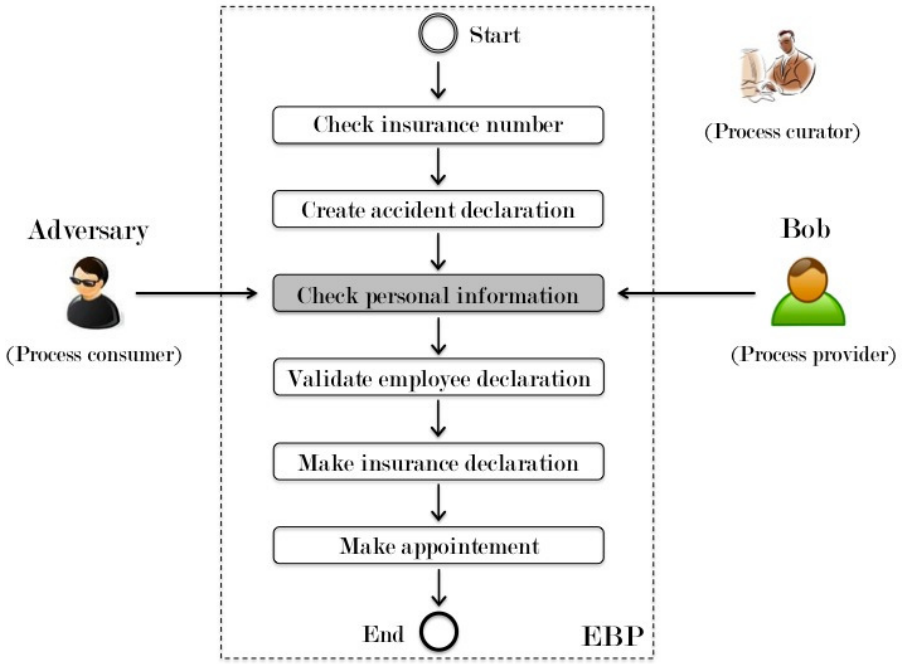

FIGURE 3.3: Process-based applications' confidentiality issue.

Example 3.2. *Let us consider the same PBA of Example 3.1 where EBP is used by HRD to manage employee accidents at work. In the novel scenario, we assume that Bob was the first using the BPaaS to outsource EBP. The fact to outsource a new business process enabled Bob to add a set of process fragments, including CPI, to the process*

repository. Sometime later, an adversary may re-use CPI to design a new PBA by selection (e.g., PACS[2]). Consequently, if the adversary is curious then he may be able to infer the provenance of CPI and make the connection between EBP and his end-users, i.e., respondents, by using SQL injection attacks to retrieve, for instance, the list of employees who have an accident during work.

3.3 Business Process as a Service

In this section, we give preliminary knowledge about business process outsourcing to the cloud. Business process as a service is also modeled at the end of the section.

3.3.1 A Model of Multi-party Cloud System

We consider the general multi-party cloud system depicted in Figure 3.4, which consists of multicloud platforms and multiple organizations or entities outsourcing their business processes (BPs). Each cloud platform includes a set of deployed process fragments (PFs) and a business process composer, i.e., BPEL engine. PFs are provided by the cloud platform itself or by external entities. For that, we define each cloud platform as being a *process curator* that hosts a set of PFs and maintains them long-term such that they are available for execution.

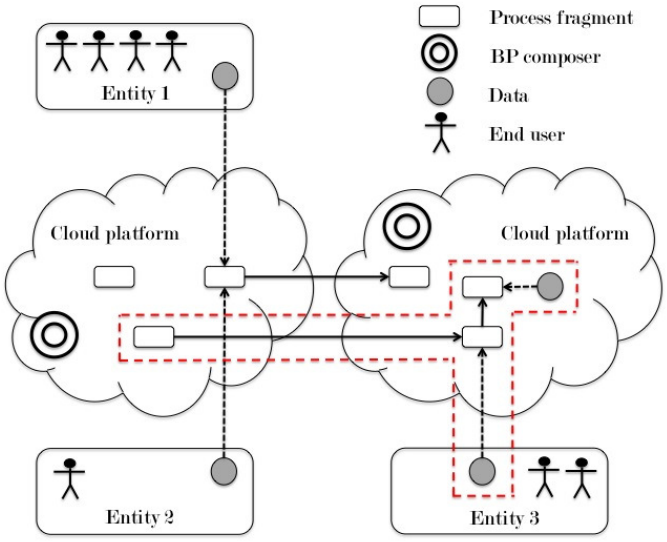

FIGURE 3.4: Multi-party cloud system.

[2]PACS (Picture Archiving and Communication System) is a hospital business process used by hospital staff to manage images and patients history.

Entities can be categorized into *process providers* and *process consumers*. Process providers are these companies or organizations that share and make their PFs available to the public. Process consumers are these organizations that re-use PFs in the cloud. An entity might at the same time be a process provider and a process consumer, and has its own end-users, i.e., respondents, and data resources. Formally,

Definition 3.1. (Multi-party cloud system) [BBA16]
A multi-party cloud system includes a set of cloud platforms $\{C_1, C_2, ...\}$ and a set of entities $\{E_1, E_2, ...\}$, where:

- Each cloud platform C_i is a tuple $(PF_{C_i}, DS_{C_i}, BC_{C_i})$, where $PF_{C_i} = \{f^1_{C_i}, f^2_{C_i}, ...\}$ is the set of all PFs in C_i; $DS_{C_i} = \{d^1_{C_i}, d^2_{C_i}, ...\}$ is the set of all data resources and databases of C_i; and BC_{C_i} is the business process composer of C_i.
- Each entity E_i is a pair (DS_{E_i}, EU_{E_i}), where $DS_{E_i} = \{d^1_{E_i}, d^2_{E_i}, ...\}$ is the set of all data resources and internal databases of E_i; and $EU_{O_i} = \{eu^1_{E_i}, eu^2_{E_i}, ...\}$ is the set of all end-users and respondents of E_i.

3.3.2 Business Process and Process Fragment

Business processes are at the core of organizations and an important success factor. They consist of a group of business activities undertaken by one or more entities. These activities are combined within or from different organizations and in turn offering them as value-added services. Therefore, software, that implement BPs, typically operate in a cross-organization and distributed environment. Based on existing works on business process modeling, e.g. Beeri *et al.* [BEKM06], where the authors model a BP as a directed labeled graph. We enrich it with the definition of process fragments.

We assume the existence of two domains $\mathcal{N}$ of nodes, and $\mathcal{L}$ of node labels. $\mathcal{L}$ is the disjoint union of several domains including data values, attribute names, data element names, and activity names. Formally,

Definition 3.2. (Business graph) [BM76, BEKM06]
A business graph is a pair $\mathcal{G} = (G, \Gamma)$, where:

- $G = (N, E, \Psi)$ is a directed graph in which $N \subset \mathcal{N}$ is a finite set of nodes, E is a set of edges with endpoints in N, and Ψ is an incidence function that associates with each edge of E an ordered pair of nodes of N; and
- $\Gamma : N \rightarrow \mathcal{L}$ is a labeling function for the nodes. Depending on their label type, we refer to the nodes in $\mathcal{G}$ as data element names, data attribute, data value, activity name, etc.

We now use business graphs to represent BPs. This representation can be considered as an early stage phase before BPEL or BPMN modeling. The business process is defined in Definition 3.3.

Definition 3.3. (Business process) [BEKM06]
A business process (BP for short) is a triple $p = (\mathcal{G}, start, end)$, where: $\mathcal{G}$ is a business graph; *start*, *end* are two distinguished activity nodes in $\mathcal{G}$; and each activity node in $\mathcal{G}$ resides on some path from start to end.

A BP is specified as a collection of business activities and is defined using a business graph. For convenience, we use the terms of *abstract process fragment* (abstract PF) and *concrete process fragment* (concrete PF) to represent each business activity, where: An abstract PF, i.e., task, define what a PF is supposed to do explicitly in the sense of a mathematical function or a black box description (with inputs and outputs). An abstract PF is implemented by several substitute concrete PFs. The choice among these substitute concrete PFs is based on their non-functional properties, which are also referred to as quality of service (QoS) [YZB11].

As discussed in [SKK+11], PFs can be created using two approaches. In the first one, called *top-down*, PFs are created by extracting connected structures from a given process. Thus, the PF is indeed a sub-graph of a process graph. In the second one, named *bottom-up*, a PF needs to be created from scratch. We consider the top-down approach, where process fragmentation is already done and concrete PFs are well distinguished and identified in the cloud platform. There are techniques in the literature, discussed in Section 1.4.2, that can help resolve BP's fragmentation issues. During process fragments composition, a list of desired abstract PFs is given to the business process composer, which instantiates each abstract PF by a concrete PF. In the following, we define process fragments.

We assume the existence of two domains $\mathcal{F}$ of concrete PFs and $\mathcal{A}$ of abstract PFs. Like instances and classes respectively in object-oriented programming. Two instances, i.e., concrete PFs, of the same class, i.e., abstract PF, are clones (see [DGRU13] for a recent paper on the topic). Then formally,

Definition 3.4. (Business subgraph) [BM76, BBA12, BBDA12, BBA16]
$\mathcal{H}$ is said a business subgraph of $\mathcal{G}$ (written $\mathcal{H} \subseteq \mathcal{G}$) iff:

- $N(\mathcal{H}) \subseteq N(\mathcal{G})$, where $N \subset \mathcal{N}$ is a set of nodes; and
- $E(\mathcal{H}) \subseteq E(\mathcal{G})$, where E is a set of edges; and
- $\Psi(\mathcal{H})$ is the restriction of $\Psi(\mathcal{G})$.

When $\mathcal{H} \subseteq \mathcal{G}$ but $\mathcal{H} \neq \mathcal{G}$, we write $\mathcal{H} \subset \mathcal{G}$ and call $\mathcal{H}$ a business proper subgraph of $\mathcal{G}$.

As BPs, we use the notion of business subgraph to define PFs as follows:

Definition 3.5. (Process fragment) [BBA12, BBDA12, BBA16]
A process fragment (PF for short) is a pair $f = (\alpha, \Delta)$, where: $\alpha \in \mathcal{A}$ is an activity requirement (abstract PF); and $\Delta : \mathcal{A} \rightarrow \mathcal{F}, \Delta(\alpha) = F_\alpha$ is a function providing a set $F_\alpha \subset \mathcal{F}$ of business proper subgraphs (concrete PFs) having the same abstract α.
If the cardinality $|F_\alpha| > 1$, then f is a multi-tenant PF with $|F_\alpha|$ clones: $f^{p_1}, f^{p_2}, \ldots, f^{p_{|F_\alpha|}}$.

We consider that a process consumer, generally an organization, submits a request for an abstract PF to the business process composer. The composer explores potential candidates and selects the best concrete PF according to functional and non-functional service level agreement (SLA), as well as, security constraints of the process consumer [YZB11, ZZYB13]. A concrete PF may need to be replaced per clone at runtime if it becomes unavailable or quality of service (QoS) degrades [KPP+13, DGRU13].

3.3.3 Business Process as a Service and Process-Based Applications

Business process as a service (BPaaS) consists of a set of BPs deployed in a multi-party cloud system containing process curators, providers, and consumers. These BPs are composed by BP composers (BPEL Engine) using multi-tenant PFs and different data resources. Usually, end-users or respondents have to use BPs in their everyday life through Web frontends and mobile applications (e.g., to submit an insurance claim or to apply for a permit to build a house). In order to model a BPaaS, we assume the existence of two domains $\mathcal{P}$ of BPs, and $\mathcal{I}$ of BP's identifiers. Then formally,

Definition 3.6. (Business process as a service) [BBA12, BBDA12, BBA16]
A BPaaS model is a pair $\mathcal{S} = (P, \Theta)$, where: $P \subset \mathcal{P}$ is a finite set of BPs deployed on the BPaaS, $P = (p_1, p_2, .., p_i)$; and $\Theta : \mathcal{P} \rightarrow \mathcal{I}$, $\Theta(p) = id_p$ is an identification function for the whole BPs. Depending on the tenant deploying a BP, we identify the BP p_i in $\mathcal{S}$ by id_{p_i}.

Example 3.3. *Let $\mathcal{S}$ be a BPaaS shown in Figure 3.5, where: $P = \{p_1, p_2, p_3, p_4\}$ is a set of BPs, and $F = \{f_1, f_2, f_3\}$ is a set of PFs. Each BP $p_i \in P$ is identified by an identifier id_{p_i}; and $\forall f_j \in F$, we define an abstract PF α_{f_j}.*

We give for each α_{f_j} in $\mathcal{S}$, $\Delta(\alpha_{f_j})$ the set of concrete PFs:

- $\Delta(\alpha_{f_1}) = \{f_1^{p_1}, f_1^{p_2}, f_1^{p_3}, f_1^{p_4}\}$, *we say f_1 is a multi-tenant PF provided by all BPs in $\mathcal{S}$.*

- $\Delta(\alpha_{f_2}) = \{f_2^{p_1}, f_2^{p_2}, f_2^{p_3}\}$, *we say* f_2 *is a multi-tenant PF provided by three BPs* (p_1, p_2, *and* p_3) *in* $\mathcal{S}$.
- $\Delta(\alpha_{f_3}) = \{f_3^{p_2}, f_3^{p_4}\}$, *we say* f_3 *is a multi-tenant PF provided by two BPs* (p_2 *and* p_4) *in* $\mathcal{S}$.

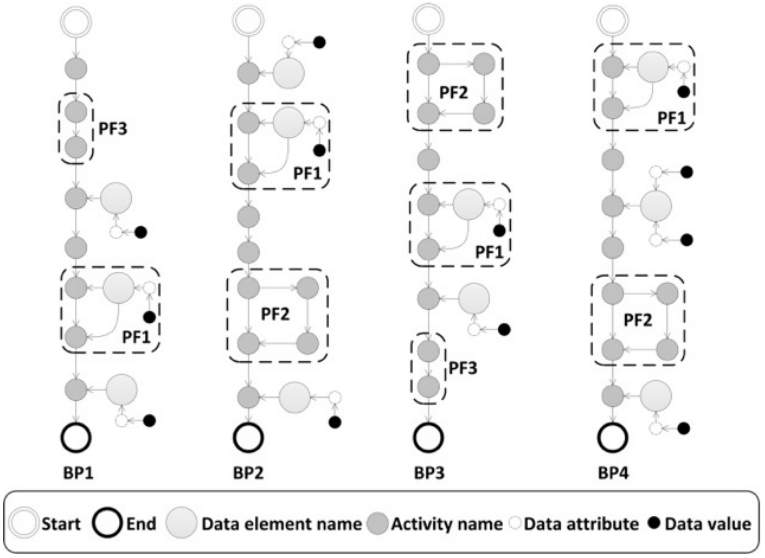

FIGURE 3.5: Multi-tenant PFs in the BPaaS

The greatest advantage of using multi-party cloud platform is the possibility to share one or a set of PFs. In fact, given a BPaaS $\mathcal{S}$ with some BPs deployed in it, we can design a new process-based application (PBA) by selecting existing PFs, and reusing them as Web Services. This concept is known as *Design by Selection* [ASKW11]. How to glue the PFs is out of the scope of the chapter, see [SKK+11] for more details. Formally,

Definition 3.7. (Design by selection) [BBA16]
Let us consider:

- $\mathcal{S} = (\mathcal{P}, \Theta)$ a BPaaS,
- p a new PBA to be developed in $\mathcal{S}$, and
- $\Omega : \mathcal{F} \rightarrow \mathcal{P}$, $\Omega(F) = \acute{p}$ a function performed to design a new BP $\acute{p}$ by selecting some PFs deployed in $\mathcal{S}$ and available in F.

In the BPaaS $\acute{\mathcal{S}} = (\mathcal{P}, \Theta, \Omega)$ where F is a set of PFs, we say that $p \longrightarrow_f \acute{p}$ w.r.t. Ω if $\acute{p}$ is obtained by reusing a PF $f \in F$ in order to develop p.
If $p \longrightarrow_{f_1} \acute{p_1} \longrightarrow_{f_2} \acute{p_2} \longrightarrow_{f_3} ... \longrightarrow_{f_k} \acute{p}$ w.r.t. Ω, then we say that $\acute{p}$ is construction of p by reusing a set $\{f_1, f_2, f_3, ..., f_k\}$ of PFs deployed in $\mathcal{S}$.

The Algorithm 1 presents the mechanism for designing and developing process-based applications by reusing PFs in BPaaS.

Algorithm 1 Design by Selection in BPaaS

Require: p a new BP to be developed in BPaaS.
Ensure: $\acute{p}$ a BP developed by reusing PFs in BPaaS.

```
for all PFs f in p do
    α ⟵ Identify f {define α the activity requirement of f}
    if Δ(α) ≠ ∅ {exist concrete PFs that implement α} then
        f_α ⟵ Select (f^{p_i}, Δ(α)); {select a concrete PF (by the composer)}
        p ⟶_{f_α} ṕ w.r.t. Ω; {concrete PF is reused to design p}
        p ⟵ ṕ; {prepare the next step}
    else
        f_α ⟵ Develop (f^p); {develop f_α from scratch}
    end if
end for
return ṕ.
```

3.3.4 Process Fragment Privacy

Hasan *et al.* [HSW07] defined *data provenance* as information that summarizes the history of the ownership of the item, as well as the actions performed on it. In other words, a record of where data came from and how it has been processed. Data provenance is extremely important for verifiability and repeatability of results, as well as for debugging and trouble-shooting workflows and business processes [DKM+11, DF08, DKR+11].

In BPaaS context, the fragment provenance permits to identify the process provider, i.e., the entity or organization that outsources, manages and monitors the process fragment. Currently, process consumers have access to the BPaaS' repository, and all information about process providers (see e.g., [AM07, AM08]). However, the provenance of PFs may be private information. Indeed, a process consumer should not be able to guess with a specified degree of certainty the provenance of a concrete PF. Formally,

Definition 3.8. (Fragment's provenance) [BBA16]
Let us consider F a set of concrete PFs f_i deployed in the BPaaS $\mathcal{S} = (\mathcal{P}, \Theta)$. $\forall f_i \in F$ there exists a set of functional and non-functional requirements that allows the description of f_i in $\mathcal{S}$.

Provenance requirements of f_i, denoted Pro_{f_i}, is any functional or non-functional requirement that uniquely identifies the provider of f_i (e.g., identity of the provider). Provenance should be removed entirely from the description of PFs in the BPaaS.

Quasi-provenance requirements denoted $QPro_{f_i}$, is a minimal set of functional and non-functional requirement that can be linked with external information to reduce the uncertainty over process providers. For instance, consider the activity requirement *PhoneService* or *SMService* can be linked external information such as business of entities to reveal the process provider's identity.

Simple requirements does not fall into any of the two categories above.

3.4 Security Definition for BPaaS

The framework of this chapter is one where organizations, i.e., process providers and process consumers, are connected to a trusted third party, i.e., process curator, in order to *(i)* outsource their BPs, and *(ii)* design new PBAs by selection, as depicted in Figure 3.6.

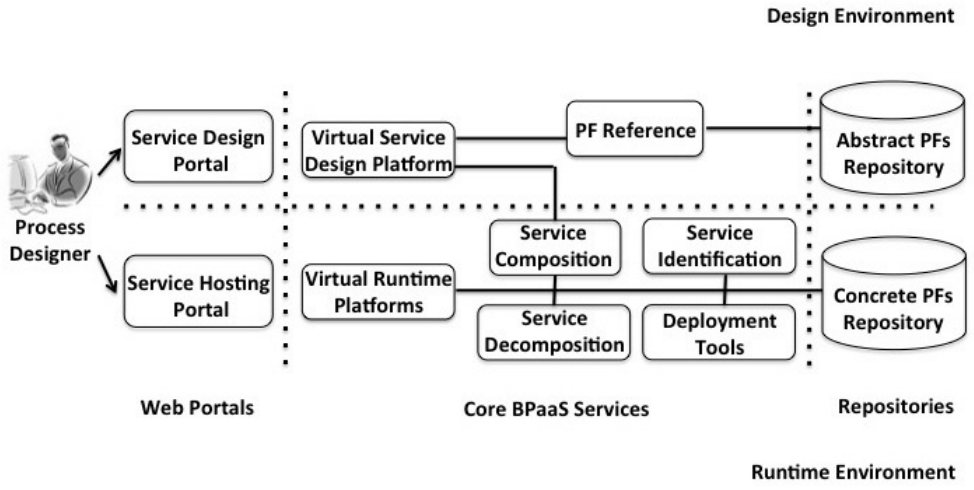

FIGURE 3.6: BPaaS delivery model.

We assume each BP designer has a personal account to use the BPaaS via Web Portals: *Service Hosting Portal* and *Service Design Portal*. Before the deployment phase, a new BP must be decomposed into a set of PFs and each PF should be identified, i.e., its activity requirement defined. The decomposition of BPs (respectively identification of PF) is carried out manually or automatically at the *Service Decomposition* Engine (respectively *Service Identification*). The Fragment Service Repository is assumed to be hosted at the process curator and, of course, the design of a new PBA requires the selection of a concrete PF in the repository.

3.4.1 Adversary Model

An adversary is defined by the capabilities that it has. We now list these resources, and of course an adversary may have combinations of these capabilities :

1. *Account (*DESIGN*,* HOST*, and* VIEW*).* An adversary may make multiple connections over time, with a personal account, to design process-based applications

(DESIGN), and/or to host process fragments to a process curator (HOST). We consider an adversary that can crack the personal accounts as outside of our attack model. An adversary is able to query BPaaS repositories through views, and sees the set of process fragments available to be selected (VIEW). Only adversaries with DESIGN access right can use views on BPaaS. We consider adversaries that can access the BPaaS repositories as outside of our attack model.

2. *Tenants (*LIST*).* An adversary may obtain the list of BPaaS tenants or process providers/consumers, possibly by homogeneity or background attacks, or by other extreme measures.

3. *Malicious (*MCS*).* An adversary can delete its own process fragments from the BPaaS. We consider an adversary that deletes the process fragments of other tenants as outside of our attack model.

4. *Curious (*CRS*).* An adversary may be able to eavesdrop on the BPaaS to disclose respondent privacy, and retrieve inputs/outputs values, i.e., intermediate data, manipulated by multi-tenants process fragments (e.g., by using SQL injection attacks).

3.4.2 Security Definitions

We look at the availability of PBAs, and confidentiality of the multi-tenant PFs. The availability of a process fragment requires that an adversary cannot make an organization unable to execute its process based application (i.e., availability of the reused fragments). The confidentiality requirements of the PFs are that an adversary should not be able to infer the provenance of a PF. We now formally define the security requirements for the notions above :

Availability. The attack we consider is that where an adversary removes concrete PF, reused to design PBAs, from the BPaaS. We present an oracle that is considered secure in our paper, and we prove availability by showing an adversary is equivalent to this oracle. Suppose the adversary has an oracle $\mathcal{A} : {V_F}^{f_i} \longrightarrow V_F$, where $\mathcal{A}({V_F}^{f_i})$ is a view on the BPaaS without f_i. In other words, the adversary can delete an arbitrary number of process fragments f_i from the BPaaS. We consider a protocol that allows such adversaries to be *strongly secure*.

Confidentiality. We present a second oracle, and we prove confidentiality by showing an adversary is equivalent to this oracle. Suppose the adversary has an oracle $\mathcal{B} : \emptyset \longrightarrow V_F$, where $\mathcal{B}()$ returns a view on the BPaaS. In other words, the

adversary sees a carefully chosen subset of PFs available in the BPaaS. A protocol with such an adversary has acceptable security only in cases where the subset is well chosen. We consider a protocol that allows such adversaries to be *weakly secure*.

3.4.3 Summary of Schemes' Security

Before we define the security of our system, we discuss the security (in the terms outlined above) of an *ideal* implementation that uses the trusted oracles. Such a system would require that the tenant uses the secure views on the BPaaS to design PBAs. The trusted oracles provide the secure views to the tenants. Clearly, we cannot do better than such an implementation.

TABLE 3.1: Security of the Protocol

Resources	Confidentiality	Availability
HOST	Strong	Strong
HOST and MCS	Strong	Strong
HOST and CRS and LIST	No secure	Strong
DESIGN and VIEW	Strong	Strong
DESIGN and VIEW and LIST	Weak	Strong
DESIGN and VIEW and CRS	Weak	Strong

Table 3.1 is a summary of an adversary's power with various resources (in our protocol); there are three categories of security: Strong, Weak, and No Secure. Where the first two are defined in the previous section, and *No Secure* means that the system does not protect this resource against this type of adversary. Thus, in many ways, the view is the lynchpin of the system. In the following sections we will present how to calculate the BPaaS view, and show the impact of each implementation on Security.

3.5 Security-Aware BPaaS

In this section, we outline a preliminary solution presented in [BBDA12] for secure business process outsourcing that should be viewed as *warmup* for the better solutions given later in the end of the section. The primary question that needs to be addressed is: *"How does the tenant develop the process-based application without knowing the business activities of process's provider?"*

3.5.1 Views on BPaaS

As explained above, a BPaaS is a set of BPs outsourced by organizations to multi-party cloud system. A BPaaS view provides a set of concrete PFs having the same abstract PF, i.e., activity, called clones [DGRU13]. Formally,

Definition 3.9. (BPaaS views).
Let us consider: $\mathcal{S}$ a BPaaS including a set of BPs, α an abstract PF, and V_α a set of concrete PFs having the same abstract PF α. V_α is called a *view* on $\mathcal{S}$ w.r.t. α.

TABLE 3.2: Process Fragments Repository.

PF id	Service Name	R. Time	Avai.	WSDL file location
FR32	SignatureVerification	165	100	http://www.securexml.net/SecureXML/SecureXML.wsdl
S6	Phone	150.45	100	http://ws.acrosscommunications.com/Phone.asmx?wsdl
GR90	PhoneVerify	131	80	http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl
TS7	CreditCardValidator	317	100	http://www.tpisoft.com/smartpayments/validate.asmx?wsdl
GBF	PhoneNotify	437.62	70	http://ws.cdyne.com/NotifyWS/phonenotify.asmx?wsdl
SSR	PhoneService	133	83	http://teleauth.com/phone/service.wsdl
...	...	...	...	...

TABLE 3.3: View on BPaaS w.r.t. Phone

PF id	Service Name	R. Time	Avai.	WSDL file location
S6	**Phone**	150.45	100	http://ws.acrosscommunications.com/Phone.asmx?wsdl
GR90	**Phone**Verify	131	80	http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl
GBF	**Phone**Notify	437.62	70	http://ws.cdyne.com/NotifyWS/phonenotify.asmx?wsdl
SSR	**Phone**Service	133	83	http://teleauth.com/phone/service.wsdl

Table 3.2 shows a process fragments repository containing a set of concrete PFs with their QoS [AM07, AM08]. The view on the repository w.r.t. Phone (depicted in Table 3.3) provides a set of concrete PFs: Phone, PhoneVerify, PhoneNotify and PhoneService, that implement this activity. In the following and in order to manage the views, we define a set of operations.

Definition 3.10. (Operations on BPaaS views).
Let us consider: $\mathcal{S}$ a BPaaS including a set of BPs; α and β two abstract PFs; V_α (resp. V_β) a view on $\mathcal{S}$ w.r.t. α (resp. β); We assume that it is possible to have one concrete PF that implements several abstract PFs, then:

1. $V_{\neg\alpha}$ is said a view on $\mathcal{S}$ w.r.t. $\neg\alpha$, iff $V_{\neg\alpha}$ contains all concrete PFs in $\mathcal{S}$ **not** having the abstract PF α (Negation).

2. $V_{\alpha\wedge\beta}$ is said a view on $\mathcal{S}$ w.r.t. $\alpha\wedge\beta$ iff $V_{\alpha\wedge\beta}$ contains all concrete PFs in $\mathcal{S}$ having the abstract PFs α **and** β (Conjunction).

3. $V_{\alpha\vee\beta}$ is said a view on $\mathcal{S}$ w.r.t. $\alpha \vee \beta$, iff $V_{\alpha\vee\beta}$ contains all concrete PFs in $\mathcal{S}$ having the abstract PFs α **or** β (Disjunction).

3.5.2 Anonymous Views on BPaaS

3.5.2.1 Definitions

As previously mentioned, PFs can be selected when designing PBAs. Unfortunately, the fact to know the provenance of a concrete PF may disclose the process provider's business secret. Therefore, the process curator would like to protect reused concrete PFs against link to process providers in $\mathcal{S}$.

Our approach to ensure BPaaS confidentiality, will be to hide a carefully chosen subset of process fragments. Inspired by k-anonymity model in databases, we have defined *k-anonyfrag*, an anonymity model for process fragments, which consists in generating anonymous views on the BPaaS [BBDA12]. In other words, we will project BPaaS repository on a restricted subset F of concrete PFs called anonymous view, allowing users access only to the V_F.

The *k-anonyfrag requirement* below, which states that in every view V_α on BPaaS repository we have at most K clones. Otherwise, there exists at most K concrete PFs having the same AF α in $\mathcal{S}$.

Definition 3.11. ($\mathbf{K_l}$ – anonyfrag requirement). [BBA12, BBDA12]
K_l – *anonyfrag* requirement is for each view V_α on BPaaS w.r.t. α, it must contain at most K clones.

Since it seems impossible or highly impractical and limiting to make assumptions on PFs to a curious adversary to discover business activities of process providers when reusing a concrete PF to design a new PBA. In the following, we define a K_l – *anonyfrag*:

Definition 3.12. ($\mathbf{K_l}$ – anonyfrag). [BBA12, BBDA12]
Given a BPaaS $\mathcal{S}$ used by l tenants; and an abstract PF α implemented by at most K concrete PFs or clones in $\mathcal{S}$. An adversary knows that it exists at most K clones implementing α are hosted in $\mathcal{S}$; and doesn't know:

1. Exactly the number of tenants that provide the K concrete PFs among l tenants.
2. Which tenants exactly have provided/hosted the abstract PF in $\mathcal{S}$.

A view V_F satisfies K_l – *anonyfrag* if for every abstract PF $\alpha_i \in F$ the cardinality $|V_{\alpha_i}| \in [1, K]$.

3.5.2.2 Security Analysis

We assume each entity deploying *exactly one* concrete PF implementing α is the *best-case* scenario, and the *worst-case* scenario when an entity provides more than one concrete PF implementing the same abstract PF α. $K_l - anonyfrag$ implies that for *any* concrete PF f_i in $\mathcal{S}$:

A) Curious adversary

A curious adversary can guess the process provider of a concrete PF with probability $P_{pro}(f_i)$, even if the view is calculated an arbitrary number of times. Note that the probability is always minimum in the best-case. We have l tenants and each tenant can deploy exactly one concrete PF. Therefore, the probability to infer the process provider for a given concrete PF is calculated as follows:

$$P_{pro}(f_i) = \frac{1}{l} \tag{3.1}$$

However, in the worst-case scenario when each tenant can deploy more then one concrete PF (and maximally K), the probability is calculated as follows :

$$P_{pro}(f_i) = \frac{K}{l} \tag{3.2}$$

Note:

1. If $|V_{f_i}| \simeq l$, the probability $P_{pro}(f_i) \simeq 1$ is maximum. It means that practically all entities in the BPaaS provide the concrete PF f_i. In this case we cannot hide the provenance of a concrete PF, i.e., all tenants have deployed the same abstract PF.
2. If $|V_{f_i}| = 1$, the probability $P_{pro}(f_i) = \frac{1}{l} \simeq 0$ is minimum. It means only one tenant in the BPaaS deploys the concrete PF f_i. In this case we have a low probability that an adversary can guess the provenance of the concrete PF f_i.

B) Malicious adversary

A malicious adversary can make unavailable a PBA with probability $P_{avai}(f_i)$. In the case where an adversary deploys exactly one concrete PF, i.e., best-case, the probability is minimum:

$$P_{avai}(f_i) = \frac{1}{K} \tag{3.3}$$

However, in the worst-case, an adversary can deploy K concrete PFs, the probability is maximum and equal to :

$$P_{avai}(f_i) = \frac{K}{K} = 1 \tag{3.4}$$

In the following, we define the new notions of confidentiality cost and availability cost in anonymous views on BPaaS:

Definition 3.13. (Confidentiality and Availability costs).
Given a BPaaS $\mathcal{S}$ used by l tenants, a set F of concrete PFs f_i deployed on $\mathcal{S}$, and V_α a view on $\mathcal{S}$ w.r.t an abstract PF α that satisfies $K_l - anonyfrag$.

1. The confidentiality cost of a view V_α, denoted $C_c(V_\alpha)$, is the probability that a curious adversary can guess the provenance of a concrete PF f_i implementing α.

$$\begin{aligned} C_c(V_\alpha) &= P_{pro}(f_i)^{worst-case} \\ &= \frac{K}{l} \end{aligned}$$

2. The availability cost of a view V_α, denoted $C_a(V_\alpha)$, is the probability that a malicious adversary can make unavailable a process-based application that reuses a PF f_i implementing α.

$$\begin{aligned} C_a(V_\alpha) &= P_{avai}(f_i)^{worst-case} \\ &= 1 \end{aligned}$$

Theorem 3.14. *Anonymous views do not guarantee the availability of process-based application.*

Proof Sketch. The proof of this claim is easy, we just have to take the worst-case (where an attacker deploys K concrete PFs). We found the availability cost $C_a(V_\alpha) = 1$ (i.e., the probability that an attacker can make unavailable a process-based application is equal to 1). □

3.5.3 Diverse Views on BPaaS

3.5.3.1 Definitions

We introduce a new notion of diverse views on BPaaS to guarantee availability of PBAs. Our notion is close to that of *l-diversity* in databases [MGKV06], in which there are at least l different values of sensitive attributes. We extend this work to BPaaS security problem. For that, we define $T_l - diverfrag$, a diversity model for process fragments, which consists in generating diverse views on the BPaaS. This means the anonymous BPaaS views will be projected on a restricted subset F' of concrete PFs (called diverse view).

We consider the BPaaS $\mathcal{S}$ used by l tenants. $V_F^\star$ a view on $\mathcal{S}$ that satisfies $K_l - anonyfrag$ requirement. The $T_l - diverfrag$ *requirement* below, which states that in every anonymous view $V_F^\star$ on BPaaS and for each concrete PF $f_i \in F$, we have at least T *different* process providers. Otherwise, there exists at least T different process providers have deployed at most K concrete PFs having the same abstract PF in $\mathcal{S}$. In the following a tenant may deploy a set of concrete PFs having the same abstract PF α in $\mathcal{S}$.

Definition 3.15. (T_l – diverfrag requirement).
$T_l - diverfrag$ requirement is for each anonymous view $V_\alpha^\star$ on BPaaS w.r.t. α, it must contain at most K concrete PFs provided by at least T different process providers.

Since it seems impossible or highly impractical for a malicious adversary to make unavailable a PBA when removing a concrete PF from the BPaaS. In the following, we define a $T_l - diverfrag$:

Definition 3.16. (T_l – diverfrag).
Given a BPaaS $\mathcal{S}$ used by l tenants; and an abstract PF α implemented by at most K concrete PFs or clones deployed by at least T different tenants in $\mathcal{S}$. A malicious adversary:

1. can make unavailable at most $K - T + 1$ concrete PFs implementing α ; and
2. can not make unavailable at least $T - 1$ concrete PFs implementing α in $\mathcal{S}$.

A view $V_{F'}^\star$ satisfies $T_l - diverfrag$ if for every abstract PF α_i: the number of tenants that deployed concrete PFs implementing α_i: $|Tenant_{id}^{\alpha_i}| \geq T$.

3.5.3.2 Security Analysis

As previously mentioned, it is assumed a tenant deploying exactly one concrete PF implementing α is the *best-case* scenario, and the *worst-case* scenario when a tenant can deploy more than one concrete PF implementing an abstract PF α. In the following, we define confidentiality and availability costs in diverse BPaaS views. $T_l - diverfrag$ implies that for *any* concrete PF f_i in $\mathcal{S}$:

A) Curious adversary

A curious adversary can guess the process provider with probability $P_{pro}(f_i)$, even if the view is calculated an arbitrary number of times. Note that the probability is always minimum in the best-case scenario i.e., where $K = T$:

$$P_{pro}(f_i) = \frac{1}{l} \tag{3.5}$$

In the worst-case scenario, an adversary can maximally deploy $K - T + 1$. Therefore, the probability is calculated as follows :

$$P_{pro}(f_i) = \frac{K - T + 1}{l} \tag{3.6}$$

We note :

1. If $|V_{f_i}| \simeq l$ and $|Tenant_{id}^{\alpha_i}| \simeq 1$, the probability $P_{pro}(f_i) \simeq \frac{l}{l} \simeq 1$ is maximum. It means that one tenant in the BPaaS deploys the l PFs f_i.
2. If $|V_{f_i}| \simeq l$ and $|Tenant_{id}^{\alpha_i}| \simeq |V_{f_i}|$. It means that all tenants in the BPaaS use the PF f_i. In this case we cannot hide the business activity of tenants, i.e., all tenants have deployed the same PF.
3. If $|V_{f_i}| = 1$, the probability $P_{pro}(f_i) = \frac{1}{l} \simeq 0$ is minimum. It means that only one tenant in the BPaaS deploys the PF f_i. In this case we have a low probability that an adversary can guess the process provider of f_i.

B) Malicious adversary

A malicious adversary can not make unavailable PBA with probability $P_{avai}(f_i)$. The probability is maximum in the best-case :

$$P_{avai}(f_i) = \frac{K-1}{K} \tag{3.7}$$

However, in the worst-case the probability is minimum:

$$P_{avai}(f_i) = \frac{K-T+1}{K} \tag{3.8}$$

Theorem 3.17. *Diverse views guarantee the availability of process-based application.*

Proof Sketch. As Theorem 1, we just have to take the worst-case. We found the availability cost $C_a(V_\alpha) = \frac{T-1}{K} \neq 0$ (i.e., the probability that an attacker can make unavailable a process-based application is different from zero). □

The table 3.4 summarizes the contribution of the diverse views to improve availability and confidentiality in the worst-case scenario.

TABLE 3.4: Anonymous vs. Diverse Views

Views	Confidentiality cost (C_c)	Availability cost (C_a)
Anonymous	$\frac{K}{l}$	1
Diverse	$\frac{K-T+1}{l}$	$\frac{T-1}{K}$

3.6 Approximation and Evaluation

In this section, we present an approximative algorithm that provides a secure views on BPaaS. We model it as a multi-objective optimization problem, which consists in optimizing simultaneously the conflicting objectives of availability and confidentiality.

3.6.1 Formalization and notation

We are given a BPaaS $\mathcal{S}$ used by l entities, a set F of concrete PFs f_i deployed on $\mathcal{S}$, and V_α a view on $\mathcal{S}$ w.r.t an abstract PF α. A view $V_\alpha(K,T)$ is feasible if it constitutes a set of at most K concrete PFs implementing α provided by at least T tenants. The objectives of availability and confidentiality are modeled with functions A and C respectively, which have to be minimized simultaneously are considered:

$$A_{V_F}(K,\, T) = Max\{\, C_a(f_i) : f_i \in F\} \tag{3.9}$$

is the maximum of availability costs of all concrete PFs $f_i \in F$; and

$$C_{V_F}(K, T) = Max\{C_c(f_i) : f_i \in F\} \tag{3.10}$$

is the maximum of confidentiality cost of all concrete PFs $f_i \in F$.

Let OPT_A (resp. OPT_C) be the minimum availability cost (resp. confidentiality cost) of a feasible view (best case), where:

$$OPT_A = \frac{1}{K} \tag{3.11}$$

and

$$OPT_C = \frac{1}{l} \tag{3.12}$$

A feasible $(\alpha, \beta) -$ *approximate* view is such that:

$$A(K, T) \leq \alpha \ OPT_A \tag{3.13}$$

and

$$C(K, T) \leq \beta \ OPT_C \tag{3.14}$$

where $\alpha \geq 1$ and $\beta \geq 1$.
An $(\alpha, \beta) -$ *approximation* secure view outputs a solution which is simultaneously $\alpha -$ *approximate* on the first criterion (the availability), and $\beta -$ *approximate* on the second criterion (the confidentiality).

3.6.2 Quality of Views

To solve secure view problem, our protocol takes into account the criterias mentioned above. In order to set parameter values K_{ideal} and T_{ideal}, we define a quality function of a BPaaS view to compare the different views that can be obtained. For this purpose, we calculate the ratio between the number of PFs requested to the BPaaS and the number of PFs obtained in the view. Formally, we have :

$$Quality_V = \frac{\mid V_{obtained} \mid}{\mid V_{requested} \mid} \tag{3.15}$$

Where: $Quality_V \in [0, 1]$. Our goal is to obtain a high $Quality_V$, which indicates that the protocol used to create BPaaS views does not eliminate requested PFs. We say

that V is feasible if $Quality_V$ is greater than a threshold q. The threshold q is chosen manually that best selects acceptable and not acceptable BPaaS views.

3.6.3 A deterministic approximation algorithm

Given a deterministic $\alpha - approximation$ algorithm **A1** for the mono-criterion secure view problem, one can build an $(\alpha, \beta) - approximation$ algorithm for the bi-criteria secure view problem. We assume two boundaries $(\ln l, \sqrt{l})$ as a starting point of our research. We think that these values are sufficient to ensure the availability and confidentiality of PBAs. The algorithm called **K-Approx** is given in the following :

Algorithm 2 (K-Approx)

Require: A BPaaS $\mathcal{S}$ used by l entities and q.
Ensure: An $(\alpha, \beta) - approximation$ secure view on the BPaaS.
1: Find $K_{min} \leq \sqrt{l}$ with **A1** where : $Quality_V \geq q$.
2: Find $T_{max} \geq \ln l$ with **A1** where : $Quality_V \geq q$.
3: **if** $T_{max} \leq K_{min}$ **then**
4: $K = K_{min} \wedge T = T_{max}$;
5: **else**
6: **if** $\ln l \leq \frac{K_{min}+T_{max}}{2} \leq \sqrt{l}$ **then**
7: $K = T = \frac{K_{min}+T_{max}}{2}$;
8: **else**
9: Degrades q;
10: **return** STATE 1
11: **end if**
12: **end if**
13: **return** $V(K, T)$.

Theorem 3.18. *$V_F(K, T)$ is a deterministic $(\frac{\sqrt{l}}{\ln l}, \sqrt{l} - \ln l) - approximation$ secure view on BPaaS.*

Proof Sketch. Three cases are considered in **K-Approx**. Table 3.5 depicts availability and confidentiality costs for each case. So, we have

$$A(K, T) \leq \alpha \ OPT_A \tag{3.16}$$

and

$$C(K, T) \leq \beta \ OPT_C \tag{3.17}$$

where $\alpha = \frac{\sqrt{l}}{\ln l} \geq 1$ and $\beta = \sqrt{l} - \ln l \geq 1$.

□

TABLE 3.5: Availability and Confidentiality costs

	$T_{max} < K_{min}$ $\ln l < T < K < \sqrt{l}$	$K_{min} = T_{max}$ $K = T \in \left[\ln l, \sqrt{l}\right]$	$T_{max} > K_{min}$ $T = K = \frac{T_{max}+K_{min}}{2} \in \left[\ln l, \sqrt{l}\right]$
Availability Cost	$\frac{\ln l-1}{\sqrt{l}}$	$(=\sqrt{l})\ \frac{\sqrt{l}-1}{\ln l-1} \times \frac{\ln l-1}{\sqrt{l}}$ $(=\ln l)\ \frac{\sqrt{l}}{\ln l} \times \frac{\ln l-1}{\sqrt{l}}$	$(=\sqrt{l})\ \frac{\sqrt{l}-1}{\ln l-1} \times \frac{\ln l-1}{\sqrt{l}}$ $(=\ln l)\ \frac{\sqrt{l}}{\ln l} \times \frac{\ln l-1}{\sqrt{l}}$
Confidentiality Cost	$(\sqrt{l} - \ln l + 1) \times \frac{1}{l}$	$\frac{1}{l}$	$\frac{1}{l}$

3.6.4 Evaluation and Experiments

To validate the effectiveness and evaluate the performance of our approach to secure process fragment reuse in the BPaaS delivery model, we design a set of experiments on real QWS datasets [AM07, AM08].

1. The dataset [AM07] is a collection of quality of service information for 9 criteria of 365 real Web services which are collected using a Web Service Crawler Engine (WSCE). We call it dataset 1.

2. An updated QWS Dataset [AM08] that includes a set of 2507 Web services and their QWS measurements that were conducted in March 2008 using a Web Service Broker (WSB) framework. We call it dataset 2.

We assume that these two datasets contain a large proportion of concrete PFs which are provided by a set of process providers to be reused in a BPaaS and allow us to test our protocol on real data.

We first randomly select a set $\mathcal{A}$ of abstract PFs to build a BPaaS view with respect to $\mathcal{A}$. Let us assume that $\mathcal{A}$ = { crypto&security, Phone, SMS, Data, calculator, news, zipcodes, ISBN, location, Fax }. $\mathcal{A}$ will be used to generate views on both dataset 1 and 2.

As previously discussed, concrete PFs instantiate abstract PFs. We consider each abstract PF can be implemented by a set of concrete PFs, i.e., clones. Figure 3.7 (resp. Figure 3.8) depicts for each AF in $\mathcal{A}$ the number of concrete PFs deployed on BPaaS as well as the number of process providers (PPs) providing the abstract PF in dataset 1 (resp. dataset 2). Note for some abstract PFs, the number of concrete PFs is higher than PPs. For example, for SMS fragment, there are eight (resp. 33) concrete PFS in dataset 1 (resp. dataset 2) provided by seven (resp. 26) providers in dataset 1 (resp. dataset 2). This confirms the fact that a process curator may offer several clones of PFs

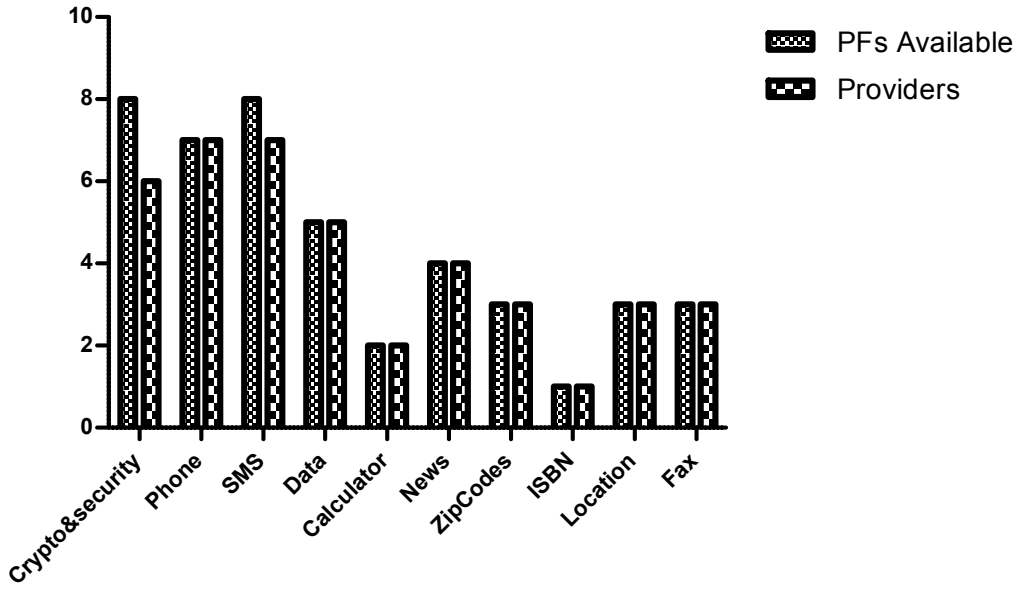

FIGURE 3.7: Dataset 1 - the number of concrete (PFs) and providers (PPs) / absract PF

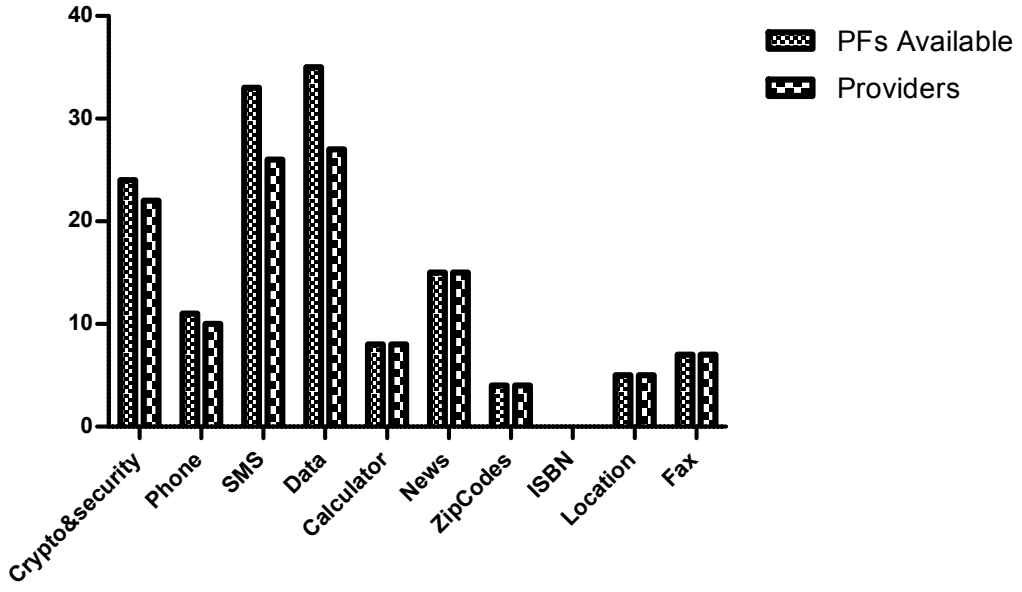

FIGURE 3.8: Dataset 2 - the number of concrete (PFs) and providers (PPs) / absract PF

having the same process provider. The effectiveness of applying a deterministic approximation algorithm in order to secure BPaaS views will be examined in the context of these datasets.

Figure 3.9 (resp. Figure 3.10) depicts the evolution of the quality of views with respect to K. We note that the quality of the views is maximum (i.e., equal to 1) when $K \approx 10$

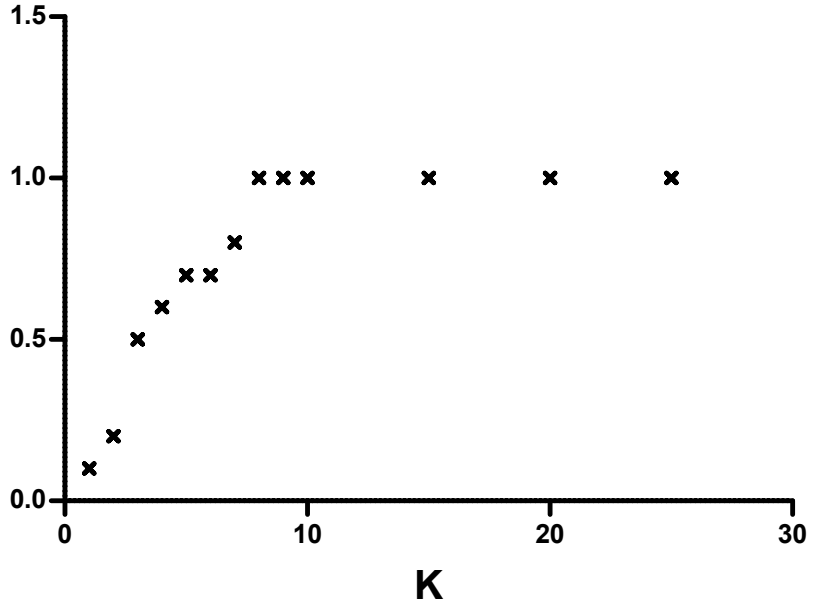

FIGURE 3.9: Dataset 1 - $Quality_V$ relative to K

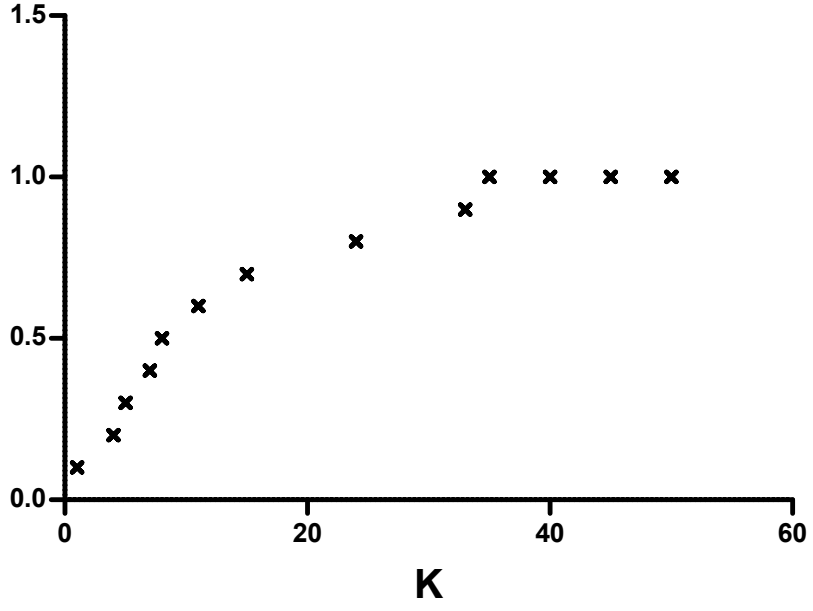

FIGURE 3.10: Dataset 2 - $Quality_V$ relative to K

in dataset 1. However, the quality of the views is maximum when $K \approx 40$ in dataset 2. This is mainly due to the size of the datasets ; and also to the number of concrete PFs that implement the abstract PFs. For instance, the SMS is implemented using 8 concrete PFs in dataset 1 and 36 in dataset 2, which is in line with the results obtained.

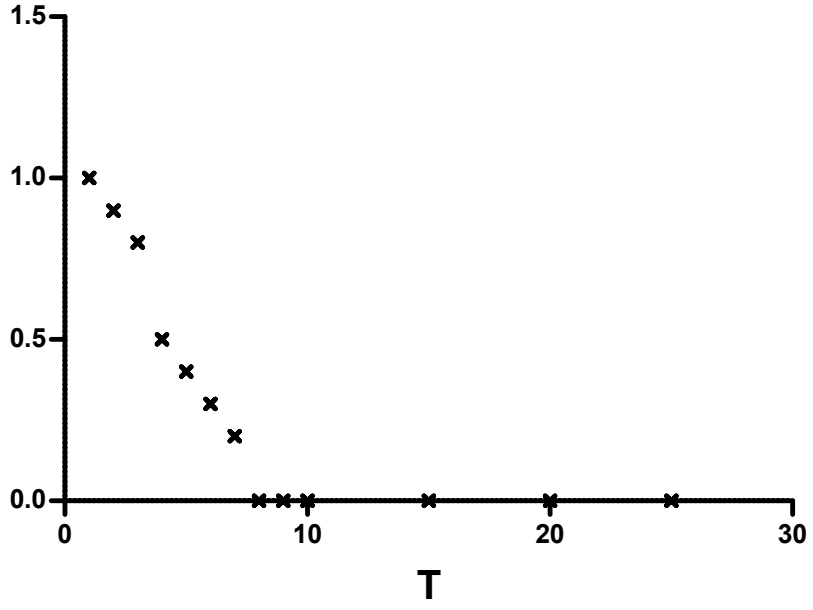

FIGURE 3.11: Dataset 1 - $Quality_V$ relative to T

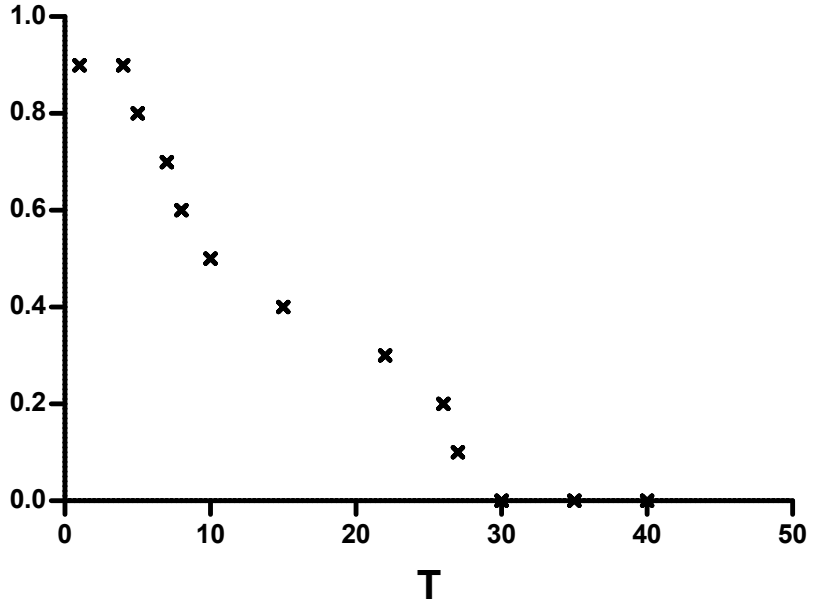

FIGURE 3.12: Dataset 2 - $Quality_V$ relative to T

Figure 3.11 (resp. Figure 3.12) depicts the evolution of the quality of views with respect to T. We note that the quality of the views is maximum (i.e., equal to 1) when $T = 1$ in dataset 1, and gradually declines up $T = 8$. This is due to the fact that we have at most 8 providers that deployed an abstract PF in dataset 1. However, the quality of the views is maximum (i.e., equal to 0.9) when $K = 4$ in dataset 2, and gradually declines

up $T = 30$. As dataset 1, this is due to the fact that it exists at most 30 providers that deployed an abstract PF in dataset 2.

3.7 Related Work

There is a huge literature on business process decomposition (fragmentation), and we briefly focus here on the work most relevant to our paper. There are two main objectives : One is to enhance the execution of the original process [BMM06, KL06], and another is to allow reusing process fragments in future business process modeling. Most of the recent work has focused on the second objective, and addressed the issue of identifying process fragments with the right level of abstraction in order to reuse, and increase the ability to communicate and analyze them [CST10, HHLZ10, ICH10, SKK+11]. In [MZZW09] a measurement approach was introduced to quantitatively evaluate service identification. Based on a set of design metrics (including: service granularity, coupling, cohesion and business entity convergence), the weighted features are combined to conduct an overall evaluation of a service. Other papers (e.g., [HHLZ10, DGRU13]) addressed the problem of managing large process model repositories. Paper [HHLZ10] designed a business knowledge repository enabling the reuse of process fragments. Along the same lines, [DGRU13] proposed an indexing structure to support the fast detection of clones (i.e., duplicate fragments) in repositories.

Moreover, nowadays, with the emerging technology of cloud computing, organizations have increased their interest in business process and service outsourcing to cloud providers [Pap12, THvdHF13]. Papazoglou [Pap12] presented a cloud blueprinting approach, which, equips developers with a unified approach that lets them develop cloud applications on top of existing applications at any layer of the cloud stack from multiple cloud providers. Taher *et al.* [THvdHF13] provided a customization tool helping to manage configuring of functional and non functional aspects related to a BPaaS offering. However, privacy and security risk issues are not addressed in these papers. [ZZYB13, YZB11] proposed techniques to calculate the QoS values of services in cloud computing as well as composite services with complex structures.

There have been some works on security-aware compositions [ALMS09, AKB11, CFH06, SYTB13, DKM+11]. In [ALMS09], it is investigated the execution of BPEL processes in different cloud computing delivery models (IaaS, PaaS and SaaS), and showed security and trust issues that affect the business processes outsourcing. However, they did not provide a solution architecture for the investigated challenges and requirements. Alsouri *et al.* [AKB11] addressed some of the security problems that arise when outsourcing business processes in the PaaS delivery model. They provided an architecture

which follows the compliance-by-design principle, allowing to remotely verify the correct execution of a business process. Works in [BDF05, CFH06, DKM+11, SYTB13] do not consider service provenance and focus on access control, data integration and provenance.

Benbernou *et al.* [BMLH07, MB10] proposed a privacy agreement model that spells out a set of requirements related to consumer's privacy rights in terms of how Web Service provider must handle privacy information as a bilateral SLA. Moreover, they provided a private data usage flow model to monitor at run time the compliance of requirements defined in the privacy agreement [BMH07, MBZ+10]. However, such approaches are not handling privacy preservation and do not deal with the availability of Web Services involved in a fragment of a business process and in a setting of the cloud.

The chapter is an extension of our earlier works [BBA12, BBDA12] in which we formalized the reuse of process fragments in the cloud, and introduced the notion of anonymous process fragments for privacy-preserving business activities of organizations. To the best of our knowledge, the work described in this chapter is the first to address the availability and confidentiality issues at the same time when reusing process fragments in the BPaaS delivery model.

3.8 Conclusion

Cloud computing and Business Process as a Services are new emerging delivery models offering the possibility to Business Process Outsourcing and enabling the enterprises to focus on their competencies. In this chapter we investigated the security issues when developing a new process-based application in BPaaS. First, we proposed an anonymization-based approach to preserve the business activities of an organization. However, we demonstrated that it is not sufficient to guarantee availability for process fragments reuse in BPaaS. For that, we extended it with the vision of diverse view of multi-tenants BPaaS. Furthermore, we presented the costs of both confidentialty and availability to be ensured at BPaaS level when reusing fragments. As a perspective, we would like to study distributed and elastic BPaaS in the cloud.

Next, we treat the security issues in PFs that manipulate sensitive data, i.e., biometric data.

4

Nonadaptive CGT for Secure Biometric Authentication

Contents

4.1 Introduction

Conventional password-based authentication is dead, like Telnet was dead and buried for the benefit of Secure Shell (SSH). It is not yet entered the minds of most cloud actors, let alone among users, but it is the case. There are several reasons for this. First, it is due to users themselves. In fact, more and more organizations outsource their business processes and in the same time, users are less and less aware of computer security issues. Second, the increased number of cloud services naturally increases security risks, especially when most of users use the same password for different cloud services. Additionally, it is accepted nowadays that logins by default are users' emails. Consequently, the centralization of email services aggravates the security problems, where one can easily find a valid login for a cloud service by a simple test.

The use of a password-based method is often seen as an unbearable constraint. For instance, there are times when it is hard to convince users that passwords are critical for the protection of personal data, and that is obligatory to choose strong ones. Furthermore, let us not forget to mention how passwords are transfered (in clear by phone or mail) and stored (using post-it or in clear on PC and mail server).

At the same time, other authentication methods have emerged, and gained more and more success thanks to smartphones and connected devices. Traditionally, three possible human authentication factors are distinguished (even if a forth one has already been introduced by Brainard *et al.* [BJR+06]). Table 4.1 depicts these factors which are based on :

- *"what I know"*, like password,
- *"what I possess"* like keys or any other object, e.g., RSA SecureID, and
- *"who I am"* like biometrics, e.g., fingerprint, iris recognition, facial images.

The questions we should ask are : *"Is the problem, faced by thousands of IT Directors, innocent ?"*, *"does the problem simply come from the use of password-based authentication ?"*, and *"are password-based authentications suitable for use at large-scale in the cloud ?"*.

The answer is password-based authentication is simply dead. Because passwords can be compromised, stolen, shared, or just forgotten. Moreover, passwords remain the main security guarantee at the responsibility of the user when using a cloud service, if one considers that other parameters are managed by the cloud provider. A solution may be to use other methods like biometrics, or the generalization of two-factor based authentications like ATM card.

TABLE 4.1: Existing user authentication techniques according to [RCB01]

Factors	Examples	Propreties
What I know	User ID Password PIN	Shared Many passwords easy to guess Forgotten
What I possess	Cards Badges Keys	Shared Can be duplicated Lost or stolen
Who I am	Fingerprint Face Iris Voice print	Not possible to share Repudiation unlikely Forging difficult Cannot be lost or stolen
What I know + what I possess	ATM card + PIN	Shared PIN a weak link (Writing the PIN on the card)

Biometrics are believed to be unique, unforgettable, non-transferable, and they do not need to be stored [TBCP08]. Due to these reasons, biometrics identifiers are now commonly used to identify individuals in more secure and more efficient ways than the conventional password-based method. For instance, Apple integrated a biometric sensor in "iPhone 5s" permitting fingerprint based authentication to access the smartphone features. And more recently, Google integrates biometrics-based authentication in its smartphones based on "Android M".

Despite of its advantages, there are some obstacles in a wide adoption of biometric authentication. Basically, biometric recognition is made in *local environment*, i.e., the matching is done with a template data stored in a secure smartcard or PC. However, the use of cloud services based authentications need to transfer and treat biometric data in the cloud. This poses a security problem, especially because biometric data are unique (i.e., they are not revocable due to their permanent nature). Therefore, unlike passwords that can be changed several times, each person has only ten fingerprints and, if biometric data are stolen they will be forever and can not be recovered. Consequently, the security of biometric data is extremely critical.

In addition, biometrics are *approximately* stable over the time. In fact, a password based authentication always provides a correct response if the passwords match, it grants access but otherwise refuses access. However, in a biometric based authentication, the overall accuracy depends on the quality of biometric data along with the basic characteristics of the underlying feature extraction and matching algorithm. Therefore, it cannot be directly integrated into most of the existing systems.

In our work, we are interested to remote biometric authentication in the cloud. More precisely, to the design of a process fragment based biometric authentication that can

be integrated in business processes as depicted in Figure 4.1. The protocol proposed is proven secure and uses computationally lightweight schemes (not expensive schemes) that carry out the comparison stage without revealing any information that can later be used to impersonate the user.

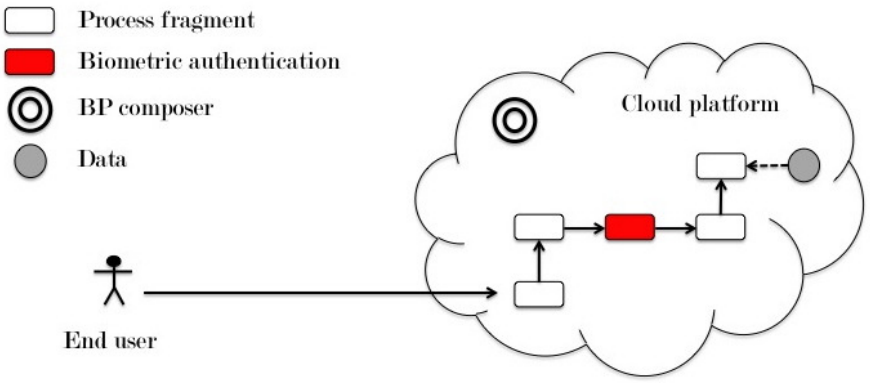

FIGURE 4.1: Remote Biometric authentication in the cloud.

Our main goal is to secure biometric based authentication on weak devices, when using cloud services, with respect to errors in repeated measurements of the same biometric data. For this purpose, we propose a nonadaptive combinatorial group testing based approach to permit a secure, approximative, and computationally non demanding remote biometric authentication. We implement the protocol and study its performances.

The remainder of the chapter is structured as follows : Section 4.2 gives preliminary definitions and describes the biometric authentication system, distance metrics used in matching algorithms and security issues engendered by these systems in the context of cloud computing. In Section 4.3, we discuss related work on techniques to secure remote biometric authentication proposed in the literature. After defining the security model for the biometric system in Section 4.4, Section 4.5 presents a first attempt to secure remote biometric authentication. In Section 4.6, we present a nonadaptative combinatorial group testing based approach to secure remote biometric authentication in the cloud. Section 4.7 presents experiment results of the proposed protocol and Section 4.8 concludes the chapter.

4.2 Preliminary Definitions

4.2.1 Biometric Systems

We distinguish between two types of biometric systems : *authentication* and *identification* systems. As password-based authentication, a biometric authentication system

aims to validate claimed logins or identities. However, in biometric identification systems, the objective is to determine the identity of a person based on his biometrics. In our work, we are interested in biometric authentication systems. Jain *et al.* gave a general definition of a biometric system as following :

Definition 4.1. (Biometric systems) [JRP06]
A biometric system may be viewed as a signal detection system with a pattern recognition architecture that senses a raw biometric signal, processes this signal to extract a salient set of features, compares these features against the feature sets residing in the database, and either validates a claimed identity or determines the identity associated with the signal.

Biometric authentication systems generally consist of two stages : *enrollment* and *authentication*. During the enrollment phase, users' biometric images are acquired and biometric templates are then created. These templates are stored in a database or on a portable storage device like a smartcard [DFM98]. During the authentication phase, the user presents a biometric sample which is compared with the stored template. The user is successfully authenticated if there is a near match between the input and the stored template.

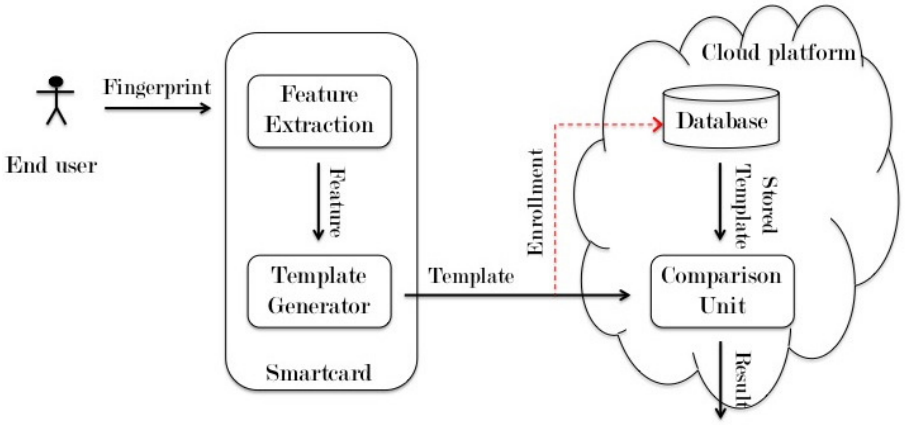

FIGURE 4.2: Biometric Authentication System.

Figure 4.2 depicts a biometric authentication system, which includes the following components :

End User. The *end user* uses his biometric, generally *fingerprint*, to authenticate himself to a remote authentication server.

Smartcard. The end user uses a *smartcard* to read a new biometric data. The smartcard contains a *feature extractor* to extract features from the biometric data and a *template generator* to generate biometric templates.

Note that we distinguish two ways to represent a fingerprint : *Fingercode representation* introduced in [JPHP00] and *Minutia representation* introduced in [MMJP09]. The smartcard connects to the terminal and sends to remote authentication server the generated biometric template.

Authentication Server. The authentication server contains a *comparison unit*, and a *database* to store clients' biometric identifiers. The authentication operation is effected at the comparison unit between the just received biometric template sent by the smartcard and the biometric identifier stored in the database.

Note that in a *biometric identification system*, the authentication operation is replaced by an *identification operation* which is done between the just received biometric template and all biometric identifiers stored in the database, in order to find a corresponding end user.

4.2.2 Similarities

For any fingerprint A and B, we assume that we have a corresponding binary fingerprint vectors $A = (a_1 \dots a_n)$ and $B = (b_1 \dots b_n)$ of length n. For simplicity, assume n is a power of 2. A is considered as the query fingerprint (i.e., acquired at the authentiction phase) and B is the stored fingerprint in the database (i.e., acquired at the enrollment phase).

A matching algorithm is interested in comparing A and B. For this purpose, we consider the following well-known similarities between the binary fingerprint vectors A and B, which are used in authentication operations.

Definition 4.2. (Hamming distance)
A Hamming distance between A and B is defined as :

$$\mathsf{HD}(A, B) = \sum_{i=1}^{n} (a_i \oplus b_i) \tag{4.1}$$

Definition 4.3. (Euclidian distance)
An Euclidian distance between A and B is defined as :

$$\mathsf{ED}(A, B) = \| A - B \| = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{4.2}$$

Definition 4.4. (Cosine correlation)

A cosine correlation between A and B is defined as :

$$\mathsf{Cos}(A,B) = \frac{A.B}{\parallel A \parallel \times \parallel B \parallel} = \frac{\sum_{i=1}^{n}(a_i \times b_i)}{\sqrt{\sum_{i=1}^{n} a_i^2} \times \sqrt{\sum_{i=1}^{n} b_i^2}} \tag{4.3}$$

Example 4.1. *Let us consider two fingerpints* A *and* B *with a corresponding binary vectors A and B depicted in Table 4.4. We calculate the different metric distance between A and B :*

TABLE 4.2: Binary representation of A and B.

$i=$	1	2	3	4	5	6	7	8
A	0	0	1	1	0	0	1	1
B	1	0	1	1	0	0	1	0

1. *Hamming distance :*

$$\begin{aligned}\mathsf{HD}(A,B) &= \sum_{i=1}^{8}(a_i \oplus b_i)\\ &= (a_1 \oplus b_1) + \ldots + (a_8 \oplus b_8)\\ &= 1+0+0+0+0+0+0+1\\ &= 2\end{aligned}$$

2. *Euclidian distance :*

$$\begin{aligned}\mathsf{ED}(A,B) &= \parallel A - B \parallel\\ &= \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}\\ &= \sqrt{(a_1-b_1)^2 + \ldots + (a_8-b_8)^2}\\ &= \sqrt{1+0+0+0+0+0+0+1}\\ &= \sqrt{2}\end{aligned}$$

3. Cosine correlation :

$$\begin{aligned}
\mathsf{Cos}(A,B) &= \frac{A.B}{\parallel A \parallel \times \parallel B \parallel} \\
&= \frac{\sum_{i=1}^{8}(a_i \times b_i)}{\sqrt{\sum_{i=1}^{8} a_i^2} \times \sqrt{\sum_{i=1}^{8} b_i^2}} \\
&= \frac{(a_1 \times b_1) + \ldots + (a_8 \times b_8)}{\sqrt{a_1^2 + \ldots + a_8^2} \times \sqrt{b_1^2 + \ldots + b_8^2}} \\
&= \frac{0+0+1+1+0+0+1+0}{\sqrt{0+0+1+1+0+0+1+1} \times \sqrt{1+0+1+1+0+0+1+0}} \\
&= \frac{3}{4}
\end{aligned}$$

Basically, an authentication operation attempts to arrive at a degree of similarity between two fingerprint vectors. This similarity is often expressed as a match score. In the case of fingerprints, the Euclidian distance is required to calculate the match score [BBC+10]. Note that when binary vectors are used to represent fingerprints, as shown in example 4.1, the Euclidian distance is equal to the square root of the Hamming distance ($Euclidian\ distance = \sqrt{Hamming\ distance}$). Thereby, we are going to use the Hamming distance as metric distance to calculate the match score.

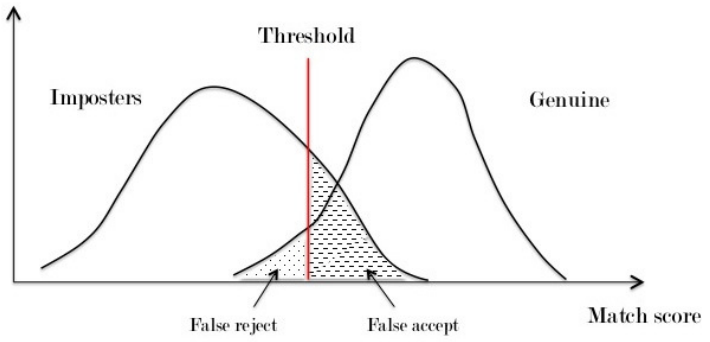

FIGURE 4.3: Error trade-off in a biometric system [RCB01]

As depicted in Figure 4.3, the final decision of match or no-match is made based on the match score [RCB01]. For this purpose, a decision threshold is first selected. If the score is less than the threshold, the fingerprints are determined not to match. However, if the score is greater than the threshold, a correct match is declared.

In biometric systems, there are two basic types of recognition errors, namely *false accepts* and *false rejects*. We have a false accept if a nonmatching pair of fingerprints is accepted as a match. On the other hand, if a matching pair of fingerprints is rejected by the

system, it is called a false reject. Depending on the technology used, the false rejection rate varies between 0.1% and 2.2%, and the false acceptance rate varies between 1.0% and 2.2% [JLG04b].

4.2.3 Security issues of biometric authentication systems

As mentioned above, the failure rate of biometric based authentications is very low. Therefore, biometric data identifiers can recognize persons with a *very high* probability. For that, they are considered as *personal* and *private* information. Belguechi *et al.* [BAC+11] summarized, in six points, privacy pitfalls arising when using biometric systems. We can mention the fact that :

- Biometric data can reveal *sensitive information* about the health, race, or ethnic origin of end users.
- Biometric data are not secret and can easily be acquired.
- Biometric templates do not ensure the privacy of biometric data. In fact, it is possible to reconstruct a biometric data using the corresponding template stored in a database.
- Biometric data do not ensure the anonymity of end users. Because an end user can be linked between different cloud services.
- Biometric data are irrevocable.

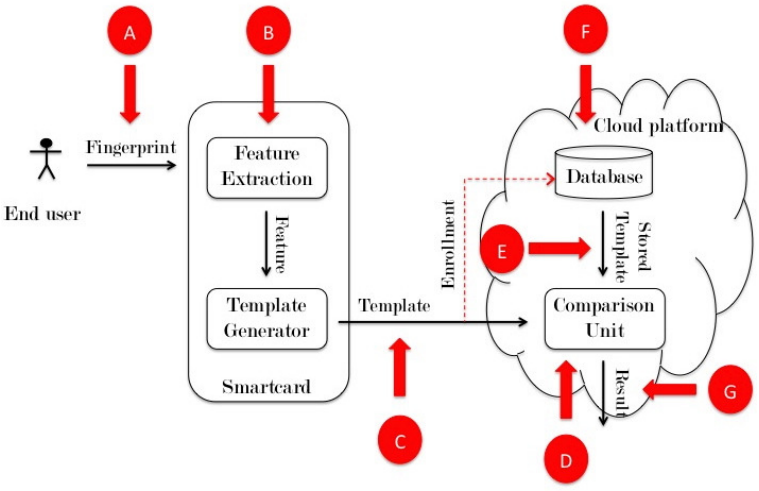

FIGURE 4.4: Ratha's attack model framework [RCB01].

Many researchers discussed security threats inherent to biometric systems (e.g., Ratha *et al.* [RCB01], Bolle *et al.* [BCR02], and Roberts [Rob07]). Figure 4.4 depicts Ratha's

framework that identified a number of points where a biometric system can be attacked. We summarize these attacks in the following :

The biometric data. Fake biometric attack has attracted the greatest publicity.

The measure device. It is possible to make a replay attack by opening the device and using a recorder containing an end user's fingerprint signal.

The feature extractor. It is necessary for the feature extractor to be tamper-proof in order to make impossible to override it.

The link between the smartcard and remote server. It is necessary to secure the transmission channel to avoid the modification of biometric templates.

The comparison unit. As the feature extractor, it is necessary for the comparison unit to be tamper-proof in order to make impossible to override it.

The database. Another obvious possible target for the attacker is the reference database.

The link between the database and the comparison unit. The attacker can intercept the data exchange between the database and the comparison unit, and thus modify the reference template.

The decision. The attacker can override the final decision.

4.3 Related Work

Many researchers pointed security issue of remote biometric authentication and several attempts to addressing them have been made. Basically, we distinguish three main approaches : *Feature transformation*, *Biometric cryptosystem*, and *Homomorphic encryption* [YSK+13].

In this section we give an overview of privacy-preserving techniques involved in each approach to secure remote biometric identification or authentication, and discuss their advantages and disadvantages.

4.3.1 Feature transformation

In this approach, Biometric data are transformed to random data by using a client-specific key to ensure the cancelability and diversity requirements. Feature transformation is practical in performance, but it is no longer secure if the client-specific key is compromised [YSK+13].

Ratha *et al.* [RCB01] introduced the concept of cancelable (i.e., changeable) biometrics to enhance privacy and security. For this purpose, the biometric signal is distorted by a chosen transformation function, and each time the transformed biometric template is compromised, another transformation function is used to generate a novel template. Later, the authors proposed, in [BCR02], a morphing method to transform the biometric signal. In the same line, Jeong *et al.* [JLKC06] proposed changeable biometrics for face recognition using an appearance based approach.

BioHashing is a specific transformation method which uses two-factor authentication approach. Thus, biometric data are combined with pseudo-random number to generate a *BioCode*. Some works have exploited BioHashing techniques. For instance, Goh and Ngo [GL03] and Teoh *et al.* [JLG04b, JL05] on face recognition, Connie *et al.* [CJOL04] on palmprint, and Teoh *et al.* [JLG04a] and Belguechi *et al.* [BRA10, BCRA13] on fingerprint matching. For more details, see [BAC+11].

4.3.2 Biometric cryptosystem

The approach is to use error correcting codes to correct a certain number of errors in a biometric template within a given metric space, by making public some additional information about the enrolled template [BCA+10]. These additional information (called helper data,Vault or Sketch) must not reveal too much information on the original template for an attacker to compromise the system by guessing the biometric template. Since this approach needs to have strong restriction of authentication accuracy, both practical and security issues are controversial [YSK+13].

Biometric cryptosystem includes *fuzzy vault*, *fuzzy commitment*, and *fuzzy extractors*. The first biometric cryptosystem combining error correction codes with biometrics, called *fuzzy commitment*, was designed by Juels and Wattenberg [JW99]. In fuzzy commitment, cryptographic keys are decommitted using biometric data, and the term *fuzzy* implies that a value *close* to the original biometric data can extract the committed value. Juels and Sudan [JS02] proposed an improvement upon the previous work, called *fuzzy Vault schemes*, which are order invariant for the fuzzy commitment scheme, but use a polynomial reconstruction problem based on an error-correction code such as the *Reed–Solomon*.

Basically, *fuzzy extractors* are used to convert biometric data into random strings, which makes it possible to apply cryptographic techniques. Thus, using biometric data as keys permits to encrypt and authenticate users records. Dodis *et al.* [DRS04] introduced two primitives : *secure sketch* and *fuzzy extractor* to securely derive public keys from

shared secrets (biometric data). The public keys are then used for the purposes of authentication. Boyen [Boy04] studied the question of generating keys of cryptographic quality from non uniformly distributed, non perfectly reproducible fuzzy processes, and addressed potential adversarial modification of public keys (possible in [DRS04]) to enable unidirectional authentication from the user to the server without the assumption of a reliable communication channel, and then, to achieve mutual authentication over a completely insecure channel in [BDK+05]. We can also mention the work of : Daugman [Dau04] on iris recognition and Kevenaar *et al.* [KSvdV+05] on face recognition using the Hamming distance as distance metric, Tuyls *et al.* [TAK+05] on fingerprint, and also Tuyls and Goseling [TG04], Dodis *et al.* [DKRS06, DKK+12], Naini and Tonien [ST11] ... etc.

4.3.3 Homomorphic encryption

In this approach, biometric data are protected by homomorphic encryption, and distance metrics such as the Hamming and the Euclidean distances are measured on encrypted biometric data. Both partially homomorphic and fully homomorphic encryption schemes can be used. Homomorphic encryption based approaches enable biometric authentication system to be considerably secure as long as the secret key is securely managed by the trusted party. The performance and the encrypted data size are main issues for the practical use of this approach [YSK+13].

Kershbaum *et al.* [KAMR04] described a secure homomorphic encryption based protocol to solve the problem of comparing fingerprints without actually exchanging them. The algorithm matches fingerprints based on minutiae and the distance metric used is Hamming distance. Schoenmakers and Tuyls [ST06] proposed to use Paillier encryptions [Pai99] based homomorphic encryption schemes for securely converting an integer into its binary representation. Then, by employing multiparty computation tools, the binary representation is used to evaluate securely whether the sample matches a stored (encrypted) biometric template in the server side. Tang *et al.* [TBCP08] proposed a general biometric-based remote authentication scheme by employing a Private Information Retrieval (PIR) protocol and the ElGamal public-key encryption scheme.

Bringer *et al.* [BCI+07] described a biometric-based authentication mechanism, which uses the Goldwasser-Micali encryption scheme to privacy protection of biometric. The Hamming distance was used as the distance metric. The authentication server is composed of three entities that must not collude, and one of them, the matcher (i.e., the comparison unit), learns the computed Hamming distance. In [BCPZ08], Bringer *et al.* proposed a scheme to generate strong biometric secret keys. The specificity of this

scheme is that the secret is the error (between the template captured and the reference biometric data) and not the biometric data itself. Based on the Boneh and Shacham group signature, it guarantees the anonymity of the client towards the server.

Barni *et al.* [BBC+10] proposed a privacy-preserving system for fingerprint-based authentication. For this purpose, they adopted the fingercode representation, and the protocol is entirely based on the use of homomorphic encryption. The similarity evaluation is based on Euclidian distance. Kikuchi *et al.* [KNON10] proposed a homomorphic encryption based method and exploited the useful property of additive homomorphism in public key ciphers to privacy-preserving similarities evaluation. However, they studied two similarities, cosine correlation and Euclidean distance. Shahandashti *et al.* [SSO12] propose a fully private fingerprint matching protocol that compares two fingerprints based on the most widely-used minutia-based fingerprint matching algorithm. They consider Paillier's encryption scheme to calculate Euclidean distance and angular difference. Remark that the common factor among these work is the use of partial homomorphic encryption.

Other tools of secure multiparty computation (SMC) as *oblivious transfers* [Rab05] and *garbled circuits* [Yao86] were also used. Oblivious transfers is a cryptographic primitive that enables a receiver to obtain one out of N elements held by a sender, without learning information about the other elements and without the sender knowing which element has been chosen. Nevertheless, garbled circuits ensure secure two-party computation of any function, once it has been represented as a binary circuit. For instance, we can mention the work of Du and Atallah [DA01], in which they investigated a number of biometric comparison scenarios by employing secure multiparty computation techniques.

In the same line, Sadeghi *et al.* [SSW09] proposed a privacy-preserving face recognition protocol based on the Eigenfaces recognition algorithm and a combination of homomorphic encryption and garbled circuits. The similarity evaluation is based on Euclidian distance. Huang *et al.* [HMEK11] presented a privacy-preserving biometric identification system using homomorphic encryption, oblivious transfer and garbled circuits to calculate similarities based on Euclidian distance. Osadchy *et al.* [OPJM10] designed a face recognition algorithm and proposed an efficient secure face identification system, called SCiFI, with the Paillier scheme and the oblivious transfer protocol. Blanton *et al.* [BG11, BA12] proposed a homomorphic encryption and garbled circuit evaluation based method for a secure two-party protocol for both iris and fingerprint identifications. For this purpose, they use the DGK scheme [DGK08], which is an additively homomorphic encryption with shorter ciphertexts than the Paillier scheme. The Hamming distance (resp. Euclidian distance) was used as the distance metric for iris (resp.fingerprint).

Recently, some implementations of Gentry's scheme for applying it to biometrics are proposed. Yasuda *et al.* [YSK+13] proposed an efficient method to compute the Hamming distance on encrypted data using the homomorphic encryption based on ideal lattices [Gen09b, GH11]. Torres *et al.* [TBS15] implemented a privacy-preserving iris biometric authentication protocol adapted to lattice-based fully homomorphic encryption.

Finally, Atallah *et al.* [AFGT05] proposed a *cryptographic hash* computations based protocol, in which biometric templates are treated as bit strings and subsequently masked and permuted during the authentication process. The comparison of two binary vectors modified following the same random transformation leads then to the knowledge of the Hamming distance. The main advantage of this protocol is to use no consuming cryptographic operations. However, as mentioned above, biometric data are approximately stable. In the same line, Di Crescenzo *et al.* [CGGA05] proposed a rigorous model for the study of approximate data authentication schemes, that are tolerant with respect to errors. The model is suitable for the veification of biometric data in authentication schemes.

4.4 Security Definition for Biometric Authentication

4.4.1 Adversary Model

An adversary is defined by the resources that it has. In the following, we list these resources based on points discussed in Section 4.2.3. We note that an adversay may have any combination of these resources.

Fingerprint (FP). An adversary may obtain end users' fingerprint by extreme measure.

Smartcard (CSC and USC). An adversary may obtain :

1. a cracked version of the smartcard (CSC) and acquire all information that it contains.
2. an uncracked version of the smartcard (USC) and test with a various fingerprints.

Eavesdrop the communication channel (ECC&C and ECC&M). An adversary may eavesdrop the communication channel and :

1. be curious (ECC&C) and learn all information sent between the smartcard and the server.

2. be malicious (ECC&M) and be able to modify information sent between the smartcard and the server.

Eavesdrop the comparison unit (ECU&C). A curious adversary may eavesdrop the comparison unit (ECC&C) and learn information sent between the smartcard and the server. We consider malicious adversary that modify information in the comparison unit as outside of our attack model.

Eavesdrop the database (ED&C). A curious adversary may eavesdrop the server database (ED&C) which contains all information about the end users. He can also eavesdrop the communication channel between the database and the comparison unit. We consider malicious adversary that modify information in the database as outside of our attack model.

4.4.2 Security definition

By security, we mean confidentiality, integrity, and availability of the biometric authentication system. The confidentiality requirements of the system are that an adversary should not be able to learn information about the fingerprint. The integrity of the system requires that an adversary cannot impersonate a client and, the availability requires that an adversary cannot make a user unable to authenticate. We take the same security definitions used in the protocol proposed by Atallah *et al.* [AFGT05].

4.4.3 Summary of Schemes' Security

TABLE 4.3: Security of the Protocol

Resources	Confidentiality	Integrity	Availability
FP	No	Strong	Strong
CSC and ED&C	No secure	No secure	No secure
USC and FP	No secure	No secure	No secure
ECC&M and ED&C	Strong	No secure	No secure
USC	Strong	Strong	No secure
ECC&M	Strong	Strong	No secure
USC and ECU&C	Weak	Weak	No secure
USC and ED&C and ECC&M	No secure	No secure	No secure

Table 4.3 summarizes the adversary's power with various resources. No secure means that the system does not protect this resource against this type of adversary. We assume that the smartcard is the lynchpin of the system. This is preferable to having the biometric be the lynchpin. Because, biometrics can be stolen without the theft being detected, however it is easy to notice the absence of the smartcard.

4.5 A vector partition based approach

We consider the problem of secure comparison of n-bits binary string, which occurs in various areas of information security. In biometrics, we assume that we have a large database of biometric reference templates stored in the cloud. In this work we assume that biometrics have been processed and have representations suitable for biometric matching, i.e., each biometric has been processed by a feature extraction algorithm. It is common practice to represent these biometrics using fingerprint vectors, where the components of a vector correspond to binary or integer values. For simplicity, in the rest of the chapter, we consider the most frequently used binary fingerprints, but most of the ideas presented can be extended to integer valued fingerprints.

4.5.1 Atallah' protocol

Our starting point is the protocol proposed by Atallah *et al.* in [AFGT05]. The protocol uses a sophisticated obfuscating technique where a random vector permutation $\prod$ is applied to the biometric template coupled to an exor with a random vector. This solution satisfies the correctness property when calculating the Hamming distance. For instance, let us consider f_0, f_1 two biometric templates, $\prod$ a fixed random permutation and r a random vector :

$$HD(\prod(f_0 \oplus r), \prod(f_1 \oplus r)) = HD(f_0, f_1) \tag{4.4}$$

However, the main lack of this obfuscating technique is that the server may learn the places in the permutated vectors where elements differ because $\prod$ is fixed over time. The solution was ameliorated to make this scheme secure even for an arbitrarily long sequence of authentication.

A novel approach was proposed which uses a multi-rounds-based authentication. In this approach, the server and the client store a small collection of values, which are recomputed after each round. A round of authentication permits to convince the server that the client has a vector close to vector stored in the database but also to refresh the information. At each round a new random boolean vector and a random permutation are generated. Finally, a decision is taken if the outcome is a match or not a match according to the Hamming distance. We wish now to take a decision not only using the Hamming distance but also from the position of the corrupted bits.

4.5.2 First attempt

The idea that comes is to devide the n-bits biometric template into sub-vectors and, then parallely apply Atallah' protocol to these sub-vectors. If the Hamming distance for a given sub-vector is different from zero then we can conclude that the corrupted bits belong to the sub-vectors. This solution has the same security requirements as the original protocol. However, it not allow us to know with precision the corrupted bit.

Example 4.2. *Let us consider two fingerpints* A *and* B *with a corresponding binary vectors A and B depicted in Table 4.4. We present two possible vector partitions : Test 1 and Test 2. In Test 1, each sub-vector contains* 3*-bits. However in Test 2, the binary vector is devided into 3 sub-vectors of* 4*-bits.*

TABLE 4.4: Binary vector partitions.

Test 1	T_1			T_2			T_3			T_4		
Test 2	T_1			T_2			T_3					
$i =$	1	2	3	4	5	6	7	8	9	10	11	12
A	0	0	1	1	0	0	1	1	0	0	0	1
B	1	0	1	1	0	0	1	1	0	0	0	0

As depicted in Example 4.2, a second key problem is *how to divide the biometric vector?* and *what will be the number of sub-vector and their size?*. Our scheme for secure biometric authentication, in fact, is based on taking this false start as a starting point. The main challenge in making this scheme is to find how to define sub-vectors in order to analyse them and find the corrupted bits.

4.6 A nonadaptative combinatorial group testing based approach

In the first part of this section, we give preliminary knowledge about the techniques used to implement our protocol. The second part outlines the protocol to secure remote biometric authentication in the cloud.

4.6.1 Preliminaries

4.6.1.1 Keyed-hash functions

We now briefly review a cryptographic primitive used in the protocol. The protocol uses keyed-hash functions such as [AFGT05] but not encryption. Cryptographic hash

functions map strings of different lengths to short, fixed-size, outputs. Let K denote an n-dimensional vector space over $GF(2)$. A keyed-hash function $h_k : k \in K, h_k(m) = m'$ is indexed by a key k. In the following, we describe its properties [BCK96] :

- Keyed-hash functions , e.g., MD5 or SHA-1, are primarily designed to be *collision resistant*; hence, $h_k(m_1) = h_k(m_2)$, but $m1 \neq m2$.
- Given key $k \in K$ and message m, it is straightforward to compute $m' = h_k(m)$.
- Unpredictability of the output when parts of the input are unknown : given message m and without knowledge of key k, it is hard to find $h_k(m)$, or given result $h_k(m)$ and without knowledge of key k, it is hard to find message m.
- Independence of input/output : Given (possibly many) pairs of message m and result $h_k(m)$, it is hard to find key k.

All the other operations used in the protocol are inexpensive (only exclusive-or and vector permutation).

4.6.1.2 Nonadaptive Combinatorial Group Testing

Combinatorial group testing (CGT) was originally formulated for testing blood supplies during World War II, with a group test comprising : a tester extracting a few drops from each blood sample in a test set, pooling them together, and testing the mixed sample for the syphilis antigen [Dor43]. This means that if we have a set C of individuals, consists of applying group tests on subsets of C for the purpose of identifying wich members of C are infected (or, more generally, defective in some way). The outcome of a group test reveals only the presence or absence of infection(s) in that group, but a number of group tests exactly identifies all infected members [AFBC08].

A testing scheme that makes all its tests in a single round, with all test sets determined in advance, is said to be *nonadaptive* [GAT05]. We assume there is an upper bound, d, on the number of possible defective bits on the binary fingerprint vector, where $1 \leqslant d < n$. For the case $d \geq 2$, the known randomized CGT schemes utilize $\Theta(d^2 n log\ n)$ random bits (see [ZK00]). In this chapter, we present a simple nonadaptive combinatorial group testing scheme, for the case $d = 1$, for the purpose of securely identifying which is exactly defective bit.

Suppose we compare two binary fingerprint vectors A and B of length n. This means that we have a set C_n containing n pairs of bits (a_i, b_i) to be compared: (a_1, b_1), (a_2, b_2), ..., (a_n, b_n). A Hamming distance between two binary vectors of equal length is the

TABLE 4.5: An illustration of a $n \times t$ matrix

	...	(a_4, b_4)	...	(a_7, b_7)	... n
. .					
T_3		1		0	
. .					
T_6		0		1	
. . t					

number of positions at which the corresponding bits are different. Suppose the Hamming distance between A and B is 1 (only one pair of bits is defective). It is straightforward to design a nonadaptive CGT scheme using $O(log\ n)$ tests to find i, where $a_i \neq b_i$, position of defective pair of bits.

The main idea of the approach is to construct a $n \times t$ binary matrix M, where each column corresponds to a pair (a_i, b_i) and each row corresponds to a test T_j, so that $M[i, j] = 1$ denotes participation of (a_i, b_i) in test T_j and 0 denotes absence (See Table 4.5).

The $n \times t$ matrix M is a d-disjunct [ZK00]. Our algorithm for building a 1-disjunct $n \times t$ matrix M is simply to set each $M[i, j] = 1$ with probability roughly $\frac{1}{d+1} = \frac{1}{2}$. We want to collect a group $\{T_1, T_2, ..., T_t\}$ of t tests, each test T_j is a subset of C_n for $1 \leq j \leq t$. The t tests are for determining which pair of bits is defective, and are as follows :

For $j = 1, 2, ..., t$, the j^{th} test is for the composition of those (a_i, b_i) for which the integer i has a 1 in the j^{th} least significant bit of its binary representation; i.e., a pair of bits (a_i, b_i) is in the j^{th} test if, in the binary representation of the integer i, the j^{th} least significant bit is a 1.

To determine which (a_i, b_i) is defective, the binary representation of integer i is constructed one bit at a time, as follows: For $j = 0, ..., (logn) - 1$ in turn, if the j^{th} computed test matches the template reference then the j^{th} bit of i is 0, and if it does not match then the bit is 1.

Formally, $n \times t$ matrix M is a binary matrix where each column i is the binary representation of $i - 1$:

$$M_{i:} = \{(i-1)_{(2)}\} \tag{4.5}$$

Example 4.3. *To illustrate, consider the case of two binary vector A_8 and B_8, depicted in Table 4.6, which only one pair of bits is corrupted (assume it is the pair (a_6, b_6).*

TABLE 4.6: An illustration of a 8×3 matrix

i	1	2	3	4	5	6	7	8
$(i-1)_{(2)}$	$(1-1)_{(2)}$	$(2-1)_{(2)}$	$(3-1)_{(2)}$	$(4-1)_{(2)}$	$(5-1)_{(2)}$	$(6-1)_{(2)}$	$(7-1)_{(2)}$	$(8-1)_{(2)}$
T_1	0	0	0	0	1	**1**	1	1
T_2	0	0	1	1	0	0	1	1
T_3	0	1	0	1	0	**1**	0	1

The 3 $(= log\ n)$ *tests reveal which item is corrupted, as follows. The 3-bit binary representation of* $(6-1)$ *is* 101*, and the item* (a_6, b_6) *is therefore a part of the tests for bit positions* $1, 3$ *(otherwise the* 2 *corresponding tests would have matched their expected values).*

In order to constitute the sub-vectors, we use a transformation function T which generates a matrix W from a fingerprint binary vector V, and the d-disjunct matrix M. Formally,

$$\begin{aligned} \mathbf{T}: \quad & \mathbb{R}^n \longrightarrow \mathbb{R}^{\frac{n}{2} \times log_2(n)} \\ & V \longrightarrow W \end{aligned}$$

$$W_{ij} = \left(\begin{array}{l} if\ M_{log_2(n)+i+1,h-1} = 1 \\ then\ W_{i,j} = V_{h-1} \end{array} \right) .$$

4.6.2 Protocol

We describe now a general biometric-based authentication scheme, where the biometric template matching can be done through binary string comparison. We first describe the enrollment phase and the verification phase, and then provide some remarks.

The server (in the database and the comparison unit) and the client(in the smartcard) store a small collection of values, which are recomputed after each round. Consequently, information obtained by an eavesdropper during one round of authentication is useless for the next round (no replay attacks are possible). We assume that : f_x and f_{x+1} are n-bits binary vectors, and $\prod_x$ and $\prod_{x+1}$ denote random permutations on $\frac{n}{2}$-bits vectors known only by the client, and r_x, r_{x+1}, s_x, s_{x+1} and s_{x+2} are$\frac{n}{2}$-bits binary vectors generated by the client.

4.6.2.1 The enrollment phase

Before a round of authentication, the server and client store the following values :

The Smartcard has :

- A permutation vector $\prod_x$.
- A set of binary vectors r_x, s_x, and s_{x+1}.

The server has :

- $\forall j : s_x \oplus \prod_x (W^x_{:j} \oplus r_x)$.
- $h_k(s_x)$.
- $h_k(s_x, h_k(s_{x+1}))$.

4.6.2.2 The authentication phase

1. The client uses the smartcard to read a new biometric f_{x+1} and to generate biometric matrix $W^x_{i,j}$, random Boolean vectors r_{x+1} and s_{x+2} and a random permutation $\prod_{x+1}$.
2. The smartcard connects to the terminal and sends to the server the following values :
   - $\forall j \ : \prod_x (W^{x+1}_{:j} \oplus r_x)$, and
   - s_x, and
   - a *transaction information* T that consists of a nonce as well as some other information related to this particular access request (e.g., date, time and IP adress).
3. The server computes the hash h_k of the just-received s_x and checks that it is equal to the previously-stored $h_k(s_x)$.
   - If this check does not match it aborts the protocol.
   - If it does match, then the server computes the exor of s_x with the previously stored $\forall j : s_x \oplus \prod_x (W^x_{:j} \oplus r_x)$ and obtains $\prod_x (W^x_{:j} \oplus r_x)$. Then the server compares between the just-computed $\forall j \ : \prod_x (W^x_{:j} \oplus r_x)$ and the received $\forall j \ : \prod_x (W^{x+1}_{:j} \oplus r_x)$ and then retrieves the corrupted bits. If the outcome is a match, then the server sends $h_k(T)$ to the client. Else the server aborts

but throws away this set of information in order to prevent replay attacks; if the server does not have any more authentication parts, then it locks the account and requires the client to re-register.

4. The client checks that the value sent back from the server matches $h_k(T)$. If the message does not match, the smartcard sends an error to the server. Otherwise, the smartcard sends the server the following information :

   - $\forall j : s_{x+1} \oplus \prod_{x+1}(W_{:j}^{x+1} \oplus r_{x+1})$,
   - $h_k(s_{x+1}, h_k(s_{x+2}))$, and
   - $h_k(s_{x+1})$.

   It also wipes from its memory the reading of fingerprint f_{x+1} and of previous random values r_x and s_x, so it is left with $\prod_{x+1}$, r_{x+1}, s_{x+1}, and s_{x+2}.

5. When the server receives this message it verifies that $h_k(s_x, h_k(s_{x+1}))$ matches the previous value that it has for this quantity and then updates its stored values to : $\forall j : s_{x+1} \oplus \prod_{x+1}(W_{:j}^{x+1} \oplus r_{x+1})$, $h_k(s_{x+1}, h_k(s_{x+2}))$, and $h_k(s_{x+1})$.

We note that the protocol requires three messages exchange in the case of a match and exactly one message exchange in the case of no match. In addition, for every successful authentication the database must update its entry to a new value (to prevent replay attacks). However, it does not require complex cryptographic primitives, but instead relies on cryptographic hashes.

4.7 Experiments and Evaluation

Our experiments consist of two parts. First, comparisons are conducted between biometric vectors where the result is a match. As previously discussed, in this case the protocol requires three messages exchanges between the client and the server. Second, comparisons results are mismatches. In this case, the protocol requires exactly one message exchange. All the experiments are conducted on computers with Intel(R) Core(TM) i5-2450M CPU Quadricore (2.50 GHz, 64 bits, and 8GB RAM) connected through a wireless network. To check the results, every experiment is made ten times and an average value is calculated with suppression of aberrant values. Table 4.7 summarizes the results obtained.

Figures 4.5 and 4.6 depict the computation time in case of match and no match. The computation time is reasonable ($\approx 6 \times 10^{-2} seconds$) and almost equivalent until a vector size of 1024-bytes. After we note that the computation time quadruples whenever we

TABLE 4.7: The experiment's Results on fingerprint vectors with different sizes.

Fingerprint			Match		No match	
n (bits)	n (bytes)	Matrix ($\frac{n}{2} \times log_2 n$)	Comput. Time (seconds)	Exec. Time (seconds)	Comput. Time (seconds)	Exec. Time (seconds)
4	0.5	2×2	3.19×10^{-2}	46.98×10^{-2}	2.15×10^{-2}	46.43×10^{-2}
8	1	4×3	3.41×10^{-2}	45.73×10^{-2}	2.53×10^{-2}	45.47×10^{-2}
16	2	8×4	3.66×10^{-2}	46.37×10^{-2}	2.81×10^{-2}	45.57×10^{-2}
32	4	16×5	3.65×10^{-2}	47.50×10^{-2}	3.40×10^{-2}	46.59×10^{-2}
64	8	32×6	4.85×10^{-2}	48.25×10^{-2}	4.74×10^{-2}	46.89×10^{-2}
128	16	64×7	5.93×10^{-2}	30.25×10^{-2}	5.88×10^{-2}	29.26×10^{-2}
256	32	128×8	7.18×10^{-2}	33.64×10^{-2}	6.81×10^{-2}	32.59×10^{-2}
512	64	256×9	6.75×10^{-2}	16.90×10^{-2}	5.70×10^{-2}	10.93×10^{-2}
1024	128	512×10	6.87×10^{-2}	18.95×10^{-2}	6.46×10^{-2}	13.70×10^{-2}
2048	256	1024×11	7.55×10^{-2}	26.09×10^{-2}	6.21×10^{-2}	21.25×10^{-2}
4096	512	2048×12	12.72×10^{-2}	66.02×10^{-2}	7.94×10^{-2}	37.68×10^{-2}
8192	1024	4096×13	32.06×10^{-2}	1.4350	21.68×10^{-2}	82.57×10^{-2}
16384	2048	8192×14	1.2568	3.8696	61.41×10^{-2}	2.5647
32768	4096	16384×15	4.8118	12.8285	2.5315	8.3848
65536	8192	32768×16	21.2268	61.3883	12.6247	31.9928

doubled the size of the vector. This is due to the fact that the processor does not made the computation on sub-vectors in parallel because of their number that exceeds the parallelism capacity.

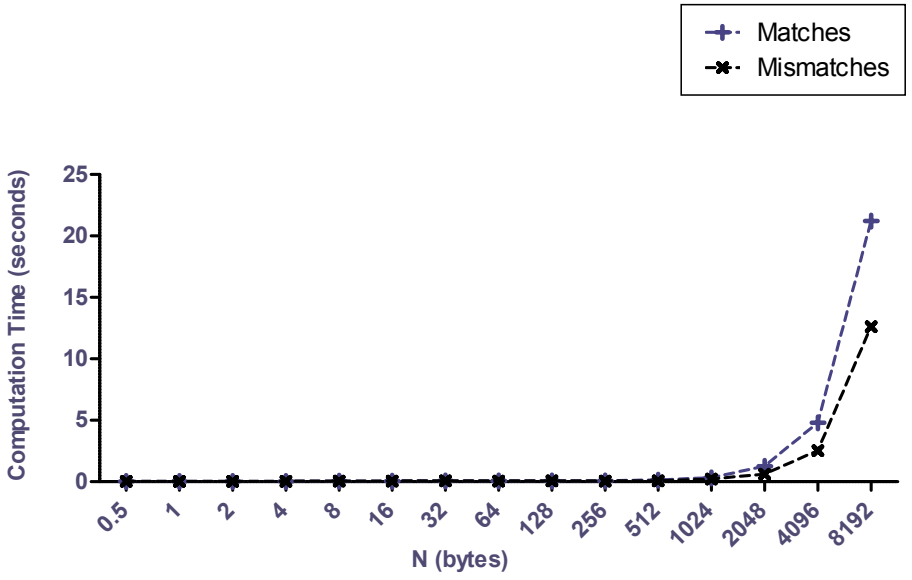

FIGURE 4.5: Computation Time ($n \in [4, 65536]\, bits$).

In Figures 4.7 and 4.8, we show the total execution time, i.e., computation time added to the messages exchanges time, in both match case and no match. The total execution time obtained is very encouraging ($\approx 0,2 second$ for 512 bytes) and almost equivalent until a vector size of 1024-bytes. We note also that it is maximum at 64 bytes. This

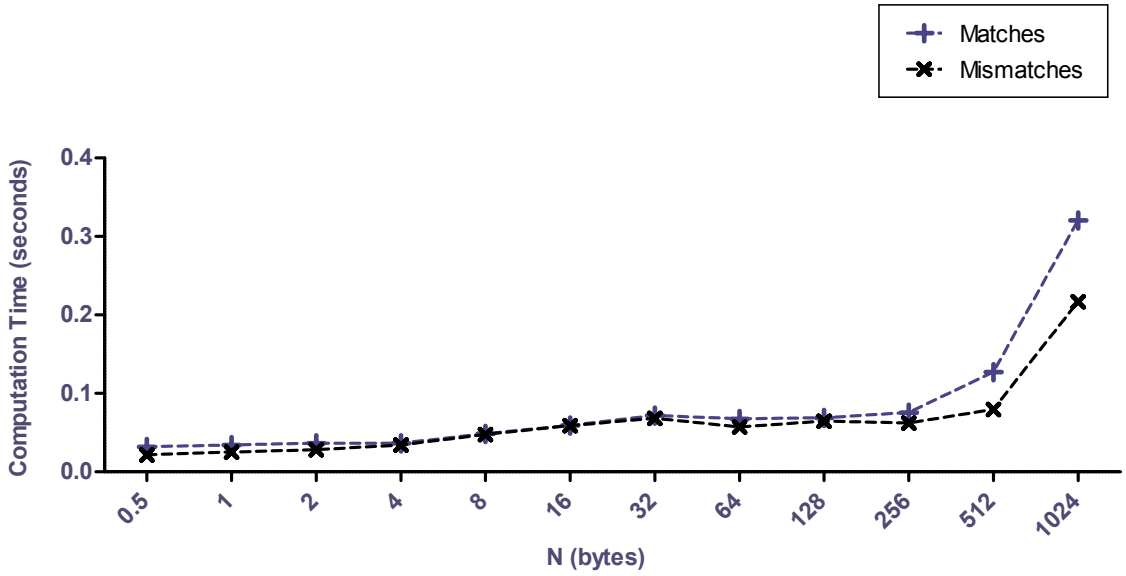

FIGURE 4.6: Computation Time ($n \in [4, 8192]\, bits$).

is principally due to the protocol of communication and the size of the trames when padding is not used. We note also that the computation time quadruples whenever we doubled the size of the vector. This is due to the fact that the communication protocol and the size of the matrice transfered.

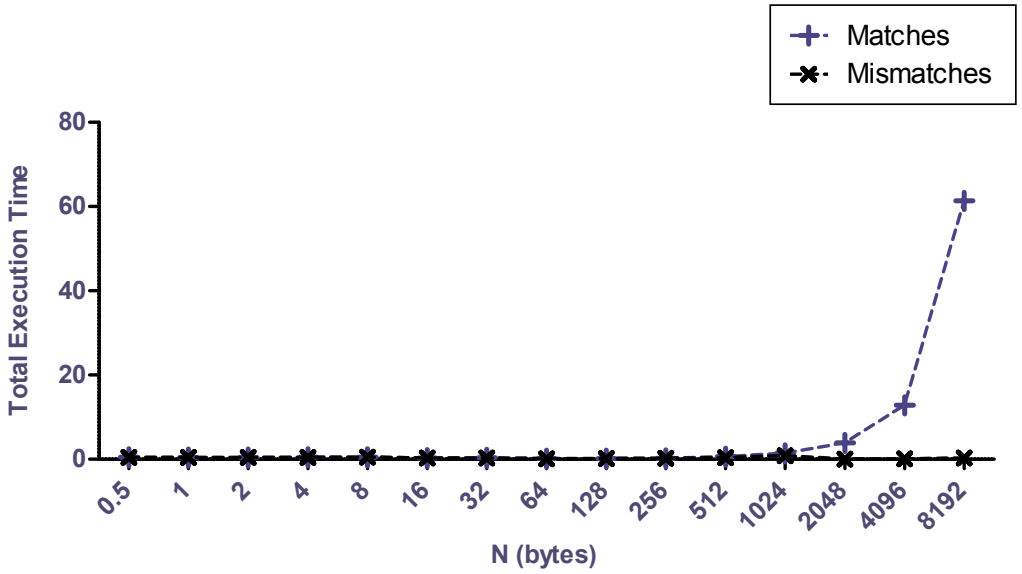

FIGURE 4.7: Total Execution Time ($n \in [4, 65536]\, bits$).

Figures 4.9, 4.10, 4.11 and 4.12 confirm the previously advance statement. Indeed, 90% of the total execution time consists of transfer time on the network according to the size of the matrices used. Relatively stable at the beginning, it declines up considerably to

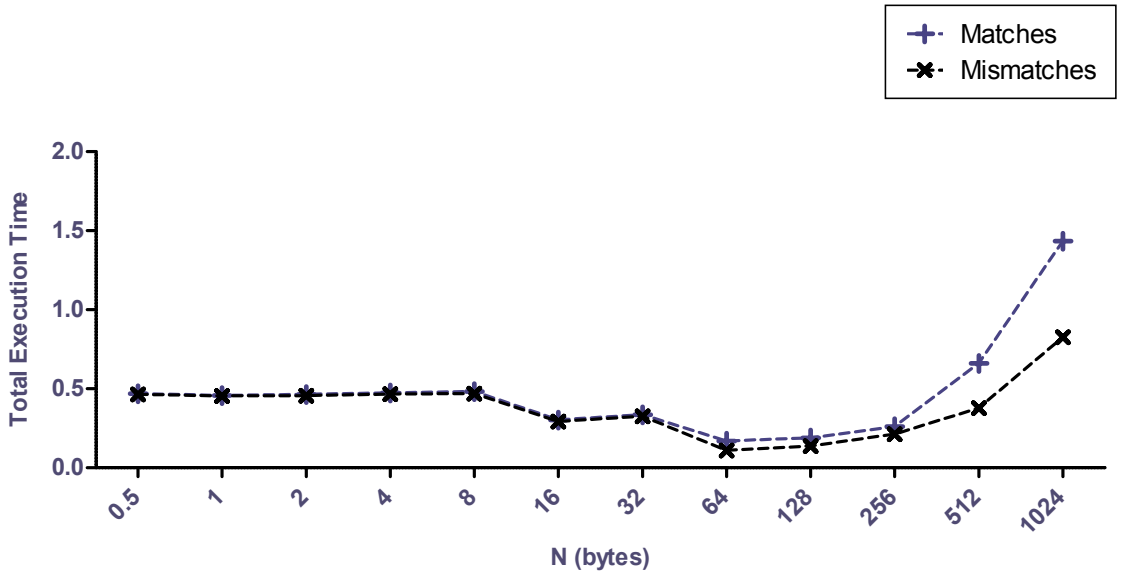

FIGURE 4.8: Total Execution Time ($n \in [4, 8192]\, bits$).

around 64 bytes then ascend pushed upward by data transfer errors which increase the transfer time.

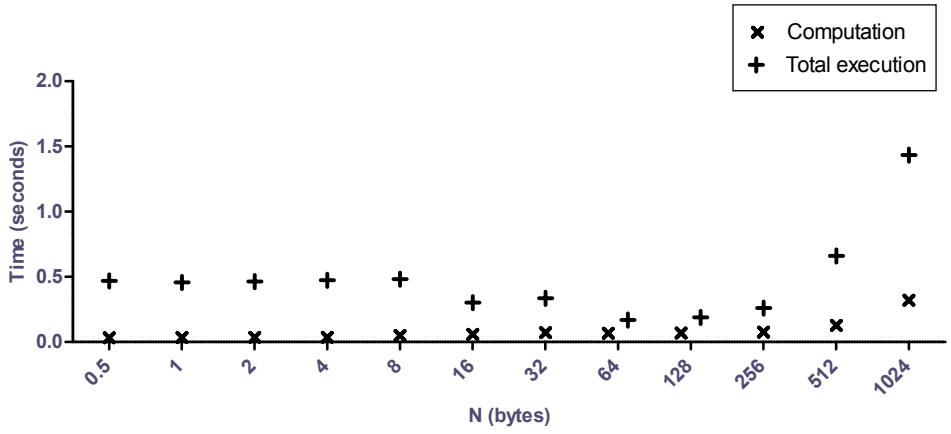

FIGURE 4.9: Match : Computation Vs. Total Execution Time ($n \in [4, 8192]\, bits$).

4.8 Conclusion

In this chapter, we present a lightweight scheme to secure remote biometric authentication that could be used by weak computational devices. The protocol does not require complex cryptographic primitives, but instead relies on cryptographic hashes and obfuscating technique based on vector permutation coupled to exor with random vectors. Additionally, it is hard to impersonate a client, due to the need of the smartcard and either the fingerprint or the server's database. The main problem with our protocol is that

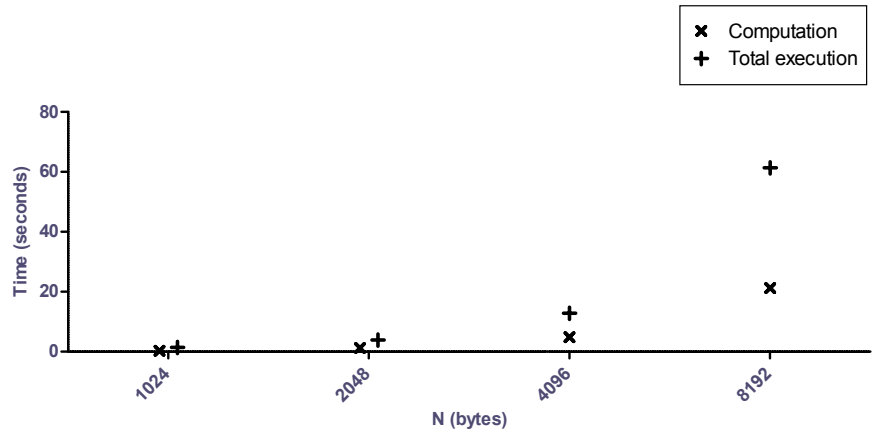

FIGURE 4.10: Match : Total Execution Time ($n \in [8192, 65536]\, bits$).

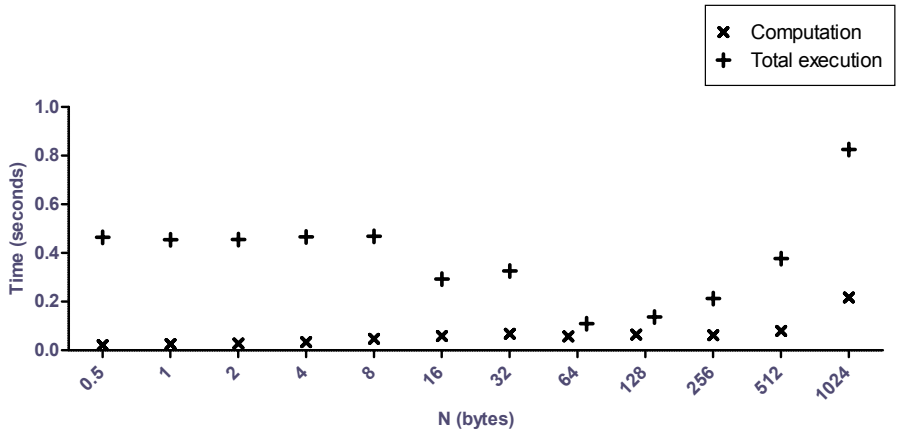

FIGURE 4.11: No match : Computation Vs. Total Execution Time ($n \in [4, 8192]\, bits$).

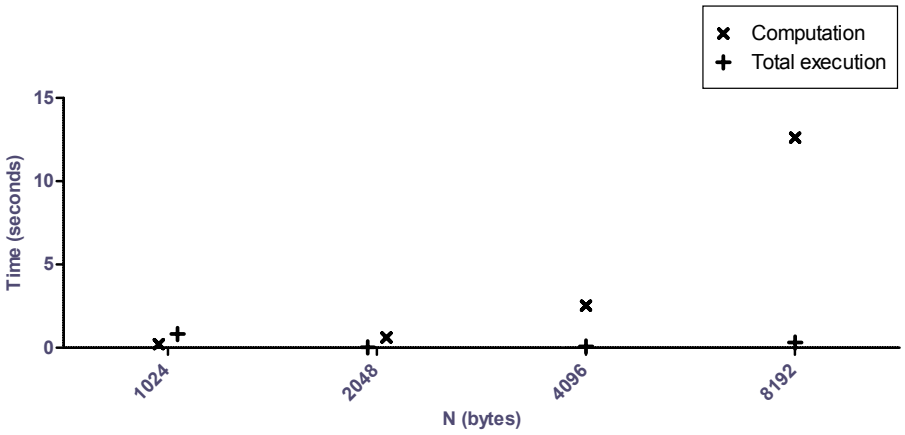

FIGURE 4.12: No match : Computation Vs. Total Execution Time ($n \in [8192, 65536]\, bits$).

it needs three messages exchanges for a match and for every successful authentication the database must update its entry to a new value.

5

Secure Event Management as a Service

Contents

5.1 Introduction

Data stream and sensor based applications are becoming vital in our every-day life ranging from real-time traffic monitoring to emergency response and health monitoring. The volume of incoming data is generally too high to be stored in time and computations on the streams have to be executed on-the-fly to promptly detect interesting events (e.g., car accident detection and notification, network congestion control, network fault management, intrusion detection). But several such isolated events may also have to be monitored globally and jointly detected in order to understand their patterns and correlation relationships, leading to adapt the system behavior and take appropriate actions considering a particular conjunction of events.

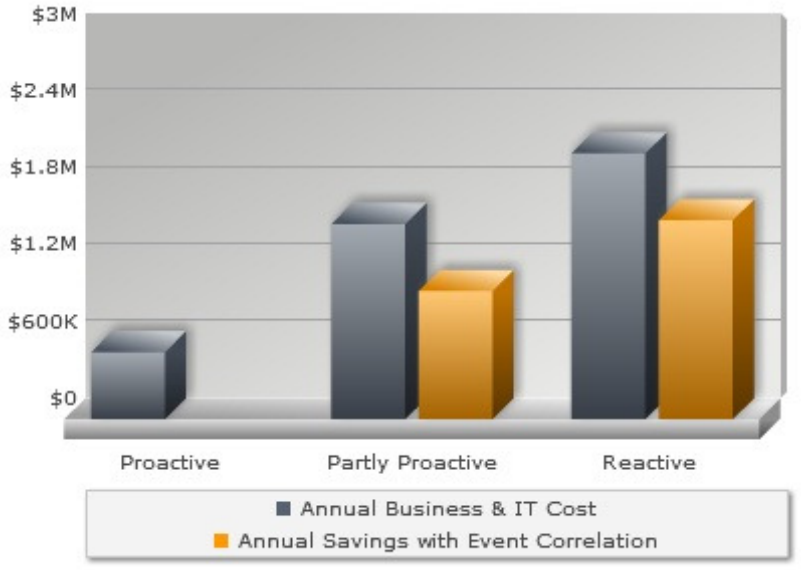

FIGURE 5.1: Cost savings with event management softwares.

For several years, companies have developed event management systems to monitor IT infrastructure which became critical. Event management systems, such as Tivoli Netcool/OMNIbus of IBM, Openview of HP, BMC Event Manager of BMC and intersCope of CA, are characterized by an extremely high-CAPEX coupled with an expensive OPEX. In addition, the immaturity of open source softwares, such as canopsis[1], requires companies to use commercial systems.

Basically, event management systems use an agent based approach for local event correlation in order to increase the scalability and to reduce the network load. Hence, they help to reduce the amount of event messages and make them clearer to a human operator. Event correlation is done manually by operators through correlation rules.

[1]Canopsis (http://www.canopsis.org)

Maintaining and updating these rules is costly. Givin an example of an European company leader in Energy, the CAPEX for monitoring its IT infrastructure containing more than 3000 servers is around 1.5 million €, and the OPEX (updating and adding new correlation rules) is 400.000 € per year. Figure 5.1 shows the gains of the company after the establishment of event management software : 1.5 million € per year (*simulated with Gartner estimates*).

With the current commercialized event management softwares from BMC, HP, CA or IBM, operators are required to generate manually a correlation rule for each category of events to display alerts. They are required as well to keep the rules' list up to date to achieve optimal monitoring of the IT infrastructure. Those correlation rules take as input heterogeneous event coming from different monitoring tools. However, the main obstacle to the broad adoption of such systems remains a high-CAPEX and OPEX.

In such context, SOMONE plans to propose an Event Management as a Service (EMaaS) shared between several SMEs to *(i)* reduce CAPEX and OPEX, and *(ii)* generalize the use of such event management tools. However, transferring and treating IT events in the cloud can be considered, by IT directors, as a breach of security. Indeed, IT events often contain sensitive data about IT infrastructure of companies like : IP adresses, host names, alerts ... etc. To this end, secure protocols should be implemented to ensure the confidentiality and integrity of IT events in the cloud.

In this chapter, we give an overview of our solution to secure event management service in the cloud. The rest of the chapter is organized as follows : In Section 5.2, we give an overview of related work on IT monitoring and data stream management, and we present the structure of an event management software in Section 5.3. Section 5.4 outlines our protocol to secure complex event processing and Section 5.6 concludes the chapter.

5.2 Related Work

In this section we give an overview of related work on IT monitoring and data stream management.

5.2.1 Data Stream Management

In many applications, data may take the form of continuous data streams, rather than static and finite stored data sets. Several aspects of data management have been reconsidered for handling data streams, offering new research directions for the database community. Some applications require knowledge of complex aggregates, Gehrke *et*

al. [GKS01] proposed single-pass techniques for approximate computation of correlated aggregates over both landmark and sliding window views of a data stream of tuples, using a very limited amount of space. Aurora [ACÇ+03a, ACÇ+03b] is a general-purpose data stream manager that is being designed and implemented to efficiently support a variety of real-time monitoring applications track data from numerous streams, filtering them for signs of abnormal activity, and processing them for purposes of filtering, aggregation, reduction, and correlation. Cuzzocrea and Chakravarthy [CC08] presented an event-based data stream compression and mining model by identifying *interesting* events occurring in the unbounded stream. In [BW01], the authors specified a general and flexible architecture for query processing in the presence of data streams and use it as a tool to clarify alternative semantics and processing techniques for continuous queries ; and in [BBD+02] they isolated a number of issues that arise when considering data management, query processing, and algorithmic problems in the setting of continuous data streams. After, they suggested a general architecture for a Data Stream Management System (DSMS). Dobra *et al.* [DGGR02] relied on randomizing techniques that compute small *sketch* summaries of the streams that can then be used to provide approximate answers to aggregate SQL queries over continuous data streams with limited memory and provable guarantees on the approximation error. Olston *et al.* [OJW03] proposed a technique for reducing the overhead incurred to monitor continuous queries over distributed data sources continuously stream. Users register continuous queries with precision requirements at the central stream processor, filters are installed to minimize stream rates while guaranteeing that all continuous queries still receive the updates necessary to provide answers of adequate precision at all times. Wu *et al.* [WSZ04] presented a new approximate approach for automatic online subsequence similarity matching over massive data streams. Paper [GJSS09] described DataDepot, a tool for generating warehouses from streaming data feeds, designed to automate the ingestion of streaming multi-sources data and to maintain complex materialized views over these sources.

Today, we face a large amounts of data spread over many physically distributed nodes because it impractical to send all the data to one central node for query processing and, finding distributed icebergs is a problem that arises commonly in practice. Zhao *et al.* [ZLOX10] presents a novel algorithm with accuracy guarantee and communication costs are independent of the way in which element counts are split amongst the nodes. The algorithm works even when each distributed data set is a stream.

Event correlation plays also a crucial role in network management systems. Vaarandi [Vaa02] presented a free platform independent tool called sec for correlating network management events locally at an agent's side. In [Al-01], the author presented a dynamic group management framework based on IP multicast to support scalable distributed event

monitoring. The framework uses the event correlation information to dynamically reconfigure the multicast group formation (i.e., join and leave). Cranor *et al.* [CJSS03] developed *Gigascope*, a stream database for network applications including traffic analysis, intrusion detection, router configuration analysis, network research, network monitoring, and performance monitoring and debugging. Gigascope is undergoing installation at many sites within the AT&T network, including at OC48 routers, for detailed monitoring. Monitoring aggregates on IP traffic data streams is a compelling application for data stream management systems. The need for exploratory IP traffic data analysis naturally leads to posing related aggregation queries on data streams, that differ only in the choice of grouping attributes. Zhang *et al.* [ZKOS05] address this problem of efficiently computing multiple aggregations over high speed data streams, based on a two-level LFTA/HFTA DSMS architecture, inspired by Gigascope.

5.2.2 IT Monitoring and Event Management

The term *monitoring* has been widely used in many disciplines and in particular in IT infrastructure and software design and engineering. Depending on a particular purpose of the designed system, on the role the monitoring process plays in the system life-cycle, and the kind of information being collected, the definition of the monitoring problem has different interpretations. In a broad sense, monitoring may be defined as a process of collecting and reporting relevant information about the execution and evolution of business processes. This general definition becomes more concrete and clear when the monitoring goals are considered. Monitoring may be used to discover problems in the business process execution. In this case monitoring may be defined as a problem of observing the behavior of a system and determining if it is consistent with a given specification [DGR04].

There are a lot of works addressing the monitoring of business processess for different types of requirements range from behavior, to information, to events. Grigori *et al.* [GCC+04] presented a set of integrated tools that support business and IT users in managing process execution quality by providing several features, such as analysis, prediction, monitoring, control, and optimization. We can also cite the work in [BGG04], it is proposed a smart monitor for web service composition specified as BPEL processes against contracts expressed as assertions, in [MS07] the monitoring is based on event calculus. Mallick *et al.* [Mal11, MHD12] provided a new modelling approach to the problem of resource prediction in virtualized systems. Models are based on historical data to forecast shortterm resource usages. Fan *et al.* [FX14, FBXS14] proposed a differential privacy-based technique for privacy-preserving monitoring web browsing.

In IT infrastructure Monitoring, several open source projects exist. The most popular is Nagios[2] [Gas07]. Nagios watches hosts and services, alerting users when things go wrong and again when they get better. Shinken[3] project consists of a complete overhaul of Nagios core in Python, giving it new architecture, more flexible and easier to maintain than the current monolithic daemon of Nagios. OpenNMS[4] is an enterprise grade network monitoring and network management platform with the goal to be a truly distributed, scalable for all aspects of the FCAPS network management model. Zabbix[5] is a network management system. It is designed to monitor and track the status of various network services, servers, and other network hardware. Some monitoring tools are marketed by companies like : Tivoli Netcool/OMNIbus of IBM, Openview of HP, BMC Event Manager of BMC and interscope of CA.

5.3 Preleminaries

In the following we give the example of Netcool/Omnibus event management software marketed by HP. The database (in memory), called ObjectServer, is installed in a central point of the IT infrastructure and the probes are installed in servers. Probes send IT events to ObjectServer with a given frequence defined by operators.

The ObjectServer provides an SQL interface for defining and manipulating relational database objects such as tables and views. The ObjectServer SQL commands include :

- Data Definition Language (DDL) commands to create, alter, and drop database objects.
- Data Manipulation Language (DML) commands to query and manipulate data in existing database objects.
- System commands to alter the configuration of an ObjectServer.
- Session control commands to alter settings in client sessions.
- Security commands to control user access to database objects.

5.3.1 Database Schema

The ObjectServer of Netcool/Omnibus consists in a set of databases :

[2] Nagios : The Industry standard in IT infrastructure Monitoring.
[3] Shinken.
[4] OpenNMS.
[5] Zabbix : An Enterprise-Class Open Source Distributed monitoring solution.

alerts. Alert data, and event list configuration.

catalog. System catalog containing Object Server metadata (can be viewed but not modified).

custom. Database for tables added by users.

iduc_system. Channel setup for accelerated event notification (AEN).

master. Compatibility with previous releases; Desktop ObjectServer tables.

persist. Triggers, procedures and signals.

precision. Tables for integration with IBM Tivoli Network Manager.

security. Authentication information for users, roles, groups, permissions.

service. Service.status table for service display (used mostly with monitors).

tools. User tool and menu structure.

transfer. Used by the ObjectServer gateways.

Each database consists in a set of tables and attributes. The most important in our work is *Alerts*. It consists in a set of tables. Table 5.1 depicts the *alerts* database that contains all alerts sent by probes.

TABLE 5.1: Netcool/Omnibus ObjectServer : alerts database.

application_types
backup_states
col_visuals
colors
conversions
details
iduc_messages
journal
login_failures
objclass
objmenuitems
objmenus
problem-events
resolutions
status

Table 5.2 depicts the status table which contains all information about an alert/event.

TABLE 5.2: Netcool/Omnibus ObjectServer : alerts database.

Name	Data Type	Length
Acknowledged	Integer	4
Agent	VarChar	64
AlertGroup	VarChar	255
AlertKey	VarChar	255
BSM_Identity	VarChar	1024
Class	Integer	4
Customer	VarChar	64
EventId	VarChar	255
ExpireTime	Integer	4
ExtendedAttr	VarChar	4096
FirstOccurrence	UTC	4
Flash	Integer	4
Grade	Integer	4
Identifier	VarChar	255
InternalLast	UTC	4
LastOccurrence	UTC	4
LocalNodeAlias	VarChar	64
LocalPriObj	VarChar	255
LocalRootObj	VarChar	255
LocalSecObj	VarChar	255
Location	VarChar	64
Manager	VarChar	64
NmosCauseType	Integer	4
NmosDomainName	VarChar	64
NmosEntityId	Integer	4
NmosEventMap	VarChar	64
NmosManagedStatus	Integer	4
NmosObjlnst	Integer	4
NmosSerial	VarChar	64
Node	VarChar	64
NodeAlias	VarChar	64
OldRow	Integer	4

continued on next page

continued from previous page

OwnerGID	Integer	4
OwnerUID	Integer	4
PhysicalCard	VarChar	64
PhysicalPort	Integer	4
PhysicalSlot	Integer	4
Poll	Integer	4
ProbeSubSecondId	Integer	4
ProcessReq	Integer	4
RemoteNodeAlias	VarChar	64
RemotePriObj	VarChar	255
RemoteRootObj	VarChar	255
RemoteSecObj	VarChar	255
RowID	Unsigned64	8
RowSerial	Incr	4
Serial	Incr	4
ServerName	VarChar	64
ServerSerial	Integer	4
Server	VarChar	64
Severity	Integer	4
StateChange	UTC	4
Summary	VarChar	255
SuppressEscl	Integer	4
Tally	Integer	4
TaskList	Integer	4
Type	Integer	4
URL	VarChar	1024
X733CorrNotif	VarChar	255
X733CorrType	Integer	4
X733ProbableCause	Integer	4
X733SpecificProb	VarChar	64

5.3.2 Data manipulation language

The Netcool/Omnibus ObjectServer provides a classical SQL langage to query its databases. For instance, to insert a new alert in the database :

```
insert into alerts.status (Identifier, Node, Manager, Severity, AlertGroup, Summary)
```

values ('Freebox0184', 'Freebox', 'SomoneProbe', 5, 'Network Problem', 'Deconnexion du reseau Free');

5.3.3 Complex Event Processing

Complex Event Processing (CEP) in the context of "IT monitoring" permits to transform into alerts, IT events coming from different probes. Basically, it consists on a rule engine coupled with a collection of processing rules, which are used to process IT events. For instance, to update the Severity of an alert in the database :

update alerts.status set Severity = 0, Summary = 'Discarded'
where Severity = 5 and Node = 'Freebox';

5.4 Encryption-based Anonymization for Complex Event Processing

In this section, we give an overview of the architecture proposed to secure TeeM, a complex IT event processing as a service.

5.4.1 TeeM Architecture

The main objective of the project is first to ensure a high level of confidentiality and integrity of IT events produced by monitoring tools (i.e., Nagios) during the transfer and treatment to the event management software (i.e., Netcool IBM) hosted in the cloud ; and second to secure the cloud platform installed on the OVH Datacenter and the frontend access against external attacks.

5.4.1.1 Client side

- Potential customers of Teem solution should have an IT infrastructure monitored by Nagios, and containing an LDAP or another access control server.
- The transfer of customer IT events to the cloud platform is done using the Nagios plug'in NTx (Nagios To x) developed by Somone.
- xTx (a ZeroMQ[6] based bus) developed by somone is installed in the Nagios Server.

[6] ZeroMQ

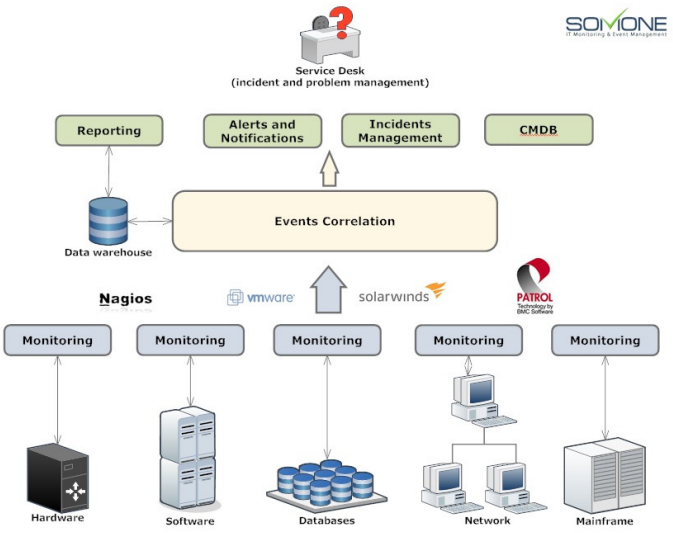

FIGURE 5.2: TeeM SaaS Project Architecture.

- The plug'in xTN (x to Netcool) uses the Syslog protocol. Thus, Nagios events will be formatted into Syslog messages and transfered to the cloud platform.
- We chose this solution because of the availability of a Syslog Probe on Netcool and the ease to implement the xTN plug'in based on Nagios Syslog.
- The customer is responsible for the security of its perimeter.

5.4.1.2 Server side

- The Object Server and Syslog probe is installed on the same Linux server hosted by OVH.
- The probe listens on a given port the arrival of IT events in Syslog format.
- The probe transmits the Syslog events to the Object Server for processing.
- The remote server administration is done via an SSH client.
- SOMONE is responsible for the security of the remote server.

5.4.2 Event Encryption

We distinguish two kinds of IT events' attributes : identifying attributes and non-identifying attributes. The non-identifying attributes stay in clear without modification. This is due to the fact that mathematical operations are basically done on these attributes. For instance, we can mention :

1. Operations : math and string operation, binary comparison operations, list comparison operations, and logical operations.
2. Functions.
3. Expressions.
4. Conditions.

However, the attributes that identify an IT event must be encrypted to ensure their anonymity. There are several encryption techniques outlined and compared in Section 2.3. One can also cite keyed-hash functions discussed in Section 4.6.1.

In our project, we need a reversible encryption function (i.e., not a one-way function), because we must decipher the identifying attributes after the treatment in the server side in order to identify the provenance of the alerts. Thus, keyed-hash functions are unusable. We have the choice between symmetric and asymmetric encryption schemes. According to the comparison given in Table 2.4, using a symmetric encryption scheme as AES is more efficient than an asymmetric scheme. In addition, key management issue does not arise in our case because we have n clients that exchange with one remote server. Therefore, only n key pairs are required.

5.4.3 Query Rewriting

Our approach is based on lightweight agile parsing techniques supported by the TXL source transformation system. TXL [Cor06] is a special-purpose programming language designed to provide rule-based source transformation using functional specification and interpretation. TXL programs have two main parts : a context-free grammar that describes the syntactic structure of inputs to be transformed, and a set of context-sensitive, example-like transformation rules organized in functional programming style.

TXL operates in three phases : parse, transform, and unparse.

1. The parsing phase creates an internal representation of the input as a parse tree under control of a context-free grammar.
2. The transformation phase transforms the parse trees created by the parser under control of a set of example-like transformation rules.
3. The unparsing phase unparses the transformed parse tree to text output with standard spacing and pretty-printing under control of the grammar.

We use the SQL grammar in order to define the identifying attributes and values in SQL queries. Then, we add a transformation rule which consists in an encryption function to encrypt identifying values using the right encryption key. Thus, SQL queries will be compiled in order to take into account the modified events and inserted in the object server.

5.4.4 Alerts Display

The question now is how to display anonymized alerts ? We have two possibilities : using a web interface or a mobile application. In the two cases, the web server and the web service are in the server side, i.e., Identification information can not be decrypted. Therefore, a plugin is necessary in the web browser (in the client side) to permit to decipher identification values and identify the origin of an alert. In the second case, the mobile application should integrate a mechanism to store the key in order to decipher alerts.

5.5 Security of the protocol

Theorem 5.1. *The protocol is as secure as the symmetric scheme used to cipher the identification information.*

Proof. Until the client arrives to guarantee the confidentiality of the encryption/decryption pair key, an attacker cannot decipher an IT event and infer its sensitive information. In addition, events are treated in the server side as they are provided by the client and are never decipher outside its security perimeter. However, the protocol is not proven secure against brute force attacks, and as depicted in Table 2.4 the key pair should be modified each 2 years. □

5.6 Conclusion

In this chapter, we provided an encryption-based anonymization approach to secure complex event processing. The field of use of this approach is IT monitoring.

6

Conclusions and Future Work.

In the previous chapters, various security issues in the context of cloud computing and particularly in the BPaaS delivery model was adressed and protocols to secure data and services proposed. In this chapter, we summarize the contributions of the thesis and identify directions for future work.

Summary

In this dissertation, we have used different approaches for the securing of sensitive data and service reusing in the cloud. We have also provided a survey on computer security and an overview on remote biometric authentication. In particular, our main research contributions are :

Security in cloud computing. We studied the main existing security mechanisms towards a survey which we consider as a toolbox for various security issues in the cloud.

Secure design by selection. We have introduced the concept of *privacy by design* in the context of business processes design. Particularly when sharing, reusing and composing process fragment in the BPaaS delivery model. For this purpose, we investigated the security issues when developing a new process-based application in BPaaS. First, we proposed an anonymization-based approach to preserve the business activities of an organization. However, we demonstrated that it is not sufficient to guarantee availability for process fragments reuse in BPaaS. For that, we extended it with the vision of diverse view of multi-tenants BPaaS. Furthermore, we presented the costs of both confidentialty and availability to be ensured at BPaaS level when reusing fragments.

Lightweight and secure biometric authentication in the cloud. we presented a lightweight scheme to secure remote biometric authentication that could be used by weak computational devices. This protocol does not require complex cryptographic primitives, but instead relies on cryptographic hashes and obfuscating technique based on vector permutation coupled to exor with random vectors. Additionally, it is hard to impersonate a client, due to the need of the smartcard and either the fingerprint or the server's database. However, the main problem with our protocol is that it needs three messages exchanges for a match and for every successful authentication the database must update its entry to a new value.

Secure event management as a service. We proposed an encryption-based anonymization approach to secure multi-party complex event processing in the cloud. The proposed approach is the implemented in the context of IT Event Management as a Service

Future Directions

The proposed work could be enhanced as follows :

1. It will be interesting to extend the approach of *secure design by selection* to all cloud layers, therefore when designing process-based applications, the infrastructure or the platform will also be securely selected.

2. It would also be interesting to use the *nonadaptative combinatorial group testing* approach coupled with another encryption method as keyed hash. This solution will permit us to avoid to update data in the server side at each authentication round.

3. Another direction would be to anonymize the biometric authentication. Indeed, the server knows who is the user that authenticates himself. For this purpose, it will be interesting to hide this information to the server.

4. As a part of complex event processing, NoSQL databases have emerged. We think to generalise the proposed protocol to the context of NoSQL databases and Big Data.

Bibliography

[20095] Council Directive 2002/58/EC. On the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal L 281*, 1995.

[AB09] Mikhail J. Atallah and Marina Blanton. *Algorithms and Theory of Computation Handbook*. Chapman & Hall/CRC, 2nd edition, 2009.

[ACÇ+03a] Daniel J. Abadi, Donald Carney, Ugur Çetintemel, Mitch Cherniack, Christian Convey, C. Erwin, Eduardo F. Galvez, M. Hatoun, Anurag Maskey, Alex Rasin, A. Singer, Michael Stonebraker, Nesime Tatbul, Ying Xing, R. Yan, and Stanley B. Zdonik. Aurora: A data stream management system. In Halevy et al. [HID03], page 666.

[ACÇ+03b] Daniel J. Abadi, Donald Carney, Ugur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stanley B. Zdonik. Aurora: a new model and architecture for data stream management. *VLDB J.*, 12(2):120–139, 2003.

[ACKM04] Gustavo Alonso, Fabio Casati, Harumi A. Kuno, and Vijay Machiraju. *Web Services - Concepts, Architectures and Applications*. Data-Centric Systems and Applications. Springer, 2004.

[AFBC08] Mikhail J. Atallah, Keith B. Frikken, Marina Blanton, and YounSun Cho. Private combinatorial group testing. In Masayuki Abe and Virgil D. Gligor, editors, *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2008, Tokyo, Japan, March 18-20, 2008*, pages 312–320. ACM, 2008.

[AFG+09] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel

Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, February 2009.

[AFGT05] Mikhail J. Atallah, Keith B. Frikken, Michael T. Goodrich, and Roberto Tamassia. Secure biometric authentication for weak computational devices. In Patrick and Yung [PY05], pages 357–371.

[AFK+05] Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. Anonymizing tables. In *ICDT*, pages 246–258, 2005.

[AFN05] Karl Aberer, Michael J. Franklin, and Shojiro Nishio, editors. *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*. IEEE Computer Society, 2005.

[AFS08] Francesco Maria Aymerich, Gianni Fenu, and Simone Surcis. An approach to a cloud computing network. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the, 4-6 Aug. 2008, Ostrava*, pages 113–118. IEEE Computer Society, 2008.

[AK92] Ibrahim A. Al-Kadi. The origins of cryptology: The arab contributions. *Cryptologia*, 16(2):97—-126, 1992.

[AKB11] Sami Alsouri, Stefan Katzenbeisser, and Sebastian Biedermann. Trustable outsourcing of business processes to cloud computing environments. In Pierangela Samarati, Sara Foresti, Jiankun Hu, and Giovanni Livraga, editors, *5th International Conference on Network and System Security, NSS 2011, Milan, Italy, September 6-8, 2011*, pages 280–284. IEEE, 2011.

[AKL+09] Tobias Anstett, Dimka Karastoyanova, Frank Leymann, Ralph Mietzner, Ganna Monakova, Daniel Schleicher, and Steve Strauch. Mc-cube: Mastering customizable compliance in the cloud. In Luciano Baresi, Chi-Hung Chi, and Jun Suzuki, editors, *Service-Oriented Computing, 7th International Joint Conference, ICSOC-ServiceWave 2009, Stockholm, Sweden, November 24-27, 2009. Proceedings*, volume 5900 of *Lecture Notes in Computer Science*, pages 592–606, 2009.

[Al-01] Ehab Al-Shaer. A dynamic group management framework for large-scale distributed event monitoring. In *2001 IEEE/IFIP International Symposium on Integrated Network Management, IM 2001, Seattle, USA, May 14-18, 2001. Proceedings*, pages 361–374, 2001.

[ALMS09] Tobias Anstett, Frank Leymann, Ralph Mietzner, and Steve Strauch. Towards BPEL in the cloud: Exploiting different delivery models for the execution of business processes. In *2009 IEEE Congress on Services, Part I, SERVICES I 2009, Los Angeles, CA, USA, July 6-10, 2009*, pages 670–677. IEEE Computer Society, 2009.

[AM07] Eyhab Al-Masri and Qusay H. Mahmoud. Qos-based discovery and ranking of web services. In *Proceedings of the 16th International Conference on Computer Communications and Networks, IEEE ICCCN 2007, Turtle Bay Resort, Honolulu, Hawaii, USA, August 13-16, 2007*, pages 529–534. IEEE, 2007.

[AM08] Eyhab Al-Masri and Qusay H. Mahmoud. Investigating web services on the world wide web. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 795–804. ACM, 2008.

[Amb97] Andris Ambainis. Upper bound on communication complexity of private information retrieval. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Automata, Languages and Programming, 24th International Colloquium, ICALP'97, Bologna, Italy, 7-11 July 1997, Proceedings*, volume 1256 of *Lecture Notes in Computer Science*, pages 401–407. Springer, 1997.

[ASKW11] Ahmed Awad, Sherif Sakr, Matthias Kunze, and Mathias Weske. Design by selection: A reuse-based approach for business process modeling. In Manfred A. Jeusfeld, Lois M. L. Delcambre, and Tok Wang Ling, editors, *Conceptual Modeling - ER 2011, 30th International Conference, ER 2011, Brussels, Belgium, October 31 - November 3, 2011. Proceedings*, volume 6998 of *Lecture Notes in Computer Science*, pages 332–345. Springer, 2011.

[AW89] Nabil R. Adam and John C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Comput. Surv.*, 21(4):515–556, 1989.

[BA12] Marina Blanton and Mehrdad Aliasgari. Secure outsourced computation of iris matching. *Journal of Computer Security*, 20(2-3):259–305, 2012.

[BAC+11] Rima Belguechi, Vincent Alimi, Estelle Cherrier, Patrick Lacharme, and Christophe Rosenberger. An overview on privacy preserving biometrics. *Recent Application in Biometrics, INTECH*, pages 65–84, 2011.

[BB84] Charles H. Bennett and Gilles Brassard. An update on quantum cryptography. In Blakley and Chaum [BC85], pages 475–480.

[BBA12] Mehdi Bentounsi, Salima Benbernou, and Mikhail J. Atallah. Privacy-preserving business process outsourcing. In Carole A. Goble, Peter P. Chen, and Jia Zhang, editors, *2012 IEEE 19th International Conference on Web Services, Honolulu, HI, USA, June 24-29, 2012*, pages 662–663. IEEE, 2012.

[BBA16] Mehdi Bentounsi, Salima Benbernou, and Mikhail J. Atallah. Security-aware business process as a service by hiding provenance. *Computer Standards & Interfaces*, 44:220–233, 2016.

[BBC+10] Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Fabio Scotti, and Alessandro Piva. Privacy-preserving fingercode authentication. In Patrizio Campisi, Jana Dittmann, and Scott Craver, editors, *Multimedia and Security Workshop, MM&Sec 2010, Roma, Italy, September 9-10, 2010*, pages 231–240. ACM, 2010.

[BBC+13] Djamal Benslimane, Mahmoud Barhamgi, Frédéric Cuppens, Franck Morvan, Bruno Defude, Ebrahim Nageba, François Paulus, Stephane Morucci, Michael Mrissa, Nora Cuppens-Boulahia, Chirine Ghedira, Riad Mokadem, Said Oulmakhzoune, and Jocelyne Fayn. PAIRSE: a privacy-preserving service-oriented data integration system. *SIGMOD Record*, 42(3):42–47, 2013.

[BBD+02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In Lucian Popa, Serge Abiteboul, and Phokion G. Kolaitis, editors, *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*, pages 1–16. ACM, 2002.

[BBDA12] Mehdi Bentounsi, Salima Benbernou, Cheikh S. Deme, and Mikhail J. Atallah. Anonyfrag: an anonymization-based approach for privacy-preserving bpaas. In Jérôme Darmont and Torben Bach Pedersen, editors, *1st International Workshop on Cloud Intelligence (colocated with VLDB 2012), Cloud-I '12, Istanbul, Turkey, August 31, 2012*, page 9. ACM, 2012.

[BC85] G. R. Blakley and David Chaum, editors. *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*. Springer, 1985.

[BCA+10] Duncan Bayly, Maurice Castro, Arathi Arakala, Jason Jeffers, and Kathy J. Horadam. Fractional biometrics: safeguarding privacy in biometric applications. *Int. J. Inf. Sec.*, 9(1):69–82, 2010.

[BCGP92] Tim Berners-Lee, Robert Cailliau, Jean-François Groff, and Bernd Pollermann. World-wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2):74–82, 1992.

[BCI+07] Julien Bringer, Hervé Chabanne, Malika Izabachène, David Pointcheval, Qiang Tang, and Sébastien Zimmer. An application of the goldwasser-micali cryptosystem to biometric authentication. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 96–106. Springer, 2007.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1996.

[BCPZ08] Julien Bringer, Hervé Chabanne, David Pointcheval, and Sébastien Zimmer. An application of the boneh and shacham group signature scheme to biometric authentication. In Kanta Matsuura and Eiichiro Fujisaki, editors, *Advances in Information and Computer Security, Third International Workshop on Security, IWSEC 2008, Kagawa, Japan, November 25-27, 2008. Proceedings*, volume 5312 of *Lecture Notes in Computer Science*, pages 219–230. Springer, 2008.

[BCR02] Ruud M. Bolle, Jonathan H. Connell, and Nalini K. Ratha. Biometric perils and patches. *Pattern Recognition*, 35(12):2727–2738, 2002.

[BCRA13] Rima Belguechi, Estelle Cherrier, Christophe Rosenberger, and Samy Ait-Aoudia. An integrated framework combining bio-hashed minutiae template and PKCS15 compliant card for a better secure management of fingerprint cancelable templates. *Computers & Security*, 39:325–339, 2013.

[BD15] Mehdi Bentounsi and Cheikh S. Deme. Procédé sécurisé d'analyse externe de données d'exploitation d'une infrastructure de traitement de données. (FR 1561009), November 2015.

[BDF05] Massimo Bartoletti, Pierpaolo Degano, and Gian Luigi Ferrari. Enforcing secure service composition. In *18th IEEE Computer Security Foundations Workshop, (CSFW-18 2005), 20-22 June 2005, Aix-en-Provence, France*, pages 211–223. IEEE Computer Society, 2005.

[BDK+05] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 147–163. Springer, 2005.

[BDSV95] C. Blundo, A. De Santis, and U. Vaccaro. On secret sharing schemes. Technical report, University di Salerno, 1995.

[BEE+10] Jean Bacon, David Evans, David M. Eyers, Matteo Migliavacca, Peter R. Pietzuch, and Brian Shand. Enforcing end-to-end application security in the cloud - (big ideas paper). In Indranil Gupta and Cecilia Mascolo, editors, *Middleware 2010 - ACM/IFIP/USENIX 11th International Middleware Conference, Bangalore, India, November 29 - December 3, 2010. Proceedings*, volume 6452 of *Lecture Notes in Computer Science*, pages 293–312. Springer, 2010.

[BEKM05] Catriel Beeri, Anat Eyal, Simon Kamenkovich, and Tova Milo. Querying business processes with BP-QL. In Klemens Böhm, Christian S. Jensen, Laura M. Haas, Martin L. Kersten, Per-Åke Larson, and Beng Chin Ooi, editors, *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*, pages 1255–1258. ACM, 2005.

[BEKM06] Catriel Beeri, Anat Eyal, Simon Kamenkovich, and Tova Milo. Querying business processes. In Dayal et al. [DWL+06], pages 343–354.

[BEMP07] Catriel Beeri, Anat Eyal, Tova Milo, and Alon Pilberg. Monitoring business processes with queries. In Koch et al. [KGG+07], pages 603–614.

[Ben10] Mehdi Bentounsi. Privacy-aware saas. Master's thesis, Université d'Évry Val d'Essonne, Evry, France, July 2010.

[BEP+14] Jean Bacon, David M. Eyers, Thomas F. J.-M. Pasquier, Jatinder Singh, Ioannis Papagiannis, and Peter Pietzuch. Information flow control for secure cloud computing. *IEEE Transactions on Network and Service Management*, 11(1):76–89, 2014.

[Ber88] Tim Berners-Lee. CERN experience. In *Proceedings of the 3rd ACM SIGOPS European Workshop: Autonomy or Interdependence in Distributed Systems? Cambridge, U.K., September 18-21, 1988*. ACM, 1988.

[BG11] Marina Blanton and Paolo Gasti. Secure and efficient protocols for iris and fingerprint identification. In Vijay Atluri and Claudia Díaz, editors, *Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, volume 6879 of *Lecture Notes in Computer Science*, pages 190–209. Springer, 2011.

[BGG04] Luciano Baresi, Carlo Ghezzi, and Sam Guinea. Smart monitors for composed services. In Marco Aiello, Mikio Aoyama, Francisco Curbera, and Mike P. Papazoglou, editors, *Service-Oriented Computing - ICSOC 2004, Second International Conference, New York, NY, USA, November 15-19, 2004, Proceedings*, pages 193–202. ACM, 2004.

[BGJ+13] Jens-Matthias Bohli, Nils Gruschka, Meiko Jensen, Luigi Lo Iacono, and Ninja Marnau. Security and privacy-enhancing multicloud architectures. *IEEE Trans. Dependable Sec. Comput.*, 10(4):212–224, 2013.

[Bit11] Thomas J. Bittman. The road map from virtualization to cloud computing. *Gartner RAS Core Research Note G00210845*, pages 1–4, March 2011.

[BJR+06] John G. Brainard, Ari Juels, Ronald L. Rivest, Michael Szydlo, and Moti Yung. Fourth-factor authentication: somebody you know. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 168–178. ACM, 2006.

[BM76] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Elsevier Science Publishing Co., Inc., fifth edition, 1976.

[BMH07] Salima Benbernou, Hassina Meziane, and Mohand-Said Hacid. Run-time monitoring for privacy-agreement compliance. In Krämer et al. [KLN07], pages 353–364.

[BMLH07] Salima Benbernou, Hassina Meziane, Yin Hua Li, and Mohand-Said Hacid. A privacy agreement model for web services. In *2007 IEEE International Conference on Services Computing (SCC 2007), 9-13 July 2007, Salt Lake City, Utah, USA*, pages 196–203. IEEE Computer Society, 2007.

[BMM06] Luciano Baresi, Andrea Maurino, and Stefano Modafferi. Towards distributed BPEL orchestrations. *ECEASST*, 3, 2006.

[BMS10] Eran Balan, Tova Milo, and Tal Sterenzy. Bp-ex: a uniform query engine for business process execution traces. In Ioana Manolescu, Stefano Spaccapietra, Jens Teubner, Masaru Kitsuregawa, Alain Léger, Felix Naumann, Anastasia Ailamaki, and Fatma Özcan, editors, *EDBT 2010, 13th International Conference on Extending Database Technology, Lausanne, Switzerland, March 22-26, 2010, Proceedings*, volume 426 of *ACM International Conference Proceeding Series*, pages 713–716. ACM, 2010.

[Bov08] Rosanna Bova. *A task Memory Network Approach for Composite Web Services Selection.* Thèse de doctorat en informatique, May 2008.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*, pages 82–91. ACM, 2004.

[BRA10] Rima Belguechi, Christophe Rosenberger, and Samy Ait-Aoudia. Biohashing for securing minutiae template. In *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, pages 1168–1171. IEEE Computer Society, 2010.

[BS90] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.

[BW01] Shivnath Babu and Jennifer Widom. Continuous queries over data streams. *SIGMOD Record*, 30(3):109–120, 2001.

[BZ10] Cor-Paul Bezemer and Andy Zaidman. Challenges of reengineering into multi-tenant saas applications. Technical Report TUD-SERG-2010-012,

Delft University of Technology - Software Engineering Research Group, 2010.

[Cat82] Hiram Caton. Descartes' anonymous writings: A recapitulation. *Southern Journal of Philosophy*, 20:299–311, 1982.

[CC08] Alfredo Cuzzocrea and Sharma Chakravarthy. Event-based compression and mining of data streams. In Ignac Lovrek, Robert J. Howlett, and Lakhmi C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems, 12th International Conference, KES 2008, Zagreb, Croatia, September 3-5, 2008, Proceedings, Part II*, volume 5178 of *Lecture Notes in Computer Science*, pages 670–681. Springer, 2008.

[CdVFS07] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. k-anonymity. In Ting Yu and Sushil Jajodia, editors, *Secure Data Management in Decentralized Systems*, volume 33 of *Advances in Information Security*, pages 323–353. Springer, 2007.

[CdVFS08] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. k -anonymous data mining: A survey. In Charu C. Aggarwal and Philip S. Yu, editors, *Privacy-Preserving Data Mining - Models and Algorithms*, volume 34 of *Advances in Database Systems*, pages 105–136. Springer, 2008.

[CdVFS09] Valentina Ciriani, Sabrina De Capitani di Vemercati, Sara Foresti, and Pierangela Samarati. *Algorithms and Theory of Computation Handbook*, chapter Theory of Privacy and Anonymity, pages 1—-35. In [AB09], 2nd edition, 2009.

[CEPR07] Raphaël Clifford, Klim Efremenko, Ely Porat, and Amir Rothschild. k -mismatch with don't cares. In Lars Arge, Michael Hoffmann, and Emo Welzl, editors, *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, volume 4698 of *Lecture Notes in Computer Science*, pages 151–162. Springer, 2007.

[CFH06] Barbara Carminati, Elena Ferrari, and Patrick C. K. Hung. Security conscious web service composition. In *2006 IEEE International Conference on Web Services (ICWS 2006), 18-22 September 2006, Chicago, Illinois, USA* [DBL06], pages 489–496.

[CG97] Benny Chor and Niv Gilboa. Computationally private information retrieval (extended abstract). In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium*

on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997, pages 304–313. ACM, 1997.

[CGGA05] Giovanni Di Crescenzo, R. F. Graveman, Renwei Ge, and Gonzalo R. Arce. Approximate message authentication and biometric entity authentication. In Patrick and Yung [PY05], pages 240–254.

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[CJOL04] Tee Connie, Andrew Teoh Beng Jin, Michael Goh Kah Ong, and David Ngo Chek Ling. Palmhashing: a novel approach for dual-factor authentication. *Pattern Anal. Appl.*, 7(3):255–268, 2004.

[CJSS03] Charles D. Cranor, Theodore Johnson, Oliver Spatscheck, and Vladislav Shkapenyuk. Gigascope: A stream database for network applications. In Halevy et al. [HID03], pages 647–651.

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[CLLS10] Graham Cormode, Ninghui Li, Tiancheng Li, and Divesh Srivastava. Minimizing minimality and maximizing utility: Analyzing method-based attacks on anonymized data. *PVLDB*, 3(1):1045–1056, 2010.

[CLN12] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato. Optimization of resource provisioning cost in cloud computing. *IEEE T. Services Computing*, 5(2):164–177, 2012.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In Stern [Ste99], pages 402–414.

[Cor06] James R. Cordy. The TXL source transformation language. *Sci. Comput. Program.*, 61(3):190–210, 2006.

[CST10] Artur Caetano, António Rito Silva, and José M. Tribolet. Identification of services through functional decomposition of business processes. In Witold Abramowicz and Robert Tolksdorf, editors, *Business Information Systems, 13th International Conference, BIS 2010, Berlin, Germany, May 3-5, 2010. Proceedings*, volume 47 of *Lecture Notes in Business Information Processing*, pages 144–157. Springer, 2010.

[CT13] Chris Clifton and Tamir Tassa. On syntactic anonymity and differential privacy. *Transactions on Data Privacy*, 6(2):161–183, 2013.

[Cur00] Wendy Currie. Expanding IS outsourcing services through application service providers. In Hans Robert Hansen, Martin Bichler, and Harald Mahrer, editors, *Proceedings of the 8th European Conference on Information Systems, Trends in Information and Communication Systems for the 21st Century, ECIS 2000, Vienna, Austria, July 3-5, 2000*, pages 961–967, 2000.

[DA01] Wenliang Du and Mikhail J. Atallah. Secure multi-party computation problems and their applications: a review and open problems. In Victor Raskin, Steven J. Greenwald, Brenda Timmerman, and Darrell M. Kienzle, editors, *Proceedings of the New Security Paradigms Workshop 2001, Cloudcroft, New Mexico, USA, September 10-13, 2001*, pages 13–22. ACM, 2001.

[Dau04] John Daugman. How iris recognition works. *IEEE Trans. Circuits Syst. Video Techn.*, 14(1):21–30, 2004.

[DBL06] *2006 IEEE International Conference on Web Services (ICWS 2006), 18-22 September 2006, Chicago, Illinois, USA*. IEEE Computer Society, 2006.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[DDvD+11] Remco M. Dijkman, Marlon Dumas, Boudewijn F. van Dongen, Reina Käärik, and Jan Mendling. Similarity of business process models: Metrics and evaluation. *Inf. Syst.*, 36(2):498–516, 2011.

[Den83] Dorothy E. Denning. A security model for the statistical database problem. In Roy Hammond and John L. McCarthy, editors, *Proceedings of the Second International Workshop on Statistical Database Management, Los Altos, California, USA, September 27-29, 1983*, pages 368–390. Lawrence Berkeley Laboratory, 1983.

[Des09a] Yvo Desmedt. *Algorithms and Theory of Computation Handbook*, chapter Cryptographic Foundations, pages 1—-15. In [AB09], 2nd edition, 2009.

[Des09b] Yvo Desmedt. *Encryption Schemes*, chapter Cryptographic Foundations, pages 1—-30. In [AB09], 2nd edition, 2009.

[DF08] Susan B. Davidson and Juliana Freire. Provenance and scientific workflows: challenges and opportunities. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1345–1350. ACM, 2008.

[DFM98] George I. Davida, Yair Frankel, and Brian J. Matt. On enabling secure applications through off-line biometric identification. In *Security and Privacy - 1998 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 3-6, 1998, Proceedings*, pages 148–157. IEEE Computer Society, 1998.

[DGGR02] Alin Dobra, Minos N. Garofalakis, Johannes Gehrke, and Rajeev Rastogi. Processing complex aggregate queries over data streams. In Michael J. Franklin, Bongki Moon, and Anastassia Ailamaki, editors, *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, June 3-6, 2002*, pages 61–72. ACM, 2002.

[DGK08] Ivan Damgård, Martin Geisler, and Mikkel Krøigaard. Homomorphic encryption and secure comparison. *IJACT*, 1(1):22–31, 2008.

[DGR04] Nelly Delgado, Ann Q. Gates, and Steve Roach. A taxonomy and catalog of runtime software-fault monitoring tools. *IEEE Trans. Software Eng.*, 30(12):859–872, 2004.

[DGRU13] Marlon Dumas, Luciano García-Bañuelos, Marcello La Rosa, and Reina Uba. Fast detection of exact clones in business process model repositories. *Inf. Syst.*, 38(4):619–633, 2013.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[Dif88] Whitfield Diffie. Innovations in internetworking. chapter The First Ten Years of Public-key Cryptography, pages 510–527. Artech House, Inc., Norwood, MA, USA, 1988.

[DKK+12] Yevgeniy Dodis, Bhavana Kanukurthi, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. *IEEE Transactions on Information Theory*, 58(9):6207–6222, 2012.

[DKM+11] Susan B. Davidson, Sanjeev Khanna, Tova Milo, Debmalya Panigrahi, and Sudeepa Roy. Provenance views for module privacy. In Maurizio Lenzerini and Thomas Schwentick, editors, *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, pages 175–186. ACM, 2011.

[DKR+11] Susan B. Davidson, Sanjeev Khanna, Sudeepa Roy, Julia Stoyanovich, Val Tannen, and Yi Chen. On provenance and privacy. In Tova Milo, editor,

Database Theory - ICDT 2011, 14th International Conference, Uppsala, Sweden, March 21-24, 2011, Proceedings, pages 3–10. ACM, 2011.

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer, 2006.

[DL86] George T. Duncan and Diane Lambert. Disclosure-limited data dissemination. *Journal of the American Statistical Association*, 81(393):10–18, 1986.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In Matt Blaze, editor, *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*, pages 303–320. USENIX, 2004.

[Dor43] Robert Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4), December 1943.

[DPW04] Andy Davis, Jay Parikh, and William E. Weihl. Edgecomputing: extending enterprise applications to the edge of the internet. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *Proceedings of the 13th international conference on World Wide Web - Alternate Track Papers & Posters, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 180–187. ACM, 2004.

[DR02] Joan Daemen and Vincent Rijmen. The design of rijndael: Aes - the advanced encryption standard. 2002.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, 2004.

[DS83] Dorothy E. Denning and Jan Schlörer. Inference controls for statistical databases. *IEEE Computer*, 16(7):69–82, 1983.

[Duh03] Joshua Duhl. Rich internet applications. *IDC white paper*, pages 1–33, November 2003.

[DWL+06] Umeshwar Dayal, Kyu-Young Whang, David B. Lomet, Gustavo Alonso, Guy M. Lohman, Martin L. Kersten, Sang Kyun Cha, and Young-Kuk Kim, editors. *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*. ACM, 2006.

[Dwo06] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[Dwo08] Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Ding-Zhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2008.

[EFF98] EFF. *Cracking DES*. O'Reilly Media., July 1998.

[EGH07] David Eppstein, Michael T. Goodrich, and Daniel S. Hirschberg. Improved combinatorial group testing algorithms for real-world problem sizes. *SIAM J. Comput.*, 36(5):1360–1375, 2007.

[EOEA10] Hazem Elmeleegy, Mourad Ouzzani, Ahmed K. Elmagarmid, and Ahmad M. Abusalah. Preserving privacy and fairness in peer-to-peer data integration. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 759–770. ACM, 2010.

[FBXS14] Liyue Fan, Luca Bonomi, Li Xiong, and Vaidy S. Sunderam. Monitoring web browsing behavior with differential privacy. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 177–188. ACM, 2014.

[Fei73] Horst Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, May 1973.

[FIP77] FIPS. Data encryption standard (des). *Federal Information Processing Standards Publication*, 46, January 1977.

[FIP01] FIPS. Advanced encryption standard (aes). *Federal Information Processing Standards Publication*, 197, November 2001.

[FNS75] H. Feistel, W. Notz, and J. Smith. Some cryptographic techniques for machine-to-machine data communications. In *Proceedings of the IEEE*, volume 63, pages 1545–1554. IEEE, November 1975.

[Fos02] Ian Foster. What is the grid? a three point checklist. June 2002.

[Fos05] Ian T. Foster. Service-oriented science: Scaling the application and impact of eresearch. In *First International Conference on e-Science and Grid Technologies (e-Science 2005), 5-8 December 2005, Melbourne, Australia*, page 2. IEEE Computer Society, 2005.

[Fre20] W. F. Freidman. The index of coincidence and its application in cryptography. In *Riverbank publication*, volume 22, Geneva, IL., 1920. Riverbank Labs.

[FSG+14] Diogo A. B. Fernandes, Liliana F. B. Soares, João V. P. Gomes, Mário M. Freire, and Pedro R. M. Inácio. Security issues in cloud environments: a survey. pages 113–170, 2014.

[FTH+12] Benjamin C. M. Fung, Thomas Trojer, Patrick C. K. Hung, Li Xiong, Khalil Al-Hussaeni, and Rachida Dssouli. Service-oriented architecture for high-dimensional private data mashup. *IEEE T. Services Computing*, 5(3):373–386, 2012.

[FW72] Edgar L. Feige and Harold W. Watts. An investigation of the consequences of partial aggregation of micro-economic data. *Econometrica*, 40(2):pp. 343–360, 1972.

[FWY05] Benjamin C. M. Fung, Ke Wang, and Philip S. Yu. Top-down specialization for information and privacy preservation. In Aberer et al. [AFN05], pages 205–216.

[FX14] Liyue Fan and Li Xiong. An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Trans. Knowl. Data Eng.*, 26(9):2094–2106, 2014.

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In Blakley and Chaum [BC85], pages 10–18.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Gas07] Carson Gaspar. Deploying nagios in a large enterprise environment. In Paul Anderson, editor, *Proceedings of the 21th Large Installation System Administration Conference, LISA 2007, Dallas, Texas, USA, November 11-16, 2007*. USENIX, 2007.

[GAT05] Michael T. Goodrich, Mikhail J. Atallah, and Roberto Tamassia. Indexing information for data forensics. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, volume 3531 of *Lecture Notes in Computer Science*, pages 206–221, 2005.

[GCC+04] Daniela Grigori, Fabio Casati, Malú Castellanos, Umeshwar Dayal, Mehmet Sayal, and Ming-Chien Shan. Business process intelligence. *Computers in Industry*, 53(3):321–343, 2004.

[GDG+14] Elio Goettelmann, Karim Dahman, Benjamin Gâteau, Eric Dubois, and Claude Godart. A security risk assessment model for business process deployment in the cloud. In *IEEE International Conference on Services Computing, SCC 2014, Anchorage, AK, USA, June 27 - July 2, 2014*, pages 307–314. IEEE, 2014.

[Gen09a] Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, September 2009.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

[GH11] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2011.

[GJSS09] Lukasz Golab, Theodore Johnson, J. Spencer Seidel, and Vladislav Shkapenyuk. Stream warehousing with datadepot. In Ugur Çetintemel, Stanley B. Zdonik, Donald Kossmann, and Nesime Tatbul, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009*, pages 847–854. ACM, 2009.

[GKS01] Johannes Gehrke, Flip Korn, and Divesh Srivastava. On computing correlated aggregates over continual data streams. In Sharad Mehrotra and Timos K. Sellis, editors, *Proceedings of the 2001 ACM SIGMOD international conference on Management of data, Santa Barbara, CA, USA, May 21-24, 2001*, pages 13–24. ACM, 2001.

[GL03] Alwyn Goh and David Ngo Chek Ling. Computation of cryptographic keys from face biometrics. In Antonio Lioy and Daniele Mazzocchi, editors, *Communications and Multimedia Security - Advanced Techniques for Network and Data Protection, 7th IFIP TC-6 TC-11 International Conference, CMS 2003, Torino, Italy, October 2-3, 2003, Proceedings*, volume 2828 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2003.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[Gol06] Philippe Golle. Revisiting the uniqueness of simple demographics in the US population. In Ari Juels and Marianne Winslett, editors, *Proceedings of the 2006 ACM Workshop on Privacy in the Electronic Society, WPES 2006, Alexandria, VA, USA, October 30, 2006*, pages 77–80. ACM, 2006.

[GR95] Barbara Guttman and Edward A. Roback. An introduction to computer security: The nist handbook. *NIST Special Publication*, 800-12, October 1995.

[GTK+05] Aggarwal G., Feder T., Kenthapadi K., Motwani R., Panigrahy R., Thomas D., and Zhu A. Approximation algorithms for k-anonymity. *Journal of Privacy Technology*, (20051120001), 2005.

[Hen08] Kevin Henry. The theory and applications of homomorphic cryptography. Master's thesis, University of Waterloo, 2008.

[HHLZ10] Zicheng Huang, Jinpeng Huai, Xudong Liu, and Jiangjun Zhu. Business process decomposition based on service relevance mining. In Jimmy Xiangji Huang, Irwin King, Vijay V. Raghavan, and Stefan Rueger, editors, *2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, Toronto, Canada, August 31 - September 3, 2010, Main Conference Proceedings*, pages 573–580. IEEE Computer Society, 2010.

[HID03] Alon Y. Halevy, Zachary G. Ives, and AnHai Doan, editors. *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003*. ACM, 2003.

[HMEK11] Yan Huang, Lior Malka, David Evans, and Jonathan Katz. Efficient privacy-preserving biometric identification. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011*. The Internet Society, 2011.

[HMLZ11] Ines Houidi, Marouen Mechtri, Wajdi Louati, and Djamal Zeghlache. Cloud service delivery across multiple cloud platforms. In Hans-Arno Jacobsen, Yang Wang, and Patrick Hung, editors, *IEEE International Conference on Services Computing, SCC 2011, Washington, DC, USA, 4-9 July, 2011*, pages 741–742. IEEE, 2011.

[HSW07] Ragib Hasan, Radu Sion, and Marianne Winslett. Introducing secure provenance: problems and challenges. In Valerie Henson, editor, *Proceedings of the 2007 ACM Workshop On Storage Security And Survivability, StorageSS 2007, Alexandria, VA, USA, October 29, 2007*, pages 13–18. ACM, 2007.

[HTTA14] Nguyen Quoc Viet Hung, Do Son Thanh, Nguyen Thanh Tam, and Karl Aberer. Privacy-preserving schema reuse. In Sourav S. Bhowmick, Curtis E. Dyreson, Christian S. Jensen, Mong-Li Lee, Agus Muliantara, and Bernhard Thalheim, editors, *Database Systems for Advanced Applications - 19th International Conference, DASFAA 2014, Bali, Indonesia, April 21-24, 2014. Proceedings, Part II*, volume 8422 of *Lecture Notes in Computer Science*, pages 234–250. Springer, 2014.

[ICH10] Dragan Ivanovic, Manuel Carro, and Manuel V. Hermenegildo. Automatic fragment identification in workflows based on sharing analysis. In Maglio et al. [MWYF10], pages 350–364.

[IK99] Yuval Ishai and Eyal Kushilevitz. Improved upper bounds on information-theoretic private information retrieval (extended abstract). In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 79–88. ACM, 1999.

[ITU91] ITU. *Security architecture for Open Systems Interconnection for CCITT applications*, march 1991.

[Iye02] Vijay S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 279–288. ACM, 2002.

[JA05] Roberto J. Bayardo Jr. and Rakesh Agrawal. Data privacy through optimal k-anonymization. In Aberer et al. [AFN05], pages 217–228.

[Jac12] Stéphane Jacob. *Cryptographic protection of databases: conception and cryptanalysis*. PhD thesis, Université Pierre et Marie Curie - Paris VI, Paris, France, March 2012.

[JL05] Andrew Teoh Beng Jin and David Ngo Chek Ling. Cancellable biometerics featuring with tokenised random number. *Pattern Recognition Letters*, 26(10):1454–1460, 2005.

[JLG04a] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, 37(11):2245–2255, 2004.

[JLG04b] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. An integrated dual factor authenticator based on the face data and tokenised random number. In Zhang and Jain [ZJ04], pages 117–123.

[JLKC06] Min Yi Jeong, Chulhan Lee, Jongsun Kim, and Jeung-Yoon Choi. Changeable biometrics for appearance based face recognition. In *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the, Baltimore, September 19, 2006, Proceedings*, pages 1–5, 2006.

[JPHP00] Anil K. Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9(5):846–859, 2000.

[JRP06] Anil K. Jain, Arun Ross, and Sharath Pankanti. Biometrics: a tool for information security. *IEEE Transactions on Information Forensics and Security*, 1(2):125–143, 2006.

[JS02] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *IACR Cryptology ePrint Archive*, 2002:93, 2002.

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In Juzar Motiwalla and Gene Tsudik, editors, *CCS '99, Proceedings of the 6th ACM Conference on Computer and Communications Security, Singapore, November 1-4, 1999.*, pages 28–36. ACM, 1999.

[KAF+10] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman J. J. te Riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768-bit RSA modulus. In

Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 333–350. Springer, 2010.

[Kah96] David Kahn. *The codebreakers : the story of secret writing*. Scribner, New York, 1996.

[KAMR04] Florian Kerschbaum, Mikhail J. Atallah, David M'Raïhi, and John R. Rice. Private fingerprint verification without local storage. In Zhang and Jain [ZJ04], pages 387–394.

[Kas63] F. W. Kasiski. Die geheimschriften und die deciffir-kunst. Berlin, 1863. Mittler & Sohn.

[KC03] Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.

[KGG+07] Christoph Koch, Johannes Gehrke, Minos N. Garofalakis, Divesh Srivastava, Karl Aberer, Anand Deshpande, Daniela Florescu, Chee Yong Chan, Venkatesh Ganti, Carl-Christian Kanne, Wolfgang Klas, and Erich J. Neuhold, editors. *Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, 2007*. ACM, 2007.

[KL06] Rania Khalaf and Frank Leymann. E role-based decomposition of business processes using BPEL. In *2006 IEEE International Conference on Web Services (ICWS 2006), 18-22 September 2006, Chicago, Illinois, USA* [DBL06], pages 770–780.

[Kle05] L. Kleinrock. A vision for the Internet. *ST Journal for Research*, 2(1):4–5, November 2005.

[KLN07] Bernd J. Krämer, Kwei-Jay Lin, and Priya Narasimhan, editors. *Service-Oriented Computing - ICSOC 2007, Fifth International Conference, Vienna, Austria, September 17-20, 2007, Proceedings*, volume 4749 of *Lecture Notes in Computer Science*. Springer, 2007.

[Klö95] Willi Klösgen. Anonymization techniques for knowledge discovery in databases. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, August 20-21, 1995*, pages 186–191. AAAI Press, 1995.

[KNON10] Hiroaki Kikuchi, Kei Nagai, Wakaha Ogata, and Masakatsu Nishigaki. Privacy-preserving similarity evaluation and application to remote biometrics authentication. *Soft Comput.*, 14(5):529–536, 2010.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 364–373. IEEE Computer Society, 1997.

[KPP+13] Kyriakos Kritikos, Barbara Pernici, Pierluigi Plebani, Cinzia Cappiello, Marco Comuzzi, Salima Benbernou, Ivona Brandic, Attila Kertész, Michael Parkin, and Manuel Carro. A survey on service quality description. *ACM Comput. Surv.*, 46(1):1, 2013.

[KSvdV+05] Tom A. M. Kevenaar, Geert Jan Schrijen, Michiel van der Veen, Anton H. M. Akkermans, and Fei Zuo. Face recognition with renewable and privacy preserving binary templates. In *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID 2005), 16-18 October 2005, Buffalo, NY, USA*, pages 21–26. IEEE Computer Society, 2005.

[KW95] J. Kim and William E. Winkler. Masking microdata files. In *ASA Proceedings of the Section on Survey Research Methods*, pages 114–119, 1995.

[LAC+14] Thi My Hanh Le, Luis Alfredo Alfaro, Hyung Rim Choi, Min Je Cho, and Chae-Soo Kim. A study on bpaas with TCO model. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing, BDCloud 2014, Sydney, Australia, December 3-5, 2014*, pages 249–256. IEEE, 2014.

[Lan01] Marc Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In Gregory D. Abowd, Barry Brumitt, and Steven A. Shafer, editors, *Ubicomp 2001: Ubiquitous Computing, Third International Conference Atlanta, Georgia, USA, September 30 - October 2, 2001, Proceedings*, volume 2201 of *Lecture Notes in Computer Science*, pages 273–291. Springer, 2001.

[LDR05] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In Özcan [Özc05], pages 49–60.

[Lew00] Robert Edward Lewand. *Cryptological Mathematics*. Mathematical Association of America, Washington, DC, USA, 1st edition, 2000.

[LHPB09] Giusy Di Lorenzo, Hakim Hacid, Hye-Young Paik, and Boualem Benatallah. Data integration in mashups. *SIGMOD Record*, 38(1):59–66, 2009.

[LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In Rada Chirkova, Asuman Dogac, M. Tamer Özsu, and Timos K. Sellis, editors, *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, pages 106–115. IEEE, 2007.

[LMG+10] Feifei Li, Mirella M. Moro, Shahram Ghandeharizadeh, Jayant R. Haritsa, Gerhard Weikum, Michael J. Carey, Fabio Casati, Edward Y. Chang, Ioana Manolescu, Sharad Mehrotra, Umeshwar Dayal, and Vassilis J. Tsotras, editors. *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*. IEEE, 2010.

[LTG+14] Lydia Leong, Douglas Toombs, Bob Gill, Gregor Petri, and Tiny Haynes. Magic quadrant for cloud infrastructure as a service. *Gartner ID:G00261698*, May 2014.

[LV01] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *J. Cryptology*, 14(4):255–293, 2001.

[LWZ10] Wen Ming Liu, Lingyu Wang, and Lei Zhang. k-jump strategy for preserving privacy in micro-data disclosure. In Luc Segoufin, editor, *Database Theory - ICDT 2010, 13th International Conference, Lausanne, Switzerland, March 23-25, 2010, Proceedings*, ACM International Conference Proceeding Series, pages 104–115. ACM, 2010.

[Mal11] Sayanta Mallick. Virtualization based cloud capacity prediction. In Waleed W. Smari and John P. McIntire, editors, *2011 International Conference on High Performance Computing & Simulation, HPCS 2012, Istanbul, Turkey, July 4-8, 2011*, pages 849–852. IEEE, 2011.

[Mat93] Mitsuru Matsui. Linear cryptoanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[MB10] Hassina Meziane and Salima Benbernou. A dynamic privacy model for web services. *Computer Standards & Interfaces*, 32(5-6):288–304, 2010.

[MBZ+10] Hassina Meziane, Salima Benbernou, Aouda K. Zerdali, Mohand-Said Hacid, and Mike P. Papazoglou. A view-based monitoring for privacy-aware web services. In Li et al. [LMG+10], pages 1129–1132.

[McN10] Robert McNeill. The evolution of business process as a service (bpaas). *Special Research Reprint Courtesy of Progress Software*, October 2010.

[MD08] Tova Milo and Daniel Deutch. Querying and monitoring distributed business processes. *PVLDB*, 1(2):1512–1515, 2008.

[MD11] Michele Mazzucco and Marlon Dumas. Reserved or on-demand instances? A revenue maximization model for cloud providers. In Ling Liu and Manish Parashar, editors, *IEEE International Conference on Cloud Computing, CLOUD 2011, Washington, DC, USA, 4-9 July, 2011*, pages 428–435. IEEE, 2011.

[MDKL11] Michele Mancioppi, Olha Danylevych, Dimka Karastoyanova, and Frank Leymann. Towards classification criteria for process fragmentation techniques. In Florian Daniel, Kamel Barkaoui, and Schahram Dustdar, editors, *Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I*, volume 99 of *Lecture Notes in Business Information Processing*, pages 1–12. Springer, 2011.

[MFWH09] Noman Mohammed, Benjamin C. M. Fung, Ke Wang, and Patrick C. K. Hung. Privacy-preserving data mashup. In Martin L. Kersten, Boris Novikov, Jens Teubner, Vladimir Polutin, and Stefan Manegold, editors, *EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, March 24-26, 2009, Proceedings*, volume 360 of *ACM International Conference Proceeding Series*, pages 228–239. ACM, 2009.

[MGKV06] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*, page 24. IEEE Computer Society, 2006.

[MHD12] Sayanta Mallick, Gaétan Hains, and Cheikh Sadibou Deme. A resource prediction model for virtualization servers. In Waleed W. Smari and Vesna Zeljkovic, editors, *2012 International Conference on High Performance*

Computing & Simulation, HPCS 2012, Madrid, Spain, July 2-6, 2012, pages 667–671. IEEE, 2012.

[MMJP09] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer-Verlag London, 2nd edition, 2009.

[Mol07] Richard A. Mollin. *An Introduction to Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 2nd edition, 2007.

[MRK+03] Milan Milenkovic, Scott H. Robinson, Rob C. Knauerhase, David Barkai, Sharad Garg, Vijay Tewari, Todd A. Anderson, and Mic Bowman. Toward internet distributed computing. *IEEE Computer*, 36(5):38–46, 2003.

[MS07] Khaled Mahbub and George Spanoudakis. Monitoring *WS-Agreement* s: An event calculus-based approach. In Luciano Baresi and Elisabetta Di Nitto, editors, *Test and Analysis of Web Services*, pages 265–306. Springer, 2007.

[MSES97] Farach M., Kannan S., Knill E., and Muthukrishnan S. Group testing problems with sequences in experimental molecular biology. In *Compression and Complexity of Sequences*, pages 357–367, 1997.

[MVO96] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.

[MW04] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *PODS*, pages 223–228, 2004.

[MWYF10] Paul P. Maglio, Mathias Weske, Jian Yang, and Marcelo Fantinato, editors. *Service-Oriented Computing - 8th International Conference, ICSOC 2010, San Francisco, CA, USA, December 7-10, 2010. Proceedings*, volume 6470 of *Lecture Notes in Computer Science*, 2010.

[MZZW09] Qian Ma, Nianjun Zhou, Yanfeng Zhu, and Hao Wang. Evaluating service identification with design metrics on business process decomposition. In *2009 IEEE International Conference on Services Computing (SCC 2009), 21-25 September 2009, Bangalore, India*, pages 160–167. IEEE Computer Society, 2009.

[NBCT06] Hamid R. Motahari Nezhad, Boualem Benatallah, Fabio Casati, and Farouk Toumani. Web services interoperability specifications. *IEEE Computer*, 39(5):24–32, 2006.

[NC11] Ahmet Erhan Nergiz and Chris Clifton. Query processing in private data outsourcing using anonymization. In Yingjiu Li, editor, *Data and Applications Security and Privacy XXV - 25th Annual IFIP WG 11.3 Conference, DBSec 2011, Richmond, VA, USA, July 11-13, 2011. Proceedings*, volume 6818 of *Lecture Notes in Computer Science*, pages 138–153. Springer, 2011.

[NCM13] Ahmet Erhan Nergiz, Chris Clifton, and Qutaibah M. Malluhi. Updating outsourced anatomized private databases. In Giovanna Guerrini and Norman W. Paton, editors, *Joint 2013 EDBT/ICDT Conferences, EDBT '13 Proceedings, Genoa, Italy, March 18-22, 2013*, pages 179–190. ACM, 2013.

[NLPvdH12] Dinh Khoa Nguyen, Francesco Lelli, Mike P. Papazoglou, and Willem-Jan van den Heuvel. Blueprinting approach in support of cloud computing. *Future Internet*, 4(1):322–346, 2012.

[NMS+15] Vivek R. Narasayya, Ishai Menache, Mohit Singh, Feng Li, Manoj Syamala, and Surajit Chaudhuri. Sharing buffer pool memory in multi-tenant relational database-as-a-service. *PVLDB*, 8(7):726–737, 2015.

[NPI+15] Yefim V. Natis, Massimo Pezzini, Kimihiko Iijima, Anne Thomas, and Rob Dunie. Magic quadrant for enterprise application platform as a service, worldwide. *Gartner ID:G00271188*, March 2015.

[OJW03] Chris Olston, Jing Jiang, and Jennifer Widom. Adaptive filters for continuous queries over distributed data streams. In Halevy et al. [HID03], pages 563–574.

[OPJM10] Margarita Osadchy, Benny Pinkas, Ayman Jarrous, and Boaz Moskovich. Scifi - A system for secure face identification. In *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*, pages 239–254. IEEE Computer Society, 2010.

[oST12] National Institute of Standards and Technology. *Recommendation for Key Management*. NIST Special Publication 800-57 Part 1 Rev. 3., July 2012.

[Özc05] Fatma Özcan, editor. *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*. ACM, 2005.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Stern [Ste99], pages 223–238.

[Pap12] Michael P. Papazoglou. Cloud blueprints for integrating and managing cloud federations. In Maritta Heisel, editor, *Software Service and Application Engineering - Essays Dedicated to Bernd Krämer on the Occasion of His 65th Birthday*, volume 7365 of *Lecture Notes in Computer Science*, pages 102–119. Springer, 2012.

[Plo09] Guillaume Plouin. Chapter 1 - contexte de l'émergence du cloud computing. In SQLI, editor, *Cloud Computing et SaaS*, pages 1 – 23. Dunod, Paris, 2009.

[Poi02] D. Pointcheval. *Le chiffrement asymétrique et la sécurité prouvée*. Habilitation à diriger des recherches, Université Paris 7, Paris, France, June 2002.

[PP12] Vinícius M. Pacheco and Ricardo Staciarini Puttini. Defining and implementing connection anonymity for saas web services. In Rong Chang, editor, *2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, USA, June 24-29, 2012*, pages 479–486. IEEE, 2012.

[PvdH11] Michael P. Papazoglou and Willem-Jan van den Heuvel. Blueprinting the cloud. *IEEE Internet Computing*, 15(6):74–79, 2011.

[PY05] Andrew S. Patrick and Moti Yung, editors. *Financial Cryptography and Data Security, 9th International Conference, FC 2005, Roseau, The Commonwealth of Dominica, February 28 - March 3, 2005, Revised Papers*, volume 3570 of *Lecture Notes in Computer Science*. Springer, 2005.

[Rab05] Michael O. Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptology ePrint Archive*, 2005:187, 2005.

[RAD] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In Richard A. DeMillo, David P. Dobkin, Anita K. Jones, and Richard J. Lipton, editors, *Foundations of Secure Computation*, pages 165–179. Academic Press.

[RCB01] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[RDtHM11] Marcello La Rosa, Marlon Dumas, Arthur H. M. ter Hofstede, and Jan Mendling. Configurable multi-perspective business process models. *Inf. Syst.*, 36(2):313–340, 2011.

[RFG10] Mohsen Rouached, Walid Fdhila, and Claude Godart. Web services compositions modelling and choreographies analysis. *Int. J. Web Service Res.*, 7(2):87–110, 2010.

[RKM10] Stefan Ried, Holger Kisker, and Pascal Matzke. The evolution of cloud computing markets. *Forrester Research, Inc.*, pages 1–16, July 2010.

[Rob07] Chris Roberts. Biometric attack vectors and defences. *Computers & Security*, 26(1):14–25, 2007.

[RR99] Michael K. Reiter and Aviel D. Rubin. Anonymous web transactions with crowds. *Commun. ACM*, 42(2):32–38, 1999.

[RRvdA+11] Marcello La Rosa, Hajo A. Reijers, Wil M. P. van der Aalst, Remco M. Dijkman, Jan Mendling, Marlon Dumas, and Luciano García-Bañuelos. APROMORE: an advanced process model repository. *Expert Syst. Appl.*, 38(6):7029–7040, 2011.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[RVC+12] Luis Rodero-Merino, Luis Miguel Vaquero, Eddy Caron, Adrian Muresan, and Frédéric Desprez. Building safe paas clouds: A survey on security in multitenant software platforms. *Computers & Security*, 31(1):96–108, 2012.

[RW04] Jeanne W. Ross and George Westerman. Preparing for utility computing: The role of IT architecture and relationship management. *IBM Systems Journal*, 43(1):5–19, 2004.

[Sac77] Luigi Sacco. *Manual of cryptography*, volume 14 of *A Cryptographic series.* 1977. Translation of: Manuale di crittografia. Bibliography: p. viii.

[Sam01] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[SF03] Howard Smith and Peter Fingar. *Business Process Management : The Third Wave.* Meghan-Kiffer Press, 2003.

[SGR97] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous connections and onion routing. In *1997 IEEE Symposium on Security and Privacy, May 4-7, 1997, Oakland, CA, USA*, pages 44–54. IEEE Computer Society, 1997.

[Sha49] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656—-715, October 1949.

[Shi00] R. Shirey. Internet security glossary. *Request For Comments - RFC*, 2828, May 2000.

[SKK+11] David Schumm, Dimka Karastoyanova, Oliver Kopp, Frank Leymann, Mirko Sonntag, and Steve Strauch. Process fragment libraries for easier and faster development of process-based applications. *Journal of Systems Integration*, 2(1):39–55, 2011.

[SSO12] Siamak Fayyaz Shahandashti, Reihaneh Safavi-Naini, and Philip Ogunbona. Private fingerprint matching. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *Information Security and Privacy - 17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9-11, 2012. Proceedings*, volume 7372 of *Lecture Notes in Computer Science*, pages 426–433. Springer, 2012.

[SSW09] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Efficient privacy-preserving face recognition. In Donghoon Lee and Seokhie Hong, editors, *Information, Security and Cryptology - ICISC 2009, 12th International Conference, Seoul, Korea, December 2-4, 2009, Revised Selected Papers*, volume 5984 of *Lecture Notes in Computer Science*, pages 229–244. Springer, 2009.

[SSY14] Yutian Sun, Jianwen Su, and Jian Yang. Separating execution and data management: A key to business-process-as-a-service (bpaas). In Shazia Wasim Sadiq, Pnina Soffer, and Hagen Völzer, editors, *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, volume 8659 of *Lecture Notes in Computer Science*, pages 374–382. Springer, 2014.

[ST06] Berry Schoenmakers and Pim Tuyls. Efficient binary conversion for paillier encrypted values. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 522–537. Springer, 2006.

[ST11] Reihaneh Safavi-Naini and Dongvu Tonien. Fuzzy universal hashing and approximate authentication. *Discrete Math., Alg. and Appl.*, 3(4):587–608, 2011.

[Sta10] William Stallings. *Cryptography and Network Security Principles and Practice*. Pearson Education, Inc., 1 Lake Street, Upper Saddle River, NY 07458, 5th edition, 2010.

[Ste99] Jacques Stern, editor. *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*. Springer, 1999.

[Sti95] Douglas R. Stinson. *Cryptography: Theory and Practice*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1995.

[Sto02] Michael Stonebraker. Too much middleware. *SIGMOD Record*, 31(1):97–106, 2002.

[Swe02] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[SYTB13] Wei She, I-Ling Yen, Bhavani M. Thuraisingham, and Elisa Bertino. Security-aware service composition with fine-grained information flow control. *IEEE T. Services Computing*, 6(3):330–343, 2013.

[SZG+14] Felipe Díaz Sánchez, Sawsan Al Zahr, Maurice Gagnaire, Jean-Pierre Laisné, and Iain James Marshall. Compatibleone: Bringing cloud as a commodity. In *2014 IEEE International Conference on Cloud Engineering, Boston, MA, USA, March 11-14, 2014*, pages 397–402. IEEE, 2014.

[TAK+05] Pim Tuyls, Anton H. M. Akkermans, Tom A. M. Kevenaar, Geert Jan Schrijen, Asker M. Bazen, and Raymond N. J. Veldhuis. Practical biometric authentication with template protection. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *Audio- and Video-Based Biometric Person Authentication, 5th International Conference, AVBPA 2005, Hilton Rye Town, NY, USA, July 20-22, 2005, Proceedings*, volume 3546 of *Lecture Notes in Computer Science*, pages 436–446. Springer, 2005.

[TBCP08] Qiang Tang, Julien Bringer, Hervé Chabanne, and David Pointcheval. A formal study of the privacy concerns in biometric-based remote authentication schemes. In Liqun Chen, Yi Mu, and Willy Susilo, editors, *Information Security Practice and Experience, 4th International Conference, ISPEC 2008, Sydney, Australia, April 21-23, 2008, Proceedings*, volume 4991 of *Lecture Notes in Computer Science*, pages 56–70. Springer, 2008.

[TBS15] Wilson Abel Alberto Torres, Nandita Bhattacharjee, and Bala Srinivasan. Privacy-preserving biometrics authentication systems using fully homomorphic encryption. *Int. J. Pervasive Computing and Communications*, 11(2):151–168, 2015.

[TFH09] Thomas Trojer, Benjamin C. M. Fung, and Patrick C. K. Hung. Service-oriented architecture for privacy-preserving data mashup. In *IEEE International Conference on Web Services, ICWS 2009, Los Angeles, CA, USA, 6-10 July 2009*, pages 767–774. IEEE, 2009.

[TG04] Pim Tuyls and Jasper Goseling. Capacity and examples of template-protecting biometric authentication systems. In Davide Maltoni and Anil K. Jain, editors, *Biometric Authentication, ECCV 2004 International Workshop, BioAW 2004, Prague, Czech Republic, May 15, 2004, Proceedings*, volume 3087 of *Lecture Notes in Computer Science*, pages 158–170. Springer, 2004.

[THNvdH11] Yehia Taher, Rafiqul Haque, Dinh Khoa Nguyen, and Willem-Jan van den Heuvel. Designing and delivering public services on the cloud. In Frank Leymann, Ivan Ivanov, Marten van Sinderen, and Boris Shishkov, editors, *CLOSER 2011 - Proceedings of the 1st International Conference on Cloud Computing and Services Science, Noordwijkerhout, Netherlands, 7-9 May, 2011*, pages 471–476. SciTePress, 2011.

[THP+11] Yehia Taher, Rafiqul Haque, Michael Parkin, Willem-Jan van den Heuvel, Ita Richardson, and Eoin Whelan. A multi-layer approach for customizing business services. In Christian Huemer and Thomas Setzer, editors, *E-Commerce and Web Technologies - 12th International Conference, EC-Web 2011, Toulouse, France, August 30 - September 1, 2011. Proceedings*, volume 85 of *Lecture Notes in Business Information Processing*, pages 64–76. Springer, 2011.

[THvdHF13] Yehia Taher, Rafiqul Haque, Willem-Jan van den Heuvel, and Béatrice Finance. α bpaas - A customizable bpaas on the cloud. In Frédéric Desprez, Donald Ferguson, Ethan Hadar, Frank Leymann, Matthias Jarke, and Markus Helfert, editors, *CLOSER 2013 - Proceedings of the 3rd International Conference on Cloud Computing and Services Science, Aachen, Germany, 8-10 May, 2013*, pages 290–296. SciTePress, 2013.

[TRF+07] Ioan Toma, Dumitru Roman, Dieter Fensel, Brahmananda Sapkota, and Juan Miguel Gómez. A multi-criteria service ranking approach based on

non-functional properties rules evaluation. In Krämer et al. [KLN07], pages 435–441.

[TV06] Traian Marius Truta and Bindu Vinay. Privacy protection: p-sensitive k-anonymity property. In Roger S. Barga and Xiaofang Zhou, editors, *Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDE 2006, 3-7 April 2006, Atlanta, GA, USA*, page 94. IEEE Computer Society, 2006.

[TYI88] Katsumi Tanaka, Masatoshi Yoshikawa, and Kozo Ishihara. Schema virtualization in object-oriented databases. In *Proceedings of the Fourth International Conference on Data Engineering, February 1-5, 1988, Los Angeles, California, USA*, pages 23–30. IEEE Computer Society, 1988.

[Vaa02] Risto Vaarandi. Platform independent tool for local event correlation. *Acta Cybern.*, 15(4):705–723, 2002.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.

[Vou08] Mladen A. Vouk. Cloud computing - issues, research and implementations. *CIT*, 16(4):235–246, 2008.

[WABS09] Christof Weinhardt, Arun Anandasivam, Benjamin Blau, and Jochen Stößer. Business models in the service world. pages 28–33, 2009.

[WFWP07] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, and Jian Pei. Minimality attack in privacy preserving data publishing. In Koch et al. [KGG+07], pages 543–554.

[Win02] William E. Winkler. Using simulated annealing for k-anonymity. Technical Report 7, U.S. Census Bureau., 2002.

[Win04] William E. Winkler. Masking and re-identification methods for public-use microdata: Overview and research problems. In Josep Domingo-Ferrer and Vicenç Torra, editors, *Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004, Barcelona, Spain, June 9-11, 2004. Proceedings*, volume 3050 of *Lecture Notes in Computer Science*, pages 231–246. Springer, 2004.

[Win11] Vic (J.R.) Winkler. Chapter 1 - introduction to cloud computing and security. In Vic (J.R.) Winkler, editor, *Securing the Cloud*, pages 1 – 27. Syngress, Boston, 2011.

[WLFW06] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, and Ke Wang. (alpha, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 754–759. ACM, 2006.

[WMK06] Ke Wei, Muthusrinivasan Muthuprasanna, and Suraj Kothari. Preventing SQL injection attacks in stored procedures. In *17th Australian Software Engineering Conference (ASWEC 2006), 18-21 April 2006, Sydney, Australia*, pages 191–198. IEEE Computer Society, 2006.

[WSZ04] Huanmei Wu, Betty Salzberg, and Donghui Zhang. Online event-driven subsequence matching over financial data streams. In Gerhard Weikum, Arnd Christian König, and Stefan Deßloch, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, pages 23–34. ACM, 2004.

[XT06] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In Dayal et al. [DWL+06], pages 139–150.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167. IEEE Computer Society, 1986.

[YB12] Qi Yu and Athman Bouguettaya. Multi-attribute optimization in service selection. *World Wide Web*, 15(1):1–31, 2012.

[YBS08] Lamia Youseff, Maria Butrico, and Dilma Da Silva. Towards a unified ontology of cloud computing. In *in Proc. of the Grid Computing Environments Workshop, GCE08*, 2008.

[YSK+13] Masaya Yasuda, Takeshi Shimoyama, Jun Kogure, Kazuhiro Yokoyama, and Takeshi Koshiba. Packed homomorphic encryption based on ideal lattices and its application to biometrics. In Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar R. Weippl, Lida Xu, Alfredo Cuzzocrea,

Christian Kittl, Dimitris E. Simos, Edgar R. Weippl, and Lida Xu, editors, *Security Engineering and Intelligence Informatics - CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Regensburg, Germany, September 2-6, 2013. Proceedings*, volume 8128 of *Lecture Notes in Computer Science*, pages 55–74. Springer, 2013.

[YWC02] William E. Yancey, William E. Winkler, and Robert H. Creecy. Disclosure risk assessment in perturbative microdata protection. In Josep Domingo-Ferrer, editor, *Inference Control in Statistical Databases, From Theory to Practice*, volume 2316 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2002.

[YZB11] Zhen Ye, Xiaofang Zhou, and Athman Bouguettaya. Genetic algorithm based qos-aware service compositions in cloud computing. In Jeffrey Xu Yu, Myoung-Ho Kim, and Rainer Unland, editors, *Database Systems for Advanced Applications - 16th International Conference, DASFAA 2011, Hong Kong, China, April 22-25, 2011, Proceedings, Part II*, volume 6588 of *Lecture Notes in Computer Science*, pages 321–334. Springer, 2011.

[ZBC10] Mohamed Anis Zemni, Salima Benbernou, and Manuel Carro. A soft constraint-based approach to qos-aware service selection. In Maglio et al. [MWYF10], pages 596–602.

[ZJ04] David Zhang and Anil K. Jain, editors. *Biometric Authentication, First International Conference, ICBA 2004, Hong Kong, China, July 15-17, 2004, Proceedings*, volume 3072 of *Lecture Notes in Computer Science*. Springer, 2004.

[ZK00] Du D. Z. and Hwang F. K. *Combinatorial Group Testing and its Applications*. World Scientific, 2nd edition, 2000.

[ZKOS05] Rui Zhang, Nick Koudas, Beng Chin Ooi, and Divesh Srivastava. Multiple aggregations over data streams. In Özcan [Özc05], pages 299–310.

[ZLOX10] Haiquan (Chuck) Zhao, Ashwin Lall, Mitsunori Ogihara, and Jun (Jim) Xu. Global iceberg detection over distributed data streams. In Li et al. [LMG+10], pages 557–568.

[ZZYB13] Huiyuan Zheng, Weiliang Zhao, Jian Yang, and Athman Bouguettaya. Qos analysis for web service compositions with complex structures. *IEEE T. Services Computing*, 6(3):373–386, 2013.