Evaluation de la confiance dans un processus d'authentification

Julien Hatin

Normandie Université

THESE

Pour obtenir le diplôme de doctorat

Spécialité informatique

Préparée au sein de l'ENSICAEN et de l'UNICAEN

Evaluation de la confiance dans un processus d'authentification

Présentée et soutenue par
Julien HATIN

Thèse soutenue publiquement le 24 Novembre 2017 devant le jury composé de		
Mme. Maryline LAURENT	Professeur des Universités, Télécom Sud Paris	Rapporteur
M. Jean-François LALANDE	Maître de conférences HDR, INSA Centre Val de Loire	Rapporteur
M. Pascal URIEN	Professeur des Universités, Télécom Paris Tech	Examinateur
M. Jean-Jacques SCHWARTZMANN	Ingénieur Recherche et Développement, Orange Labs	Examinateur
Mme. Estelle CHERRIER	Maître de conférence, ENSICAEN	Examinateur
M. Christophe ROSENBERGER	Professeur des Universités, ENSICAEN	Directeur de thèse

Thèse dirigée par Christophe ROSENBERGER, laboratoire GREYC

UNIVERSITÉ CAEN NORMANDIE

ENSICAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

ed 181

GREYC

Résumé

Dans notre quotidien, le smartphone est devenu un outil indispensable pour effectuer nos tâches courantes. Accéder à des services en ligne depuis son téléphone mobile est devenu une action commune. Afin de s'authentifier à ces services parfois sensibles, la seule protection est généralement l'usage d'un mot de passe. Ces mots de passe pour être robustes doivent être de plus en plus longs. Ceci représente, sur les téléphones mobiles, une contrainte plus forte que pour les ordinateurs de bureau puisque les claviers tactiles disposent de moins de touches. D'autres méthodes d'authentification ont vu le jour sur téléphones mobiles comme la reconnaissance faciale sur les appareils android ou bien l'empreinte digitale qui gagne le marché des smartphones et même le domaine bancaire avec Apple Pay.

Afin de simplifier l'authentification, la biométrie prend une part de plus en plus importante dans l'usage des téléphones mobiles. Au delà des capteurs dédiés à l'acquisition de données biométriques, il est aussi possible d'utiliser l'environnement du téléphone mobile pour authentifier les utilisateurs. Si les méthodes d'authentification tendent à se transformer pour devenir de plus en plus transparentes, cela amène deux questions :

— Comment utiliser ces nouvelles techniques d'authentification dans les processus actuels d'authentification ?
— Quels impacts ces nouvelles méthodes peuvent avoir sur la vie privée des utilisateurs ?

L'objectif de cette thèse est de proposer des méthodes d'authentification transparentes qui soient respectueuses de la vie privée des utilisateurs tout en permettant leur intégration dans les systèmes actuels d'authentification.

Dans le manuscrit de thèse, nous abordons ces deux questions en analysant tout d'abord les travaux existants sur la collecte des données permettant l'authentification

sur téléphone mobile. Puis, une fois les données collectées, nous verrons les processus permettant la mise en place d'une authentification respectueuse de la vie privée. Enfin, nous évaluons concrètement ces méthodes d'authentification par la réalisation de prototypes à l'échelle industrielle.

Summary

In our daily life, the smartphone became an unavoidable tool to perform our common tasks. Accessing to online services from its mobile phone is an usual action. In order to authenticate to those services, that might be sensitive, the one and only protection is usually a password. Those passwords must be longer and longer to stay robust. This is a bigger constraint on mobile phones than on desktop computers. Other authentication solutions are dedicated to smartphones, like facial recognition on android and now Apple smartphones or the fingerprint that conquier new phones.

To ease the authentication process, biometrics is more and more often used on mobile phones. In addition to the dedicated biometric sensors, it is also possible to use the phone environement to authenticate users. However, if authentication methods are becoming more and more transparent, it brings two questions :

— How to integrate those new methods within the actual authentication framework ?

— What is the impact of those new methods on users' privacy ?

The main goal of the phD is to offers privacy compliant transparent authentication methods while integrating them in current authentication systems.

In this document, we evaluates those two questions by first analyzing existing works on the data collection for transparent authentication on mobile phones. Then, once the data are collected, we will see wich process can enable the privacy protection. To conclude, we will evaluates concretly those solutions by building industrial prototypes.

Table des matières

Chapitre 1

Introduction

1.1 Contexte

Cette thèse est financée par le biais d'une convention Cifre avec l'entreprise Orange. Elle porte sur l'établissement de méthodes d'authentification transparente respectueuses de la vie privée.

Cette thèse a été effectuée à Caen au sein du laboratoire Groupe de Recherche en informatique, image, automatique et instrumentation (GREYC) dans l'équipe Monétique & Biométrie, et de la société Orange Labs à Caen depuis octobre 2015 sous la direction de Christophe Rosenberger et l'encadrement de Estelle Cherrier et de Jean-Jacques Schwarztmann. Les transactions électroniques sécurisées ainsi que la biométrie font parties des domaines de recherche de ces équipes. C'est pour cette raison qu'une collaboration a été réalisée autour de cette thèse.

Les smartphones offrent de nouvelles possibilités de calcul et de collecte de données. En parallèle, les données et les services sont de plus en plus connectés et le paradigme d'une authentification pour se connecter à une session devient obsolète dès lors que la connexion est permanente. Cependant, malgré cette évolution, la règle est toujours l'utilisation du mot de passe. Dans ce cadre, la thème de la thèse s'articule autour de l'utilisation du téléphone mobile comme nouvel objet d'authentification dans les environnements connectés.

1.2 Motivations

Alors que nous avons basculé avec les technologies 3G et 4G dans un environnement continuellement connecté, nous utilisons encore des mécanismes d'accès qui sont adaptés à des transactions uniques. Il suffit désormais de dérober un téléphone mobile

déverrouillé pour avoir accès aux comptes emails, bancaires ou encore aux données personnelles d'un individu.

La biométrie est de plus en plus présente sur nos téléphones mobiles par le biais de nouveaux capteurs et de nouvelles méthodes d'authentification. On peut citer l'apparition de l'authentification par empreinte digitale, mais aussi par reconnaissance faciale. Si cela permet de sécuriser d'avantage les interactions, ce ne permet pas d'agir en continu.

Il est donc nécessaire de redéfinir les mécanismes d'authentification afin d'intégrer ce nouveau paradigme de connexion permanente. Il est nécessaire d'authentifier en continu l'utilisation pour un meilleur confort d'usage et une meilleure sécurité.

Une solution simple déjà évoquée dans la littérature [25] est d'inverser la problématique actuelle liée au mot de passe. Plutôt que de demander à l'utilisateur de mémoriser une donnée, ce qui est difficile pour un humain et simple pour un ordinateur, pourquoi ne pas demander à une machine d'apprendre à reconnaître en continu son utilisateur.

Cependant, effectuer cette transformation suppose de respecter un certain nombre de règles. Tout d'abord, ces authentifications ne peuvent pas être intrusives dans la sphère privée de l'utilisateur. Elles doivent rester le plus transparent possible pour l'utilisateur, sans quoi l'usage des services deviendraient un véritable calvaire. De plus, une authentification en continu ne doit pas se transformer en espionnage de l'utilisateur, au risque de briser la confiance de l'utilisateur dans les services qui lui sont proposés.

Ceci nous amène donc à évaluer la confiance que nous pouvons avoir dans l'utilisateur d'un service tout en simplifiant l'usage des services et sans porter atteinte à sa vie privée. Cela ne veut pas dire que nous devons cesser de prendre en compte les anciennes méthodes d'authentification. Celles-ci trouvent encore un usage que ce soit pour sceller des transactions particulières ou encore dans le contexte machine to machine.

Néanmoins, la multiplicité et la variété des données présentes sur nos téléphones mobiles permet aujourd'hui d'établir des profils d'utilisateurs intéressants qui permettent d'envisager des solutions plus intuitives pour l'utilisateur.

1.3 Organisation du manuscrit

Le manuscrit est organisé en 5 chapitres ayant pour objectif de présenter le contexte de cette thèse puis les contributions dans les domaines des techniques d'authentification transparente et de leurs intégrations, pour finir par une conclusion

sur les travaux effectués ainsi qu'une présentation de différentes perspectives sur ce sujet.

Le manuscrit est articulé de la façon suivante :

— **Chapitre 2 :** Positionne le problème en définissant les notions les plus importantes pour mieux appréhender les contributions de la thèse.

— **Chapitre 3 :** Fait un état des lieux des différentes données collectables sur un téléphone mobile et de la manière de les utiliser pour développer de nouvelles techniques d'authentification. Les différents classifieurs utilisés dans la littérature y sont décrits ainsi que les méthodes d'évaluation.

— **Chapitre 4 :** Compare différentes méthodes d'authentification respectueuses de la vie privée élaborées dans le cadre de cette thèse. Ces évaluations ont été menées sur des bases de données publiques et privées afin de répondre au mieux aux attentes de cette thèse.

— **Chapitre 5 :** Expose les différentes solutions qui existent dans les normes et l'état de l'art pour évaluer une authentification. A l'issue de cette état de l'art, une solution basée sur la théorie de Dempster Shaffer est proposée pour réaliser un framework d'évaluation de la confiance.

— **Chapitre 6 :** Décrit une solution afin d'intégrer les propositions effectuées dans le cadre de cette thèse dans un environnement industriel. Un cadre logiciel permettant aux équipes de R&D et aux équipes de développement est définie. Cette solution est en cours d'implémentation dans la société Orange.

Enfin, nous concluons ce manuscrit et donnons quelques perspectives.

Chapitre 2

Positionnement de la problématique

Ce chapitre décrit les fondamentaux de l'authentification. Une définition est proposée afin de poser les bases de ce manuscrit. Les différentes solutions d'authentification transparente de l'état de l'art sont également abordées avant de décrire les objectifs de cette thèse.

Sommaire

2.1 Introduction

Aujourd'hui, nous sommes entourés de services connectés et sommes contraints de retenir un mot de passe pour chacun d'eux. Facebook Connect [1] et Google Sign In [2] sont deux solutions qui permettent de se connecter à plusieurs autres services avec un identifiant unique. Conscients de la faiblesse des schémas à mot de passe, ces acteurs proposent maintenant des solutions d'authentification bi-facteurs. Dans ces procédures d'authentification, l'utilisateur doit entrer à la fois un mot de passe à usage unique (One Time Password) généré par une application présente sur son

1. https://developers.facebook.com/products/login
2. https://developers.google.com/identity/protocols/OpenIDConnect

smartphone en plus de son mot de passe [52]. Ces solutions ne sont que des parades aux problématiques actuelles de l'authentification et il est nécessaire d'aller au-delà de simples ajouts en repensant le processus d'authentification.

L'objectif de ce chapitre est donc de présenter le rôle de l'authentification et de formaliser une définition que nous allons suivre dans ce manuscrit. Dans un second temps, nous présentons les différents acteurs à même de jouer un rôle dans l'univers de "L'authentification 2.0". Enfin, nous présentons les exigences du système d'authentification en terme de vie privée, sécurité et usage. Ceci nous permet de définir les objectifs de cette thèse.

2.2 Authentification

Nous nous focalisons dans cette section sur l'authentification d'un individu.

2.2.1 Définitions

Bob se réclame d'une identité afin de pouvoir régler ses achats en utilisant un chèque. Pour cela, il doit prouver que l'identifiant, constitué du couple $\langle prénom, nom \rangle$, lui est propre. Cet identifiant a été enregistré auprès de l'état au moment de sa déclaration de naissance. Afin de s'authentifier, il va alors présenter une preuve : sa carte d'identité.

Afin que Bob puisse prouver son identité, il lui faut :
— Un identifiant : $\langle prénom, nom \rangle$
— Une preuve de son identité : sa carte d'identité

L'ANSSI [99] définit l'authentification de la manière suivante :

Definition 1. *L'authentification a pour but de vérifier l'identité dont une entité (personne ou machine) se réclame. L'authentification est toujours précédée ou combinée avec une identification qui permet à cette entité de se faire reconnaître du système par un élément dont on l'a doté : un identifiant. En résumé, s'identifier c'est communiquer un identifiant présumé, s'authentifier c'est apporter la preuve que l'entité s'est vue attribuer cet identifiant.*

Deux notions importantes sont abordées dans cette définition. La première est la nécessité d'être enregistré auprès du système d'authentification et la seconde est

l'apport d'une preuve que l'identifiant appartient bien à l'utilisateur.

Dans l'exemple d'introduction, Bob présentait un identifiant et un justificatif d'identité. Tous les deux doivent être connus par le système qui va l'authentifier.

Cet enregistrement est appelé l'enrôlement. L'enrôlement consiste à associer à un identifiant, un ou plusieurs justificatifs. Ces justificatifs sont appelés facteurs d'authentification. Dans le cas de Bob, cet enrôlement a eu lieu avec l'émission de son certificat de naissance et un renouvellement périodique de son facteur d'authentification est réalisé avec le renouvellement de sa carte d'identité.

L'ISO [61] définit l'authentification comme :

Definition 2. *La provision d'assurance dans l'identité proclamée par une entité*[3].

Dans cette seconde définition, la norme introduit la notion "d'assurance dans l'identité". Cette assurance implique une relation de confiance tacite entre entités. On retrouve cette notion d'assurance au sein de l'identité dans la définition des spécifications SAML [88]. SAML ajoute que ce niveau de confiance qui permet l'authentification, doit soit être spécifié par un accord préalable entre les parties, soit compris.

Or, la confiance est une notion subjective [78]. Ceci met en évidence l'importance de la relation d'humain à humain dans la notion d'authentification.

Jøsang [64] propose huit classes d'authentification. Ces classes d'authentification sont établies en fonction des entités en jeu qui peuvent être : des personnes physiques, morales ou des infrastructures. Ces huit classes d'authentification sont représentées sur la figure 6.1

A partir de ces éléments, on peut définir l'authentification comme :

Definition 3. *Le processus consistant à fournir des éléments en vue d'établir un certain niveau de confiance dans l'identité d'une entité. Une entité peut être une personne physique, morale ou une infrastructure ayant un rôle dans ce processus.*

Nous utilisons cette définition dans la suite de ce rapport.

3. Traduit de l'anglais : "Provision of assurance in the claimed identity of an entity"

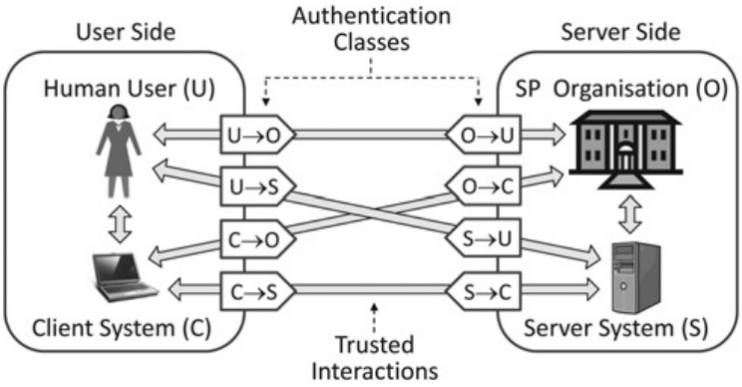

FIGURE 2.1 – Relations dans un processus d'authentification [64]

2.2.2 Facteurs d'authentification

Un facteur d'authentification est une information qui permet d'authentifier l'utilisateur. Cameron *et al.* [18] parlent de revendication primordiale [4] et la définissent comme :

Definition 4. *Une preuve – basée sur un/des secret(s) et/ou des données biométriques – qu'un seul et unique sujet est capable de présenter à un fournisseur spécifique de revendications dans le but d'être reconnu et d'obtenir un jeu de revendications substantielles* [5].

Dans cette définition, Cameron *et al.* [18] parlent de revendication primordiale, ceci est à prendre au sens littéral, c'est à dire, la première revendication réalisée, celle qui permet l'authentification. On peut donc définir cette revendication, dont on a fait la preuve, comme une preuve de l'appartenance du facteur à l'utilisateur.

On classe généralement les facteurs d'authentification en quatre catégories [61] :
— Facteur mémoriel (Ce que l'utilisateur sait)
— Facteur matériel (Ce que l'utilisateur possède)
— Facteur corporel (Ce que l'utilisateur montre)
— Facteur réactionnel (Ce que l'utilisateur fait)

D'autres propositions ont été réalisées pour classifier les facteurs d'authentifica-

4. Traduit de l'anglais "Primordial Claim"
5. Traduit de l'anglais "A proof – based on secret(s) and/or biometrics – that only a single subject is able to present to a specific claims provider for the purpose of being recognised and obtaining a set of substantive claims"

tion. Ainsi, Arias *et al.* [3], utilisent le terme "Ce que votre dispositif sait"[6]. Il s'agit d'inscrire un secret dans un dispositif intelligent, typiquement le smartphone. Ceci revient à utiliser un facteur matériel comme facteur d'authentification.

Halunen et Evesti [54] proposent d'utiliser le contexte comme un facteur d'authentification. D'autres utilisent des éléments pouvant être introduits dans le contexte. Brainard *et al.* [14] proposent d'ajouter "Quelqu'un que vous connaissez" et Bardram *et al.* [5] proposent la localisation comme facteur d'authentification.

Une revendication prouvée est la preuve que l'utilisateur détient un facteur d'authentification qui peut être mémoriel, matériel, corporel ou réactionnel. La nature de cette preuve, ainsi que le moyen par lequel elle a été obtenue, permettra de déterminer la confiance à lui accorder. C'est donc la difficulté à usurper une revendication prouvée qui détermine le niveau de confiance que l'on peut avoir dans l'identité clamée par un utilisateur pour une revendication prouvée donnée.

Afin d'obtenir une preuve sur l'identité et donc la véracité du facteur d'authentification, un protocole d'authentification est généralement mis en place. Ce dernier peut être composé de plusieurs facteurs d'authentification. Il s'agit alors d'une authentification multi-facteurs. La plus connue des authentifications multi-facteurs a lieu lors d'un retrait d'argent avec une carte bancaire. Afin d'obtenir de l'argent, l'utilisateur s'authentifie auprès de la banque émettrice au travers du distributeur automatique de billets avec deux facteurs :

— Sa carte bancaire (Facteur matériel)
— Son code secret (Facteur mémoriel)

Le fait de combiner plusieurs facteurs d'authentification permet d'augmenter le nombre de revendications prouvées, et ainsi de renforcer la conviction que l'identité clamée et donc les informations qui constituent cette identité sont celles de l'utilisateur qui se trouve au distributeur automatique de billets.

Le terme "credentials" est souvent utilisé afin de décrire la preuve de la détention du facteur d'authentification. Ce terme francisé décrit aussi bien les revendications faites sur l'utilisateur que les revendications prouvées. De plus, ce terme est propice à la confusion entre facteurs d'authentification et revendication prouvée. Cette confusion vient du fait que les "credentials" sont traditionnellement associés au couple $\langle nom\ d'utilisateur, mot\ de\ passe \rangle$. On remarque que, si dans ce schéma d'authen-

6. Traduit de l'anglais "What your device know ?"

tification, le mot de passe est transmis en clair, alors la revendication et le facteur d'authentification sont confondus.

2.2.3 Canaux d'authentification

Dans certains cas, les facteurs d'authentification sont demandés sur des canaux de communications séparés [79]. Afin de réaliser un achat en ligne, il est souvent demandé de saisir un code qui a été envoyé par SMS au travers du réseau GSM, en plus des informations de la carte bancaire (solution dite "Out of the band"). Ceci permet donc d'avoir deux canaux de communications distincts et d'éviter de corrompre l'ensemble des facteurs d'authentification si un des deux canaux est lui-même corrompu. Ceci n'empêche pas les attaques si les deux canaux sont corrompus [7].

Lorsqu'on parle de sécurité informatique, l'analogie avec une chaîne cadenassée est souvent réalisée. Ainsi, on dit de la chaîne de sécurité, qu'elle est aussi faible que le plus faible de ses maillons. Le fait d'utiliser des canaux de communications différents pour transporter les facteurs d'authentification revient à mettre deux maillons en parallèles au lieu d'un seul.

Selon la revendication prouvée qui est envoyée afin de s'authentifier, les préjudices d'une interception de la preuve peuvent varier grandement. Dans le cas d'une image de l'empreinte digitale de l'utilisateur (preuve de la détention d'un facteur corporel), si l'image est interceptée, elle peut alors être rejouée. Une fois révoquée, l'image de l'empreinte digitale d'un individu, contrairement à un mot de passe, ne peut pas être remplacée. La qualité du canal de transmission de l'information détermine la confiance que l'on peut en avoir. Si maintenant, seules des coordonnées de points caractéristiques de la main avaient été chiffrées en incluant des données aléatoires pour empêcher le rejeu (biométrie révocable), alors les conséquences sur le facteur d'authentification auraient été moins graves.

Il existe par ailleurs des canaux de communications qui ne sont pas maîtrisés. Si un facteur d'authentification est partagé entre plusieurs services d'authentification et qu'un de ces services d'authentification n'applique pas une politique de préservation des données suffisamment fiable, alors le mécanisme d'authentification des autres services est lui aussi affaibli.

7. http://securityintelligence.com/man-in-the-mobile-attacks-single-out-android

On remarque donc qu'il y a une imbrication entre le canal de communication et le facteur. Ils sont tous les deux liés dans le mécanisme d'authentification. Ainsi, la combinaison de plusieurs facteurs peut permettre d'améliorer la confiance dans un mécanisme d'authentification si et seulement si le processus de transmission des revendications prouvées est assuré convenablement.

2.2.4 Authentification forte et authentification mutuelle

L'authentification forte est communément décrite comme une authentification multi-facteurs.

Elle suppose que les facteurs soient robustes dans le sens où ils offrent un niveau d'assurance suffisant. Ensuite, elle implique aussi que ces facteurs soient correctement acheminés jusqu'à l'entité qui a besoin d'une preuve sur l'identité. En d'autres termes, il est nécessaire que les facteurs soient transportés par des canaux distincts et sécurisés. Enfin, la notion de force est une notion subjective. Pour quels facteurs une authentification est-elle forte ? Doit-on avoir forcément plusieurs facteurs ou bien est ce qu'un seul facteur très robuste ne serait pas suffisant ?

Divers niveaux d'assurance existent, dans la littérature, pour évaluer la confiance dans l'identité à partir du nombre de facteurs d'authentification et d'une architecture donnée [61, 51, 4, 115, 1]. Ces niveaux d'assurance permettent de conforter la définition de l'authentification forte comme une authentification multi-facteurs. En effet, ils apportent des pré-requis à satisfaire quant à la nature des facteurs et à leur gestion.

En repartant de la définition de l'authentification précédemment énoncée, nous pourrions définir l'authentification forte comme :

Definition 5. *Le processus consistant à fournir des éléments en vue d'établir un niveau de confiance* ***élevé*** *dans l'identité d'une entité.*

Là encore, le terme élevé est subjectif, il est sous-entendu que le niveau de confiance adopté est suffisant pour un processus sensible. Le but de l'authentification forte est de s'assurer que le niveau de confiance est suffisant pour la ou les transactions à venir.

Le terme authentification forte sous-entend aussi que les deux parties se sont authentifiées l'une envers l'autre. On parle alors d'authentification mutuelle. C'est le cas, par exemple, lorsqu'un utilisateur accède à ses relevés de comptes sur le site

internet de sa banque. Le serveur hébergeant le site internet de la banque présente un certificat TLS [34] qui certifie l'identité de la banque en tant que personne morale (voir figure 6.1).

2.2.5 Authentification unique

L'authentification unique est une authentification qui permet à un consommateur de services de réutiliser le résultat de ladite authentification pour accéder à plusieurs services. L'authentification unique simplifie le parcours utilisateur. Si chaque service demande de manière indépendante à l'utilisateur de s'authentifier alors l'utilisateur devra :

— S'authentifier à de multiples reprises
— Mémoriser de nombreux mots de passe
— Transporter de nombreux facteurs matériels différents (Carte à puce, Tokens)
— S'enrôler pour chaque service

Jøsang et Pope [65] proposent deux classes distinctes pour réaliser un système d'authentification unique :

— Modèle fédéré de gestion d'identité (voir 2.2)
— Modèle centralisé de gestion d'identité (voir 2.3)

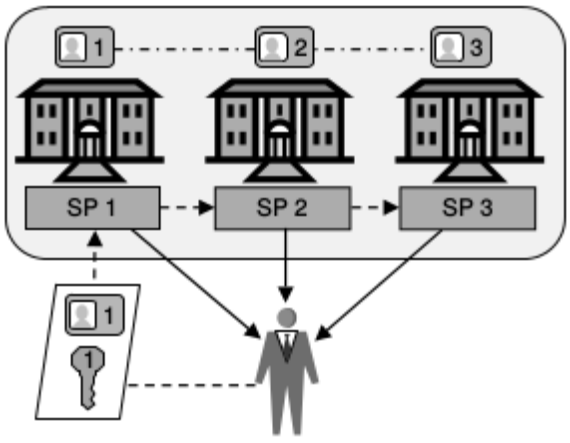

FIGURE 2.2 – Modèle fédéré de gestion d'identité [64]

Le modèle fédéré de gestion d'identité peut être décrit comme un ensemble d'accords, de normes, de standards et de technologies qui permettent à un ensemble de fournisseurs de services de reconnaître les identifiants et les droits d'un utilisateur

depuis les autres fournisseurs de services au sein d'un même domaine fédéré [65] (voir figure 2.2).

Les spécifications SAML emploient la définition suivante pour le terme fédérer :

Definition 6. *Lier ou attacher deux ou plusieurs entités ensembles*[8]

Le résultat est un domaine unique d'identité incluant chaque domaine d'identité des fournisseurs de services fédérés. Parmi les standards proposant cette solution, il est possible de citer : OASIS SAML, OpenID Connect/Oauth.

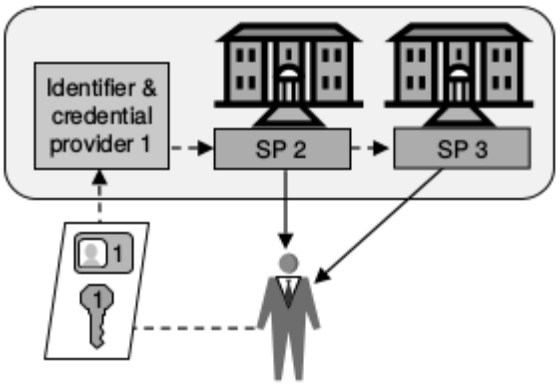

FIGURE 2.3 – Modèle centralisé de gestion d'identité [64]

Dans le modèle centralisé de gestion d'identité, il existe un fournisseur d'identifiants et de facteurs d'authentification qui est utilisé par tous les fournisseurs de services. On retrouve généralement ce scénario dans les systèmes d'authentification unique d'entreprise. Dans le cas d'une authentification à partir d'un serveur Kerberos, le serveur Kerberos agit comme le fournisseur d'identifiants et de facteurs d'authentification.

Le modèle le plus adapté pour accéder à des services en ligne est incontestablement le modèle fédéré de gestion d'identité. Il est difficile d'imaginer un modèle centralisé de gestion d'identité dans un espace aussi vaste que l'Internet. Cependant, il est aussi impossible d'imaginer un modèle de gestion d'identité fédérant l'ensemble des fournisseurs de services de la planète. Il est donc important de maximiser la compatibilité entre les différents domaines d'identités. Les deux acteurs majeurs à proposer ce type de solutions sont Google et Facebook [118].

8. Traduit de l'anglais "To link or bind two or more entities together"

2.2.6 Authentification continue et ponctuelle

L'authentification permet d'établir un certain niveau de confiance dans l'identité d'une entité. Dans le cadre d'une authentification classique, ce niveau de confiance est établi de manière ponctuelle en début de session. Si nous représentons le niveau de confiance en fonction du temps, il est impossible de garantir qu'après l'authentification, l'utilisateur soit toujours l'utilisateur légitime une fois la preuve d'authentification fournie. Ce cas arrive lorsqu'une session est ouverte par le détenteur pour un tiers. Par exemple, il est possible pour Bob de déverrouiller son téléphone pour que Alice puisse appeler avec. Nous avons ainsi représenté l'apport de confiance fournit par l'authentification ponctuelle comme un trait sur la figure 2.4. A l'opposé de la vérification ponctuelle, l'authentification continue vérifie l'identité de l'utilisateur sans interruption. Il n'est ainsi pas possible, avec une authentification continue, pour Bob d'ouvrir une session à Alice.

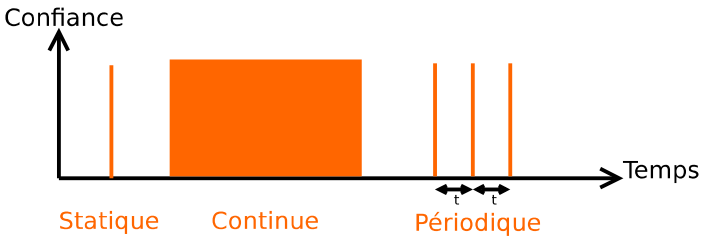

FIGURE 2.4 – Apport de confiance dans l'authentification ponctuelle et continue

Sur téléphone mobile, les premiers travaux d'authentification continue remonte à 2004 avec les travaux de Clarke [23]. Ces premiers travaux utilisaient des techniques d'authentification biométriques basées sur la dynamique de frappe au clavier afin d'authentifier de manière continue l'utilisateur. Depuis, les smartphones ont envahi le marché et offrent une large variété d'informations sur le comportement de l'utilisateur.

Les techniques d'authentification continue sont généralement utilisées en complément d'une authentification explicite. Dans l'étude comparative de Khan *et al.* [70], l'authentification continue repose sur le principe d'authentifier une personne en fonction de ses agissements. Il s'agit donc de l'authentifier en utilisant de la biométrie comportementale. Certains intègrent des éléments de contexte afin d'en déduire des comportements. [17, 107, 76, 97, 102, 72].

On peut citer :
— Probabilité d'être à un lieu donné à une heure donnée [57, 98]
— Appel vers un numéro connu/inconnu [119]

— Accès à des sites web connus / inconnus [107]

Une proposition atypique est le projet ePet [15, 112, 111]. Afin de faciliter l'interaction entre le système d'authentification continue et l'utilisateur, les auteurs proposent de réaliser une application sur téléphone mobile qui imiterait un animal de compagnie.
Il faudrait alors interagir avec cet animal virtuel afin qu'il apprenne à connaître son utilisateur. Il agirait ensuite comme un chien de garde du téléphone mobile qui empêcherait les intrus d'entrer. Dans ce projet, la continuité de l'authentification peut être achevée par des méthodes d'authentification ponctuelle telles que la reconnaissance faciale.

Afin de s'assurer de la présence de l'utilisateur au cours du temps il est souvent demander à l'utilisateur de ressaisir un facteur d'authentification au bout d'un certains temps. Ceci s'apparente davantage à une succession d'authentifications ponctuelles plutôt qu'à une réelle authentification continue. Nous préférons employer le terme d'authentification périodique pour définir ce type d'authentification.

Les authentifications ponctuelles sont aujourd'hui la norme. Ce système est plus simple à implémenter et s'attend à recevoir un attribut prouvé (comme un mot de passe, une donnée biométrique, ...) qui est prédéterminé [110]. Cependant, les authentifications réalisées de manière ponctuelles ne permettent pas de s'assurer que l'utilisateur légitime est toujours le même après l'ouverture de la session.

2.3 Exigences du système d'authentification

Bonneau *et al.* [10] évaluent les systèmes actuels d'authentification autour de trois axes majeurs, qui sont l'Usabilité, le Déploiement et la Sécurité de la solution d'authentification. Un axe récurrent est celui de la protection de la vie privée [3, 64, 2]. Nous avons choisi de nous confronter aux exigences suivantes :

— Vie privée
— Usabilité
— Sécurité

Ces trois besoins sont antagonistes et il est nécessaire de choisir un triangle adapté à l'authentification et au niveau de confiance requis (voir figure 2.5).

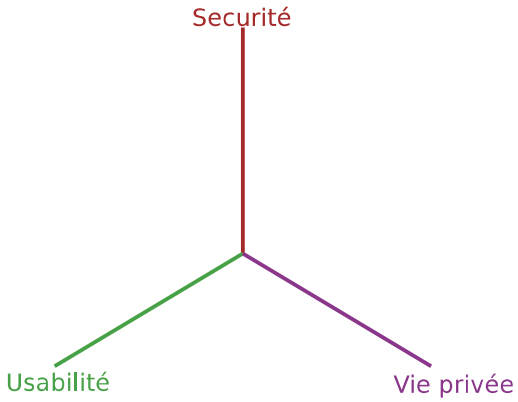

FIGURE 2.5 – Représentation graphique des exigences d'un système d'authentification

2.3.1 Vie privée

La protection de la vie privée des utilisateurs est une notion qu'il est désormais nécessaire de prendre en compte dès la conception d'un système (privacy by design). La CNIL dans son guide "Gestion des risques vie privée" propose des méthodes et des recommandations pour minimiser les risques liés aux données à caractère personnel. Au niveau européen, le Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 [9], régit le traitement des données à caractère personnel et la protection de la vie privé dans le secteur des télécommunications. Aux États Unis, la protection des données personnelles est traitée par le "Privacy Act" [116]. Le concept de "privacy by design" est activement défendu sur Internet [10].

Lors de la gestion de ces risques, les méthodes à mettre en place (chiffrement, hachage, anonymisation) sont dépendants des données traitées. L'objectif est d'éviter la divulgation d'informations sensibles.

Afin de définir les données à protéger, il est nécessaire d'évoquer la notion d'éléments d'intérêts.

Definition 1. *Un élément d'intérêt est un élément qui ne doit pas être divulgué à un attaquant potentiel pour le propriétaire de cet élément.*

On distingue quelques types de données dont il faut prêter attention particulièrement au moment de l'anonymisation.

9. Accessible en ligne : https://www.cnil.fr/fr/reglement-europeen-protection-donnees
10. http://www.viepriveeintegree.ca/

— **Les identificateurs :** ce sont des attributs qui identifient directement un utilisateur : un nom, un numéro de téléphone. Ceux-ci doivent êtres retirées de la base.
— **Les attributs sensibles :** Ils doivent êtres impérativement protégées d'un attaquant. Si celui-ci parvient à relier un attribut sensible à l'identité réelle d'une personne, cela représenterait une atteinte à la vie privée de cette personne. Un historique d'achats sur Internet ou des requêtes sur un moteur de recherche sont des exemples de données sensibles.
— **Les quasi-identificateurs :** ce sont des attributs qui n'identifient pas une personne directement. A première vue, un quasi-identificateur ne paraît pas être sensible, cependant, en utilisant des informations externes, ils peuvent êtres utilisées pour identifier un utilisateur. Un quasi-identificateur peut-être une date de naissance ou une adresse IP par exemple.

Le rapport technique de Pfitzmann et Hansen [93] établit une terminologie pour décrire la protection de la vie privée par minimisation des données. Les termes décrits sont :

Anonymat : Un sujet ne peut pas être identifié au sein d'un groupe par un attaquant

Impossibilité d'établir un lien : Un ou plusieurs articles d'intérêts (sujet, messages, actions, ...) ne peuvent pas être reliés du point de vue d'un attaquant

Indétectabilité : L'indétectabilité d'un article d'intérêt signifie qu'il est impossible pour un attaquant de savoir si cet article existe

Inobservabilité : Un article d'intérêt est inobservable s'il est à la fois anonyme et indétectable.

Les contraintes de protection de la vie privée sont définies vis à vis d'un attaquant, c'est à dire d'une personne visant à utiliser des données à caractère personnel de manière illégitime. Ces contraintes visent plus généralement les systèmes de gestions d'identités. Cependant, lors de la conception d'un système d'authentification, le stockage et l'utilisation des attributs prouvés peuvent amener des faiblesses et des attaques sur la vie privée des utilisateurs peuvent être possibles. Par exemple, le lien entre une photographie utilisée pour la reconnaissance faciale et la couleur de peau peut être fait. Ceci nuit à la protection des données à caractère personnel puisqu'une information à caractère racial peut être déduit de la méthode de stockage ou de transmission de l'attribut prouvé.

L'objectif de la protection de la vie privée et des données à caractère personnel

est de garantir les libertés fondamentales de l'individu. Les données et en particulier les données biométriques sont des données qui lui appartiennent et sur lesquels il doit garder les pleins pouvoirs.

2.3.2 Sécurité

L'authentification est la première barrière contre l'usurpation d'identité. Une fois une identité numérique usurpée, il est possible d'accéder à des données à caractère personnel ou professionnel. Il est aussi possible de réaliser des actions au nom de l'utilisateur.

La sécurité est une question globale qui inclut aussi la gestion des ressources non informatiques, nous ne nous intéressons ici qu'aux aspects sécuritaires de l'authentification.

Menezes *et al.* [80] définissent les attaques suivantes comme pouvant être faites sur un protocole d'authentification en plus de la recherche exhaustive :

Attaque par imitation : Une tromperie pour laquelle une entité prétend être une autre.

Attaque par rejeu : Une imitation ou une autre tromperie impliquant l'utilisation d'une information provenant d'une exécution précédente unique du protocole auprès d'un même ou d'un autre vérificateur

Attaque par entrelacement : Une imitation ou une autre tromperie impliquant une combinaison sélective d'informations, d'une ou plusieurs, précédentes ou simultanées, exécutions du protocole (sessions parallèles), incluant la possibilité qu'une ou plusieurs exécutions du protocole proviennent de l'adversaire lui même.

Attaque par réflexion : Une attaque par entrelacement impliquant l'envoi d'informations depuis un protocole en cours d'exécution en retour vers l'auteur de l'information.

Attaque par retard forcé : Un retard forcé se produit lorsqu'un attaquant intercepte un message (qui contient un numéro de séquence par exemple) et le retransmet plus tard dans le temps. Il ne s'agit pas d'une attaque par rejeu.

Attaque par texte choisi : Une attaque dans laquelle l'attaquant choisit le challenge afin d'obtenir de l'information sur le secret partagé.

En addition des attaques réalisables sur le protocole, il est aussi nécessaire de vérifier l'implémentation. Ainsi Kainda *et al.* [66], proposent d'évaluer un système en

supposant que l'utilisateur suivra toujours le chemin de plus faible résistance. C'est à dire le chemin qui l'emmènera le rapidement possible à l'accomplissement de sa tâche. Ceci peut engendrer des failles dans le système d'authentification. Un exemple criant de ce type de comportement est que lorsque les mots de passe deviennent de plus en plus compliqués, les utilisateurs ont tendance à les écrire à proximité plutôt que d'essayer de les mémoriser.

2.3.3 Usabilité

Selon l'ISO 9241-11, l'utilisabilité ou (usabilité) est définie par le "degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié".

Nielsen [86] explique que l'usabilité est une sous-partie de l'acceptabilité des systèmes. L'acceptabilité d'un système est la composante qui décrit si un système est suffisamment bon pour répondre à tous les besoins d'un utilisateur et des autres parties prenantes. On peut encore décomposer l'acceptabilité en deux parties qui sont :

— L'acceptabilité sociale
— L'acceptabilité pratique

Dans les composantes de l'acceptabilité pratique, on trouve l'utilisation qui peut encore être séparée entre l'utilité et l'usabilité. La figure 2.6 reprend le positionnement de l'usabilité vis à vis des autres composantes de l'acceptabilité d'un système.

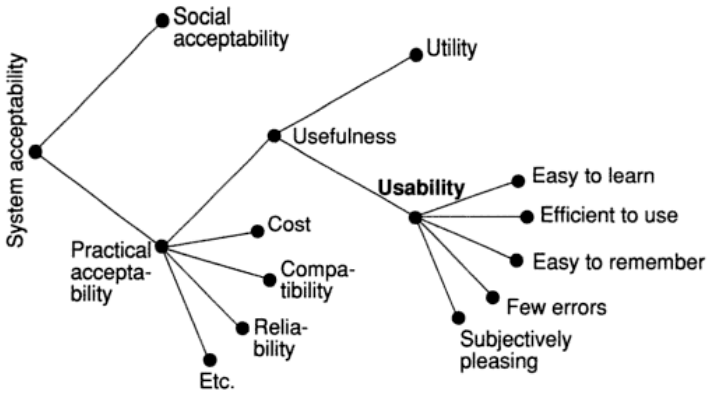

FIGURE 2.6 – Modèle des attributs de l'acceptabilité d'un système [86]

L'usabilité est traditionnellement décrite avec les cinq composantes suivantes [86] :

Facilité d'apprentissage : Le système doit être facile à apprendre de manière à ce que l'utilisateur puisse rapidement réaliser des tâches avec le système.

Efficacité d'utilisation : Le système doit être efficace à utiliser, après la phase d'apprentissage pour garantir un haut niveau de productivité.

Facilité de mémorisation : Le système doit être facile à mémoriser de telle façon qu'un utilisateur occasionnel puisse revenir au système après une période de non utilisation sans avoir à tout réapprendre.

Erreurs : Le système doit avoir un faible taux d'erreur, et il doit être facile de corriger les erreurs éventuelles.

Satisfaction : Le système doit être plaisant à utiliser et les utilisateurs doivent être subjectivement satisfaits en l'utilisant.

2.4 L'authentification transparente sur mobile

L'authentification transparente [24] est non intrusive. Ces méthodes d'authentification sont aussi appelées implicites [107] ou encore authentification active [53]. Il est possible d'utiliser à la fois des moyens d'authentification classiques comme des cartes à puce sans contact que des techniques biométriques.

2.4.1 Modalités intrusives

Afin d'assurer une authentification transparente en continue, il est souvent nécessaire de faire appel à des facteurs d'authentification qui ne sont pas transparents. Les sessions commencent généralement par la saisie d'un code d'identification personnel (PIN) ou d'un mot de passe. Cette étape est nécessaire et dans les systèmes commerciaux proposant de l'authentification en continue, il est nécessaire de pouvoir avoir recours à ces facteurs plus conventionnels afin de pouvoir palier à des défaillances du système ou à l'indisponibilité de facteur d'authentification transparent.

2.4.2 Biométrie classique

Les capteurs d'empreintes digitales se sont largement démocratisés depuis la sortie du système Touch ID d'Apple (figure 2.7 sur l'Iphone 5). On les retrouve désormais sur une large gamme de smartphone haut de gamme. Ces capteurs permettent d'authentifier l'utilisateur de manière transparente puisque l'empreinte est capturée au moment d'effectuer un geste naturel lors de l'utilisation du téléphone. Ces capteurs d'empreintes digitales sont d'ailleurs accessibles directement à l'application via l'API

de dévellopement sur les appareils Android [11]. Les empreintes digitales, sont dans ces systèmes, stockées dans des environnements sécurisés au seins même du téléphone.

FIGURE 2.7 – Le système TouchID d'Apple

Toujours afin de faciliter le déverrouillage, la reconnaissance faciale est disponible sur les téléphones Android depuis la version 4.0 (Ice Cream Sandwich) [12]. Partant du constat que cette authentification par reconnaissance faciale pourrait se faire de manière implicite lorsque l'utilisateur regarde son écran, les auteurs de [28] proposent de combiner les relevés effectués par les capteurs gyroscopiques, accéléromètriques et magnétométriques du téléphone afin de réorienter l'image capturée de manière implicite par le téléphone mobile de l'utilisateur. Ceci permet de détecter un imposteur en moins de 2 minutes dans 89% des cas.

2.4.3 Dynamique d'interaction tactile

Depuis l'apparition de l'Iphone en 2007, les smartphones sont maintenant équipés d'écrans tactiles. L'utilisateur interagit au travers de ces écrans tactiles avec les applications de son appareil. Il est alors possible de récupérer sans avoir à demander d'action spécifique à l'utilisateur les mouvements effectués sur les applications du téléphone mobile.

Ce type d'authentification présente l'avantage d'être disponible sur l'ensemble des smartphones. De plus, rares sont les usages du téléphone mobile qui n'utilisent pas l'écran tactile. Ceci permet d'avoir une modalité qui est utilisée en quasi continuité.

Cependant, l'accès à ces données en continue nécessite d'avoir accès au coeur du système. Ceci implique de disposer des droits d'administration. Ces restrictions

11. https://developer.android.com/reference/android/hardware/fingerprint/FingerprintManager.html
12. https ://developer.android.com/about/versions/android-4.0-highlights.html

ont étaient mises en place avant d'éviter que des programmes malveillants puissent observer les interactions de l'utilisateur afin de récolter des informations sensibles comme la saisie d'un code personnel (PIN) par exemple. La figure 2.8 présente des exemples de capture.

FIGURE 2.8 – Exemple d'interactions tactiles [43]

Les auteurs de [43], utilisent des techniques d'apprentissage supervisé (Support vector machine et K plus proche voisins) et obtiennent un EER ((taux d'égal erreur correspondant au compromis de faux rejets et fausses acceptations) de 3% en utilisant une fenêtre de 13 interactions tactiles. Cette étude est réalisée sur une base de 100 sujets. Cependant, les auteurs de [70] obtiennent des résultats plus faibles sur une bases d'interactions tactiles différentes. Ceci montre la forte variabilité des systèmes d'authentification comportementales lorsqu'ils sont exécutés dans un environnement non contrôlé.

2.4.4 Reconnaissance de la démarche

D'après les auteurs de [70], une solution d'authentification transparente basée sur les interactions tactiles de l'utilisateur avec son appareil est utilisable à 90% du temps. Afin de pouvoir continuer à authentifier l'utilisateur dans les 10% restant, il est nécessaire d'utiliser d'autres méthodes d'authentification. Une alternative naturelle pour authentifier l'utilisateur est de l'authentifier lorsqu'il marche. Ainsi, les auteurs de [31], utilisent les données issues de l'accéléromètre afin d'identifier les cycles de marches d'un utilisateur et de l'authentifier en utilisant l'algorithme de déformation temporelle dynamique. ils obtiennent un FRR (taux de faux rejet) de 10.7% pour un FAR (taux de fausse acceptation) de 1.4% (voir section 3.5.1). Enfin, ces données peuvent être combinées avec des objets connectés. En combinant les

données capturées avec le téléphone et une smartwatch, les auteurs de [74] obtiennent un taux de reconnaissance de 98%.

2.4.5 Habitudes

Le téléphone mobile regorge d'applications différentes où il est possible d'extraire des données sur les habitudes d'usage du téléphone. Parmi les sources possibles, nous retrouvons l'utilisation des données de géolocalisation pour authentifier un utilisateur [44, 77] qui sont généralement combinées avec les autres capteurs du téléphone mobile afin d'obtenir une base d'enrôlement dépendant du contexte [69]. Ces données d'usage peuvent aussi être combinées aux habitudes d'appel de l'utilisateur [76], aux textes recueillis dans les SMS [101] ou encore à l'usage des applications et aux interactions Bluetooth [85].

2.4.6 Systèmes multi modaux

De plus, pour obtenir une authentification dite forte d'après les standards [61], il est nécessaire de présenter plusieurs modalités de types différents. Afin de pouvoir s'adapter aux normes d'authentification régissant le web, de satisfaire aux exigences de sécurités et de pouvoir authentifier en permanence, l'utilisation d'une seule modalité trouve donc ses limites.

Afin de palier à cette problématique, différents systèmes multimodaux existent. Les auteurs de [71] proposent un modèle d'architecture logicielle qui permet d'intégrer différents frameworks d'authentification implicite afin de verrouiller ou non l'accès au téléphone. Les auteurs de [26] exposent une solution basée sur l'utilisation de registre tampons des authentifications réalisées avec la voix, le code PIN et le mot de passe.

2.5 Objectifs de la thèse

L'authentification est indispensable au bon fonctionnement des services numériques. En effet, elle représente la première barrière de sécurité. Cependant, dans l'univers numérique, l'étape d'authentification est souvent implémentée comme un simple couple {login, mot de passe} à saisir par l'utilisateur.

Cette étape représente un problème à la fois en terme d'usabilité mais aussi en terme de sécurité. La mémorisation d'un mot de passe différent pour chacun des services représente une tache titanesque pour les utilisateurs qui utilisent des

solutions de contournement pour palier cette difficulté.

L'objectif de cette thèse est donc de proposer une alternative réaliste permettant d'utiliser une authentification utilisateur transparente et continue. Afin d'atteindre cette objectif, il est nécessaire de vérifier les trois points suivants :

— La sécurité
— L'usabilité
— La protection de la vie privée

|1em Enfin, une solution industrielle permettant à Orange de fournir ce service doit être démontrée.

2.6 Conclusion

Dans ce chapitre, nous avons vu comment définir l'authentification. Il s'agit d'établir la confiance entre plusieurs entités qui peuvent être des personnes morales ou physiques au travers d'infrastructures techniques.

Pour permettre d'authentifier un utilisateur, nous avons vu qu'il était nécessaire de relier des preuves à son identité. Cette identité numérique est constituée d'attributs dont certains peuvent être prouvés. La preuve de la possession d'un facteur d'authentification est un attribut prouvé. Dans le cadre de cette thèse, ce sont les attributs prouvés qui nous intéressent plus particulièrement. Ils sont la réponse naturelle à la problématique de l'authentification.

Une authentification faisant appel à plusieurs facteurs d'authentification et plusieurs canaux de communication est généralement appelée authentification forte. Il s'agit cependant d'une notion subjective et non exhaustive. Il est préférable de voir le niveau de confiance apporté lors de l'authentification en fonction des besoins et d'adapter ce niveau de confiance aux besoins.

Le processus d'authentification est soumis à un certains nombre d'exigences. Nous avons défini les trois exigences principales qui sont le respect de la vie privée, la sécurité et l'usabilité du système. Les données des utilisateurs, et en particulier les données biométriques, sont des données personnelles. Elles sont donc la propriété des utilisateurs et le fournisseur d'identité se doit de s'assurer, à la fois lors du stockage mais aussi de l'utilisation, de divulguer le minimum d'information possible sur ces données.

Le processus d'authentification étant une étape clé de la sécurité des systèmes, il est nécessaire d'évaluer les protocoles d'authentification en terme de sécurité vis à vis d'attaquants potentiels. Enfin, l'usabilité d'un système d'authentification est un facteur d'acceptation par les utilisateurs finaux et une contrainte nécessaire au bon fonctionnement du système.

L'objectif de cette thèse est donc de pallier les manques dans l'évaluation de la confiance et dans l'usabilité des systèmes d'authentification.

Dans le chapitre suivant, nous dressons un état des techniques visant à réaliser une authentification des utilisateurs dans la littérature.

Chapitre 3

Collecter et traiter les données issues du téléphone mobile

Ce chapitre liste les données collectables depuis le téléphone mobile. Après énumération des données exploitables, nous nous intéressons aux mécanismes permettant de les protéger et de les analyser dans un but d'authentification. Cette analyse est réalisée avec des classifieurs. Enfin, nous exposons les méthodes d'évaluation que nous réutiliserons dans le chapitre suivant pour comparer nos travaux à l'état de l'art.

Sommaire

3.1 Introduction

Les téléphones mobiles sont devenus de véritables compagnons de vie. Ils nous accompagnent dans l'ensemble de nos tâches. Nous y stockons, souvent par l'intermédiaire du "Cloud", nos informations personnelles et professionnelles. Par ces aspects, ils sont donc les témoins privilégiées de nos habitudes et de notre comportement.

En parallèle de cela, les téléphones mobiles sont maintenant des smartphones et disposent d'une puissance de calcul comparable aux ordinateurs de bureautiques. De plus, ils disposent de nombreux capteurs leur permettant d'avoir conscience de l'environnement et du contexte.

Enfin, les téléphones mobiles sont hyper-connectés. On retrouve maintenant sur les mobiles des connexions vers les réseaux les plus courants tels que le bluetooth, le Wifi et bien sûr les réseaux des opérateurs téléphoniques (2G, 3G, 4G et maintenant 5G).

L'ensemble de ces aspects permet d'envisager ces compagnons comme de formidables outils de collecte de données sur notre comportement. Dans la suite de ce chapitre, nous allons aborder les données collectables. Nous aborderons ensuite concrètement comment ces données peuvent être récupérées et quelles sont les éventuelles limitations techniques liées à cette collecte. Enfin, si les données représentent un enjeu scientifique important, elles n'en restent pas moins la propriété de l'utilisateur. Afin de pouvoir les manipuler et les utiliser dans le respect de la vie privée de l'utilisateur, nous établirons les solutions qui permettent de protéger ces données.

3.2 Collecte d'informations sur mobile

L'écosystème d'un téléphone mobile se compose de services installés ainsi que des connexions avec les réseaux et la capacité de collecte matérielle. Le marché des smartphones se compose de deux grands acteurs qui sont les téléphones Android et les téléphones IOS. Si l'écosystème des téléphones Apple est relativement simple car propriétaire, les téléphones mobiles Android proposent une large variété de modèles dont les caractéristiques diffèrent d'un constructeur à l'autre (voir table 3.1).

Malgré cette diversité, les fonctionnalités proposées sont similaires tant en terme de capteur disponible qu'en terme de logiciels proposés. Nous allons maintenant proposer un formalisme pour décrire les informations collectables depuis un téléphone mobile.

3.2.1 Formalisme et définitions

L'environnement des téléphones mobiles propose un certain nombre d'informations qu'il est possible de récupérer de manière logicielle. Cependant, la diversité des

Nom	Orange Nura	Samsung Galaxy A5
Dimensions	150 x 76 x 8.5 mm	144.8 x 71 x 7.3 mm
Poids	165 g	155 g
Processeur	ARM Cortex A7 - 1.2 GHz	ARM Cortex A53 - 1.6 GHz
Nombre de coeurs	4	8
Mémoire vive (RAM)	1 Go	2 Go
Capacité de la batterie	3100 mAh	2900 mAh
Capteur photo	8 Mpx	13 Mpx
Système d'exploitation	Android 4.4	Android 5.1
Capteur d'empreintes	Non	Oui

TABLE 3.1 – Comparaison de deux mobiles Android

informations disponibles rend parfois difficile la compréhension des données.

Afin de simplifier la lecture et la récupération des données, nous utilisons le formalisme exposé dans la table 3.2 pour décrire la donnée.

Type	Description
Int	Entier
Double	Nombre décimal
String	Chaîne de caractères
Enumerate	Elément parmi une liste finie
List[*]	Liste d'éléments

TABLE 3.2 – Formalisme du type de données récupérées

Par exemple, si une donnée est une liste de chaînes de caractères quelconque, nous la noterons *List[String]*. Afin de décrire précisément les données, nous avons aussi fait le choix de les classer selon les catégories suivantes :

Definition 7. ***Données issues d'éléments matériels*** *: Une donnée issue d'un élément matériel est une donnée dont la source d'information est un capteur matériel du téléphone mobile. Il s'agit donc d'une mesure physique de l'environnement du mobile. Une évaluation de l'incertitude de cette mesure est généralement disponible avec cette mesure.*

Definition 8. ***Données issues d'éléments logiciels*** *: Une donnée issue d'un élément logiciel est une donnée ayant pour origine une action logicielle. Il peut s'agir d'une information mesurée suite à une interaction avec un logiciel ou avec le système d'exploitation du téléphone mobile.*

Definition 9. ***Données issues d'éléments du réseau de l'opérateur*** *: Une donnée issue d'un élément du réseau de l'opérateur est une donnée générée à partir des éléments du réseau de l'opérateur.*

3.2.2 Les données issues de capteurs biométriques

Les facteurs corporels utilisables à des fins d'authentification sur téléphone mobile sont de plus en plus répandus. Ainsi, Samsung et Apple ont récemment mis en place un capteur d'empreintes digitales sur leur téléphone mobile, le second utilisant d'ailleurs l'authentification par empreinte digitale dans le système Apple Pay. La reconnaissance faciale a aussi été intégrée depuis Android 4.0 Ice Cream Sandwich [1]. Clarke [24] propose d'utiliser le lobe de l'oreille afin d'authentifier le porteur d'un téléphone mobile lorsqu'il passe un appel téléphonique.

Les capacités de calcul des téléphones sont de plus en plus proches de celles des micro-ordinateurs et permettent d'envisager des calculs plus importants. C'est particulièrement le cas sur les smartphones. Ainsi, Kang [68] décrit une implémentation sur mobile de reconnaissance de l'iris.

La reconnaissance vocale est un facteur morphologique (cordes vocales) et comportemental. Sur téléphone mobile, la complexité provient essentiellement de l'échantillonnage, qui est réalisé dans un environnement non contrôlé. Dans [73], les auteurs évaluent différents systèmes de reconnaissance vocale sur la base de données MOBIO. Cette base de données a été construite avec des enregistrements vocaux réalisés sur téléphone mobile.

L'utilisation de la forme de la main peut aussi constituer un facteur d'authentification. Sae-Bae et Jakobsson [100] utilisent les capacités multi-touch et la taille des écrans des tablettes pour authentifier un utilisateur par rapport aux caractéristiques morphologiques de sa main. D'autres solutions utilisent l'appareil photo des téléphones mobiles afin de réaliser cette reconnaissance [30, 21]. Ils ont l'avantage de disposer de davantage de caractéristiques mais sont facilement victimes d'attaques par rejeu puisqu'ils ne disposent pas, pour le moment, de mécanismes prouvant la vitalité de la main. Choraś et Kozik [21] se concentrent sur les lignes de l'articulation des doigts afin de réaliser une authentification multimodale avec les lignes caractéristiques de la paume.

1. http://developer.android.com/about/versions/android-4.0-highlights.html

TABLE 3.3 – Capteurs disponibles sur un téléphone mobile

Capteur	Nature de la donnée	Unité	Type de donnée	Lien avec l'utilisateur
Accéléromètre	accélération x, y, z	$m.s^{-2}$	double, double, double	Mouvement de l'utilisateur
Gyroscope	vitesse de rotation x, y, z	$rad.s^{-1}$	double, double, double	Mouvement de l'utilisateur
Détecteur de pas	Détection d'un pas	Sans unité	bool	Mouvement de l'utilisateur
Rotation	Rotation autour des axe x, y, z	Sans unité	double, double, double	Position du téléphone dans la main, poche, ...
Champ magnétique	Champ magnétique	μT	double, double, double	Localisation du téléphone
Orientation	Azimut, roulis, tangage	°	double, double, double	Position du téléphone dans l'espace
Capteur de proximité	Proximité avec un objet	cm	double	Distance du téléphone à l'oreille, présence dans la poche
Température	Température du téléphone	$C°$	double	
Température ambiante	Température ambiante	$C°$	double	
Luminosité	Luminosité	lx	double	
Pression	pression atmosphérique	hPa	double	
Hygromètre	Humidité relative	%	double	
GPS	Position GPS	$°, min, sec$	double, double, double	

3.2.3 Les données issues d'éléments matériels

Les téléphones mobiles possèdent des capteurs permettant de remonter des informations de l'environnement. Ces capteurs retournent une mesure possédant une unité physique. Par exemple, les capteurs d'accélération remontent une valeur exprimée en $m.s^{-2}$. En plus de cette valeur, l'incertitude liée à la mesure est disponible.

A cela, s'ajoute l'ensemble des capteurs d'acquisition du téléphone tels que la ou les caméra(s), l'écran tactile, le microphone.

L'ensemble de ces données étant liées à un capteur physique du téléphone mobile,

il est nécessaire d'alimenter ce capteur afin d'obtenir une valeur. Ceci amène donc une consommation de batterie supplémentaire.

3.2.4 Les données issues d'éléments logiciels

Bien qu'accessibles via des API logicielles, les données précédentes nécessitent de posséder le capteur sur son téléphone mobile. D'autres données sont liées uniquement aux logiciels et aux interactions de l'utilisateur avec les applications. L'ensemble des applications du téléphone sont à même de stocker des données. Pour cette raison, il est impossible de lister l'ensemble des données possibles car il en existe autant que d'applications possibles. On peut par exemple citer le bluetooth, les ouvertures fermetures des applications ou encore les données générés par les contacts téléphoniques.

Ces dernières, sont d'ailleurs souvent des données considérées comme sensible par l'utilisateur. Il est donc important d'utiliser des méthodes de protection des données personnelles afin de les protéger. Nous abordons ce point dans la section suivante.

3.3 Protection des données personnelles

Avant de dresser les tecniques possibles de protection des données personnelles, nous listons les propriétés recherchées.

3.3.1 Propriétés recherchées

Nous utilisons la terminologie employée dans le rapport technique de Pfitzmann et Hansen [93] (voir section 2.3.1). Dans cette terminologie, l'élément central que nous cherchons à protéger est le point d'intérêt [2]. Nous définissons dans le cadre de ces travaux un point d'intérêt de la manière suivante :

Definition 10. *Toute donnée qui prise indépendamment ou combinée avec d'autres révèle une information que l'utilisateur souhaite garder secrète.*

Pour qu'un système d'authentification basée sur des éléments collectés depuis un téléphone mobile soit respectueux de la vie privée, il est nécessaire de rendre indétectable les points d'intérêt. Il est ajouté à cela des contraintes réglementaires. Ces contraintes sont liées à la Commission Informatique et Libertés (CNIL). Parmi ces contraintes, on peut lister :

— Le droit d'accès

2. Traduit de l'anglais : Point of Interest

— Le droit de rectification
— Le droit d'opposition

Contrairement à des clés cryptographiques ou à un mot de passe, les données collectées ont une durée de vie considérablement plus longue. Au même titre que des données biométriques, les données issues d'un téléphone mobile ne peuvent pas être révoquées. Il est donc nécessaire en plus de l'indétectabilité, de les rendre révocables.

Pour résumer, les données collectées doivent respecter les propriétés d'**indétectabilité** et de **révocabilité** afin de pouvoir être utilisées dans un protocole d'authentification.

3.3.2 Conservation des données sur le téléphone mobile

La solution la plus triviale afin de protéger les données est de ne pas les récupérer. Plusieurs propositions [91, 81] ont été faites sur des modèles proposant cette solution. Dans le cadre d'une authentification à des services, cela suppose que les calculs soient réalisés sur le téléphone mobile et que le résultat soient transmis par la suite.

Les calculs nécessaires sont alors à prendre en compte dans l'usage du téléphone, en particulier la consommation énergétique. Les auteurs de [81] évaluent l'intérêt des différents capteurs du téléphone mobile pour l'authentification.

Cependant, cette solution ne permet pas d'envisager une extension avec des protocoles prenant en compte plusieurs supports (mobile, smartwatch...) et est difficile à adapter à des services web.

3.3.3 Fonction de hachage

Les fonctions de hachage sont des fonctions qui admettent les propriétés suivantes :
— Non inversibilité : Il n'est pas réaliste de reconstruire la donnée initiale à partir de son hash.
— Déterminisme : Pour une entrée donnée, la sortie est toujours la même.
— Résistance aux collisions : Il n'est pas réaliste de trouver deux entrées possédant la même sortie.

Les fonctions de hachage sont généralement standardisées à la suite de concours. On peut citer les standards SHA2 [87] et SHA3 [6]. Dans le cadre de nos recherches, nous considérerons ces propriétés afin de créer des empreintes uniques qui correspondent à des événements de la vie de l'utilisateur.

3.3.4 Chiffrement classique

Utiliser une méthode de chiffrement classique comme AES [42] permet d'assurer la confidentialité des données transmises. Cependant, il est nécessaire de déchiffrer

le contenu pour pouvoir authentifier. Le serveur a donc accès aux données en clair durant la vérification. Par exemple, le système de paiement par reconnaissance vocale de La Poste est protégé avec cette méthode (voir Délibération n°2016-037 du 18 février 2016 délivré par la CNIL).

3.3.5 Chiffrement homomorphe

Le chiffrement homomorphe est proposé comme une solution de gestion de données personnelles respectueuse de la vie privée. Effectuer une comparaison au sein de ces schéma n'est pas aisé parce que les modèles de sécurité et le nombre d'acteurs ne sont pas les mêmes d'un schéma à l'autre. Par exemple, l'acteur garant de la clé secrète change ne manière significative l'analyse de sécurité.

Bringer et al [16] combinent le schéma de chiffrement de Goldwasser-Micali avec un protocole PIR (Private Information Retrieval) dans un schéma d'identification biométrique où la comparaison des données biométriques est effectuée avec une distance de Hamming (par exemple pour un iriscode). La partie serveur de leur schéma est séparée en trois entités :

— La base de données
— Le comparateur
— Le serveur d'authentification

La base de données stocke les données en clair, alors que le comparateur possède la clé secrète pour le déchiffrement. De ce fait, la protection des données personnelles n'est assurée que dans un modèle global, honnête mais curieux, sans collusion entre ces entités.

Plus tard, différents schémas ont été proposés en combinant le schéma de chiffrement de Paillier avec un protocole interactif (transfert inconscient/circuit brouiller) pour un système d'identification par reconnaissance faciale dans lequel la comparaison des données est basée sur une distance euclidienne (par example Osadchy et al [90]). Des techniques similaires sur les empreintes digitales et l'iris (avec une distance de hamming ou euclidienne) sont présentées par Blanton et Gasti dans [8]. Tous ces schémas utilisent un modèle honnête mais curieux, basés sur une base de données dans laquelle les données biométriques sont stockées en clair, comme précédemment.

Un protocole privé non interactif est proposé par Troncoso-Pastoriza et al pour de l'authentification par reconaissance faciale [114]. Dans ce protocole, la donnée biométrique est chiffrée avec un schéma de entièrement homomorphe de Gentry. Cela entraine un coût de communication relativement important (autour de 400 Mo). Un autre schéma, permettant d'utiliser des données biométriques chiffrées est proposé par Gomez-Barrero et al sur les signatures [50]. Néanmoins, dans leur schéma, le

serveur est de nouveau séparé en deux entités : une base de données et un serveur d'authentification, où ce dernier possède la clé secrète, ce qui entraine une brèche de sécurité s'il y a collusion entre ces entités.

Délocaliser l'authentification est une autre approche utilisée dans le contexte de l'authentification biométrique sur mobile. Ainsi, Carter et al [19], et plus tard, Gasti et al [46] réalisent des circuits brouillés délocalisés pour effectuer des comparaisons avec des distances de hamming sur des données personnelles comme la localisation courante ou les contacts du mobile. Trois acteurs sont impliqués :

— Le téléphone mobile
— Le serveur d'authentification
— Le cloud

Là encore, le modèle de sécurité ne prend pas en compte la collusion entre le serveur et le cloud.

Le premier schéma proposant de l'authentification transparante sur mobile dans lequel la clé est possédée par l'utilisateur est proposé par Shahandashti et al [104]. Ce schéma est une combinaison d'additions de chiffrés homomorphes avec un chiffrement conservant l'ordre. Le schéma est implémenté avec de la dynamique de frappe au clavier par Halunen and Vallivaara [55]. Cependant, les chiffrements conservant l'ordre souffrent de plusieurs problèmes de fuites de données [37]. Plus tard, le schéma d'authentification transparent proposé par Domingo-Ferrer [36], combine le schéma de chiffrement de Paillier avec un protocole d'intersection des ensembles. Cependant, ce protocole est coûteux en temps d'exécution, en particulier à cause du nombre important d'exponentiations modulaires requis par la machine de l'utilisateur.

3.3.6 Biométrie révocable

La biométrie révocable est une approche consistant à transformer une donnée biométrique de façon non inversible et conservant la similarité. Cette transformation est généralement paramétrée par une clé permettant de révoquer le résultat de la transformée en changeant de clé.

Pour une vue d'ensemble des schémas de biométrie révocable, nous vous renvoyons au papier[96]. L'algorithme de BioHashing a été le premier décrit dans [47] et [113], il a été respectivement appliqué à la reconnaissance faciale et aux empreintes digitales.

Le biohashing est un algorithme permettant de protéger des données biométriques. Les méthodes de traitement étant assez similaires entre les habitudes d'usage du téléphone mobile et des données biométriques, il est possible d'adapter cet algorithme aux données personnelles de manière générale. En effet, les données biométriques

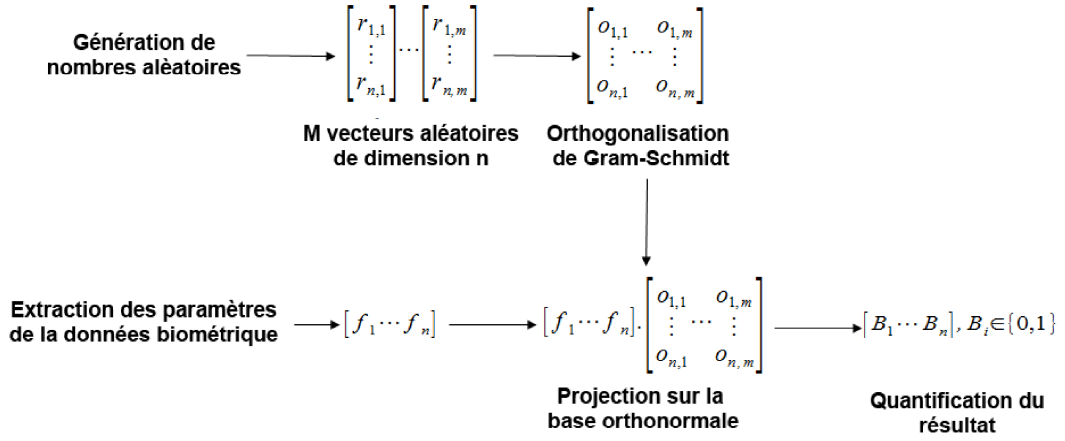

FIGURE 3.1 – Processus de biohashing

sont, par nature, non révocables et très sensibles. L'algorithme de biohashing est un cas particulier de biométrie révocable [95, 9].

Le concept de la biométrie révocable réside dans le fait de transformer une donnée brute afin de résoudre à la fois les problèmes de protection de la vie privée et de sécurité. Le principe général consiste à générer un nouveau template biométrique, à partir d'un vecteur de données biométriques et d'un secret.

Il peut ainsi être vu comme un double facteur d'authentification. Le principe est le suivant : La fonction de transformation du Biohashing combine une clé secrète qui appartient à l'utilisateur K avec les attributs biométriques qui doivent être exprimés comme un vecteur de taille finie : $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$.

Afin d'assurer une protection efficace, il est nécessaire de stocker cette clé K de manière sécurisée. Une solution qui se prête particulièrement aux smartphones car ces derniers disposent d'un élément sécurisé, la carte SIM, dans lequel la clé peut être stockée [56].

Le processus de Biohashing est divisé en deux étapes (voir figure 3.3.6) :

— Projection aléatoire : la clé K est utilisée comme une graine pour générer m vecteurs aléatoires $r_j \in \mathbb{R}^n$, $j = 1, \ldots, m$ et $m \leq n$. Après orthonormalisation avec la méthode de Gram-Schmidt [7], on obtient une matrice $O = (O_{i,j})_{i,j \in [1,n] \times [1,m]}$. Une projection du vecteur d'attributs biométriques $(f_1, \ldots, f_n)$ est alors calculée.

— Quantification : Cette étape a pour but de transformer les valeurs réelles de notre vecteur d'attributs en valeurs binaires en utilisant un simple seuillage. Plus précisément, un vecteur binaire $B = (B_1, \ldots, B_m)$ appelé Biocode est obtenu depuis le vecteur précédemment projeté avec un seuillage. Le but de cette étape est de garantir l'irréversibilité du processus complet.

De cette manière, en combinant la sécurité de la clé aux données biométriques, les variations inter-classes augmentent alors que les variations intra-classe sont préservées.

En effet, la protection de la vie privée est garantie. De plus, la propriété de révocation est assurée automatiquement : si le template transformé est compromis, la clé secrète est remplacée pour générer de nouveaux BioCodes.

Initialement proposé pour des données biométriques, cette méthode peut s'appliquer facilement à tout vecteur de paramètres de taille fixe. La comparaison de deux BioCodes est une tâche simple et rapide car il suffit juste de calculer leur distance de Hamming.

3.4 Classification

Afin de traiter les données dans un mécanisme d'authentification, il est nécessaire de les classifier en deux catégories. Nous cherchons à savoir si la donnée récoltée est membre de la classe des utilisateurs légitimes ou de celle des imposteurs.

Ce processus s'appelle la classification. Dans le cadre d'une authentification, nous utilisons des algorithmes supervisés. Dans le cadre d'un apprentissage supervisé, nous disposons d'une base d'apprentissage qui permet d'entraîner nos algorithmes. Le processus de classification utilisent des algorithmes standards pour classifier les données biométriques. Ces algorithmes sont décrits ci dessous.

3.4.1 Règle empirique

Il est possible de classifier les données de l'utilisateur à partir de n'importe quelle règle empirique. Cette règle peut être liée à la fréquence d'apparition d'un événement [69], ou encore au type d'événement [75]. Ces solutions sont très adaptées aux téléphones mobiles puisqu'elles ne requièrent pas de calculs coûteux et peuvent offrir des performances remarquables. En contrepartie, elles sont généralement très spécifiques et demandent une étude pointue du fonctionnement du système à analyser.

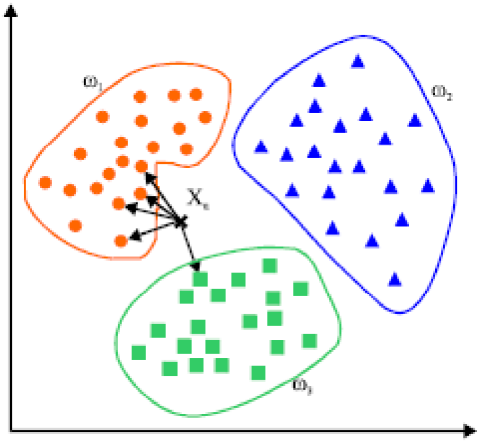

FIGURE 3.2 – Un point est attribué à une classe en fonction des ses plus proches voisins

3.4.2 K plus proches voisins

Cet algorithme crée l'espace des paramètres en plaçant les données d'entraînement dans un espace à n dimensions. Lorsqu'une nouvelle donnée doit être affectée à une classe, cette donnée est alors elle aussi placée dans le même espace. La classe qui lui est assignée est alors obtenue en regardant la classe à laquelle appartient ses k plus proches voisins (figure 3.4.2).

Ce classifieur est très intéressant à utiliser sur téléphone mobile puisqu'il ne demande que très peu de calculs. Il peut aussi être adapté afin de n'effectuer un apprentissage qu'avec une seule classe. Pour ce faire, la distance au plus proche voisin est alors calculée. Un seuil limite permet de déduire si la donnée proposée est celle de l'utilisateur légitime.

3.4.3 Séparateur à vaste marge

Ce classifieur est habituellement utilisé pour classifier deux classes de manière supervisées [11]. Le modèle représente les données comme des points qui sont divisés par un hyperplan dans le cas linéaire (figure 3.4.3) et l'on cherche à maximiser la marge représentée avec les tirets.

Chacune des deux classes est alors respectivement d'un côté de l'hyperplan. D'autres versions ont été proposées qui n'utilisent pas une base linéaire mais une base radiale (figure 3.4.3) permettant la définition de frontières de décion non linéaire.

Ces classifieurs sont utilisés dans [44]. Ils nécessitent cependant de posséder des données de deux classes différentes. La première classe représente les données de l'utilisateur légitime tandis que la seconde classe contient des données d'imposteur

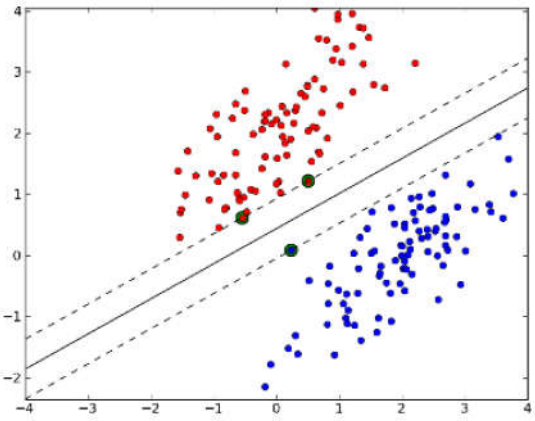

FIGURE 3.3 – SVM avec un noyau linéaire

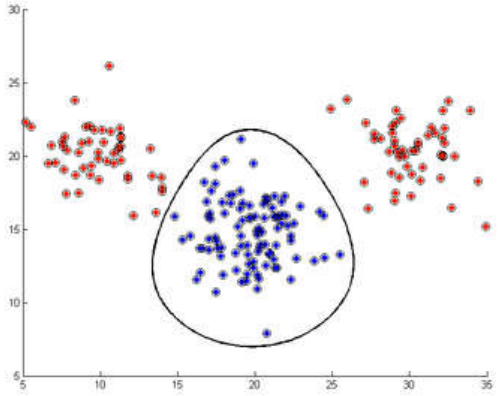

FIGURE 3.4 – SVM avec un noyau radial

(ou d'autres individus).

Afin de palier à cette problématique, il est possible de n'entraîner le classifieur qu'avec une seule classe. Pour cela, les données de l'utilisateur sont fournies en précisant une quantité de données aberrantes qui peut être exclue (figure 3.4.3).

3.4.4 Réseaux de neurones

Les réseaux de neurones sont inspirés de le modèle neuronal biologique présent dans le cerveau humain. Ce réseau consiste en une série de nœuds neuronaux qui sont inter-connectés de telle manière qu'ils peuvent être utilisés pour modéliser les relations complexes qui existent entre les entrées et les sorties. Un des usages des réseaux de neurones est de trouver des modèles au sein des données. Un réseau de

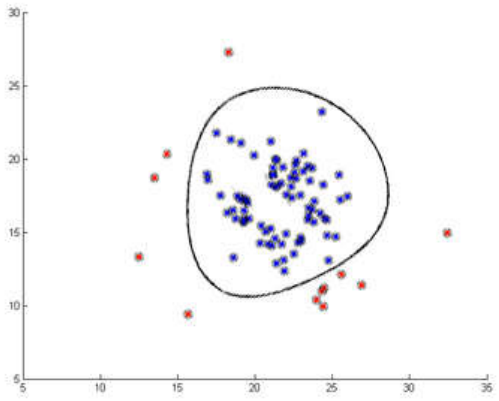

FIGURE 3.5 – One class SVM

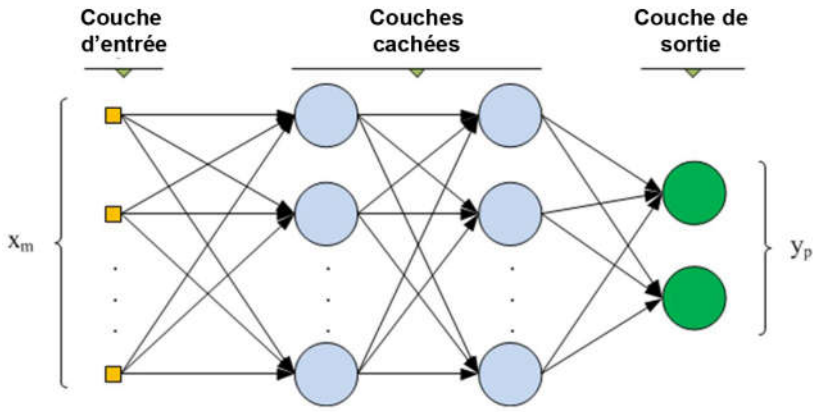

FIGURE 3.6 – Exemple de réseau de neurones

neurones basique (dit perceptron multi-couches) consiste en au moins trois couches :

— les entrées
— les sorties
— les couches cachées

Les nœuds de chacune des couches sont connectés entre eux de façon à ce que chaque nœuds passe sa sortie à chacun des nœuds de la couche suivante comme illustré par la figure 3.3.6. Les interconnexions peuvent être pondérées, et la phase d'apprentissage consiste à mettre à jour ces poids pour chacune des interconnexions. Un exemple est présenté dans la figure 3.4.4.

Généralement, les réseaux de neurones ont un bonne précision mais sont long à entraîner. Leur utilisation nécessite de posséder des données légitimes et des données d'imposture. On peut retrouver leur utilisation pour classifier des données comportementales sur téléphone mobile dans [77].

3.4.5 Discussion

Dans la pratique, il n'est pas toujours possible d'obtenir des données provenant d'attaquants. Ceci est en partie dû à des problématiques liées aux environnements de production ou encore liées à la protection des données personnelles (voir section 4.2).

Dans la suite de ce manuscrit, nous nous sommes intéressés aux classifieurs permettant de travailler avec une seule classe de donnée pour cette raison. Afin de valider le bon usage d'un classifieur, il est nécessaire d'estimer ses performances dans le cadre d'un mécanisme d'authentification. Cette évaluation est abordée dans la section suivante.

3.5 Évaluation

Afin d'estimer la performance de systèmes d'authentification, il est nécessaire de définir des métriques adaptées aux différents contextes.

3.5.1 Evaluer une authentification biométrique

Un système biométrique permet d'authentifier un individu par rapport à un enregistrement d'un modèle de l'individu. La ou les caractéristiques utilisées pour construire le modèle doivent être suffisamment uniques pour pouvoir différencier deux individus [62].

On peut décrire de manière formelle le problème de la vérification biométrique de la manière suivante :

$$(I, X_Q) \in \begin{cases} w_1, & \text{si } S\left(X_Q, X_I\right) \geq t \\ w_2, & \text{sinon} \end{cases}$$

Où w_1 est l'ensemble vrai (accepté), w_2 est l'ensemble faux (rejeté), I est l'identité dont un utilisateur se réclame, X_Q est le modèle extrait, X_I est le modèle à comparer et S est une fonction de similitude. On voit alors apparaître un seuil t. Celui-ci provient du fait que les acquisitions des données biométriques ne sont jamais parfaitement similaires.

L'efficacité d'un système d'authentification biométrique est évaluée en fonction du nombre de fausses acceptations et de faux rejets, c'est à dire en fonction du nombre

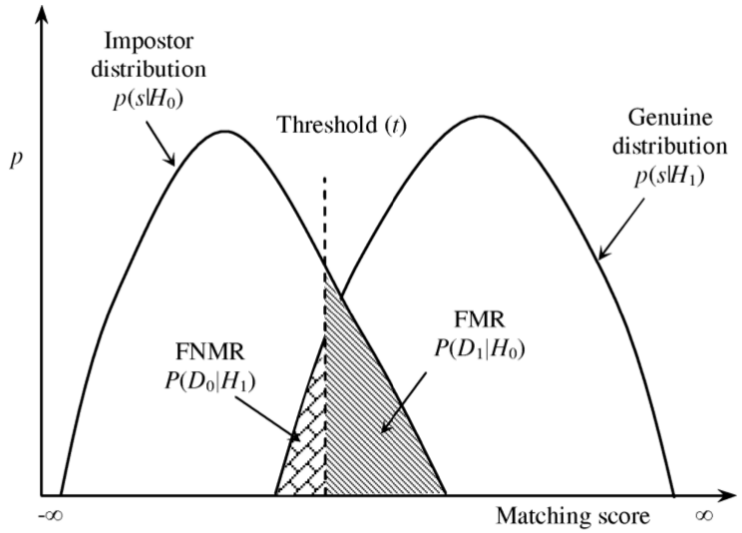

FIGURE 3.7 – Distribution du FMR et du FNMR [62]

de fois où un imposteur est accepté et du nombre de fois où un utilisateur légitime est refusé. Les mesures qui y sont associées sont appelées FMR (False Match Rate et FNMR (False Non Match Rate). Une distribution est présentée à la figure 3.7. Le FMR et le FNMR sont obtenus en testant les modèles des utilisateurs contre tous les autres modèles de la base de modèles. Nous pouvons faire une analogie avec les systèmes déterministes. Les systèmes déterministes sont généralement difficiles à utiliser. En effet, il est facile d'oublier un mot de passe ou une carte d'accès, ceci correspondrait au FNMR. Ces méthodes d'authentification reposent sur le partage d'un secret qu'il est possible de casser par force brute. A partir de cette observation, O'Gorman [89] propose de définir l'entropie pour les systèmes biométriques. Les utilisateurs choisissent généralement des mots de passe simples qu'il est possible de casser par recherche exhaustive ou à l'aide d'un dictionnaire. La résistance à ces attaques est équivalente au FMR.

3.5.2 Évaluer une authentification continue

Le concept d'utilisation de courbes de confiance n'est pas récent et est utilisé par exemple par [82, 12] afin de réaliser une authentification continue.

Le but de ce concept est de palier au manque d'usabilité liée à l'utilisation de méthodes d'authentification biométrique traditionnelles dans des systèmes d'authentification transparente. En effet, les systèmes traditionnels sont étudiés sous forme de bloc de données entraînant une ré-authentification de manière périodique.

Fonction de confiance

Afin de pouvoir réaliser une évaluation continue, il est nécessaire de définir la notion d'une fonction de confiance. Les bases de l'introduction d'une nouvelle métrique pour l'authentification continue, ont été influencées par Patrick Bours et ses collègues [12, 13] avant d'être repris plus largement dans le domaine de l'authentification continue [32].

Cette nouvelle métrique était nécessaire pour prendre en compte l'ajout de la temporalité dans l'authentification. Un système d'authentification ponctuel évalue un certain bloc de donnée et autorise ou non l'accès de manière définitive à un service. A l'opposé, un système d'authentification continue, évalue les actions de l'utilisateur instantanément afin d'en déduire s'il s'agit de l'utilisateur légitime. L'objectif d'une authentification continue est donc de détecter le plus rapidement possible un imposteur.

Pour cela, nous calculons un score similitude entre les données envoyées et les données stockées dans le template de l'utilisateur afin de mesurer un score similitude. Un seuil est alors appliqué à ce score. Si le score est au dessus de ce seuil, le niveau de confiance augmente, sinon, il diminue.

ANIA et ANGA

Si le FMR et le FNMR permettent de catégoriser un système d'authentification ponctuel, ces métriques trouvent leurs limites pour évaluer un système d'authentification continue. En effet, il est plus pertinent, de savoir quand un imposteur est détecté que si il est détecté. Pour être plus précis, il est important de savoir combien d'actions cet imposteur a pu réaliser avant d'être détecté. De même, pour l'utilisateur légitime, il est important d'évaluer la gêne occasionnée par un système d'authentification continue.

De la même manière que pour un imposteur, il est important d'évaluer le nombre d'actions que peut réaliser l'utilisateur légitime sans être déconnecté du service. En conclusion, il est nécessaire de mesurer les performances en terme d'ANIA et d'ANGA. En général, si un imposteur i lorsqu'il est évalué sur le template de l'utilisateur légitime g, est bloqué après les k actions $N_1, N_2, ..., N_k$. On peut alors définir :

$$ANIA_g^i = \frac{1}{k} \cdot \sum_{j=1}^{k} N_k \tag{3.1}$$

Comme l'ANIA qu'un imposteur i peut réaliser contre le modèle d'un utilisateur légitime g. Il est possible de déduire que l'ANIA contre un utilisateur légitime g est :

$$ANIA_g = \frac{1}{M-1} \cdot \sum_{i=1}^{M} ANIA_g^i \tag{3.2}$$

Enfin, nous obtenons l'ANIA contre n'importe quel utilisateur légitime du système :

$$ANIA = \frac{1}{M} \cdot \sum_{g=1}^{M} ANIA_g \tag{3.3}$$

La définition de l'ANGA est triviale et est obtenue de la même manière.

Equivalence avec le FMR et le FNMR

Dans [13], les auteurs proposent une équivalence entre FMR, FNMR et ANIA, ANGA. Dans un système d'authentification ponctuelle qui agit avec des blocs de N actions, n'importe quel utilisateur, que ce soit un imposteur ou un utilisateur légitime, peut réaliser ces N actions avant d'être authentifié.

Prenons la probabilité p que cet utilisateur soit reconnu comme l'utilisateur légitime et qu'il soit donc autorisé à réaliser les N actions suivantes. Après quoi, l'utilisateur est de nouveau authentifié en utilisant un bloc de N actions avec la même probabilité p d'être de nouveau authorisé à continuer à utiliser le service. De cette observation, il est possible de déduire que le nombre d'actions E que peut réaliser un utilisateur est égal à :

$$E = N + p.N + p^2.N + ... = \sum_{i=0}^{\infty} p^i.N = \frac{N}{1-p} \tag{3.4}$$

De cette formule, on peut alors déduire que pour un système d'authentification ponctuel, qui fonctionne avec des blocs de N actions et dont le FMR est égal à p, on a alors un ANIA égal à $N/(1-p) = N/(1-FMR)$.

On peut aussi l'exprimer en fonction du FNMR. Supposons le FNMR soit égal à q, alors cela signifie que l'utilisateur n'est pas authentifié sur son propre template avec une probabilité de q. On a donc $p = 1 - q$ où p est toujours la probabilité pour un utilisateur de continuer à utiliser le service. On trouve alors que l'ANGA est égal à $N/(1-p) = N/q = N/FNMR$

3.6 Discussion

Les téléphones mobiles et plus généralement les smartphones regorgent d'informations et nous accompagnent tout au long de notre journée.

Les smartphones sont équipés de nouveaux capteurs et intègrent de nombreuses méthodes d'authentification qu'elles soient matérielles, biométriques ou mémorielles.

La large gamme de capteurs et le temps que nous passons dessus permet d'envisager des métriques nouvelles pour l'authentification. Cependant, comme nous l'avons vu, ces données peuvent se révéler très intrusives. De plus, les attaques possibles sont d'autant plus importantes qu'il devient possible de combiner plusieurs bases de données entre elles. Par exemple, des données météorologiques peuvent trahir une position géographique et des données anodines peuvent alors se transformer en données sensibles.

Cependant, restreindre l'usage des données au seul téléphone mobile et ne pas les partager avec d'autre éléments empêcherait de nouvelles innovations dans le domaine de l'authentification. De plus, cela obligerait l'utilisateur à se réenroller aussi souvent qu'il change d'appareil.

Pour toutes ces raisons, il est nécessaire d'établir des modèles permettant de sauvegarder les données personnelles des utilisateurs contre les attaques potentielles.

Dans ce chapitre, nous avons vu les données collectables sur un téléphone mobile ainsi que des techniques de protection. Il est important de noter que certaines techniques de protection des données personnelles ne s'appliquent pas à toutes les données collectées.

Ainsi, il est impossible de construire une version unique d'un module d'anonymisation et l'architecture d'un système d'authentification dépend donc de ce module. Nous évaluons différentes propositions dans le chapitre suivant.

Chapitre 4

Contributions à l'authentification transparente

Ce chapitre expose trois méthodes d'authentification transparentes et garantes de la protection de la vie privée des utilisateurs. La première solution utilise un histogramme des habitudes des utilisateurs masquées à l'aide de fonctions de hachage. La seconde méthode utilise du chiffrement homomorphe pour protéger les données privées de l'utilisateur et la dernière méthode utilise le biohashing. Ces différentes approches sont comparées à l'état de l'art et entre elles afin de mettre en exergue leurs avantages et inconvénients.

Sommaire

4.1 Introduction

Les smartphones ont largement pénétré le marché [108] et possèdent une large gamme de capteurs. Cependant, leur usage est d'avantage associé à l'utilisation de données personnelles qu'à leur protection. Avec l'essor du Cloud et du tout connnecté, il est devient de plus en plus essentiel de ne pas se focaliser sur les données présentes

sur le téléphone mobile mais de se concentrer sur le téléphone comme outil d'accès à des données stockés en ligne.

4.2 Méthode 1 basée sur les fonctions de hachage

En réalisant une analogie avec les mots de passes, et la RFC 2898 [67], il est possible d'imaginer des techniques basées sur des fonctions de hachage afin de rendre indétectable les éléments d'intérêts. L'objectif est d'obtenir des données qui même si elles sont interceptées par un attaquant puissent être révoquées de la même manière que nous changeons nos mots de passe.

Le principe de cette RFC est de protéger les bases de mots de passe en associant le mot de passe à un sel avant de le hacher. De cette manière, si un attaquant récupère le hash, il lui est impossible d'obtenir des informations sur la donnée initiale. Cependant, les hashs, ont une propriété de diffusion. Il devient donc impossible de mesurer des distances entre données une fois hashées.

4.2.1 Événement

Chaque capteur du téléphone mobile fournit des informations mesurables qui caractérisent le contexte de l'utilisateur. Nous appellerons ces informations des événements. Ces événements peuvent être issus de données logicielles (usage des applications, durée d'un appel téléphonique) ou d'éléments matériels (l'accéléromètre, le gps...).

Il est alors possible de matérialiser une information par une valeur unique lorsqu'il s'agit d'une mesure discrète (comme le nom d'une application par exemple) ou bien il peut s'agir d'une valeur continue, dans le cas de la durée d'un appel téléphonique par exemple.

Dans le cas d'une valeur continue, il est nécessaire de la discrétiser dans le but de toujours obtenir un espace fini pour les mesures issues des capteurs du téléphone mobile. Ceci a aussi l'avantage de réduire le bruit lié à la mesure. Nous ne travaillerons alors qu'avec des valeurs discrètes.

Un événement est déclenché à chaque fois qu'une mesure est réalisée. Un évènement peut alors être défini dans ce cadre comme : une valeur discrète mesurée par un des capteurs du smartphone qu'il soit matériel ou logiciel. Par exemple, "l'application flappy bird est lancé" est un événement au même titre que "l'accéléromètre enregistre une accélération selon l'axe X de 2 $m.s^{-2}$". Nous observons alors que lorsque des événements arrivent de manière simultanée, cela constitue un facteur très discriminant pour authentifier une personne. Par conséquent, nous avons construit un modèle basé

sur le nombre d'occurrences des apparitions simultanées d'événements distincts. Nous appelons cela des associations d'événements. Nous avons observé expérimentalement, que les meilleurs résultats sont obtenus en associant trois événements distincts.

4.2.2 Association d'événements

Les associations d'événements sont définies dans notre système par un ensemble composé de plusieurs valeurs distinctes qui ont été mesurées pendant un intervalle de temps t, sans tenir compte de l'ordre d'apparition. Cet intervalle de temps est une période d'échantillonnage, et doit être la plus courte possible afin de détecter rapidement les imposteurs.

Dans le jeu de données que nous avons collecté (décrit dans la section 4.5.1), cette période d'échantillonnage est fixée à quelques minutes (entre une et trois minutes). Notre proposition ne considère que les valeurs distinctes. Si une valeur apparait deux fois pendant l'intervalle d'échantillonnage t, ceci est considéré comme une valeur unique.

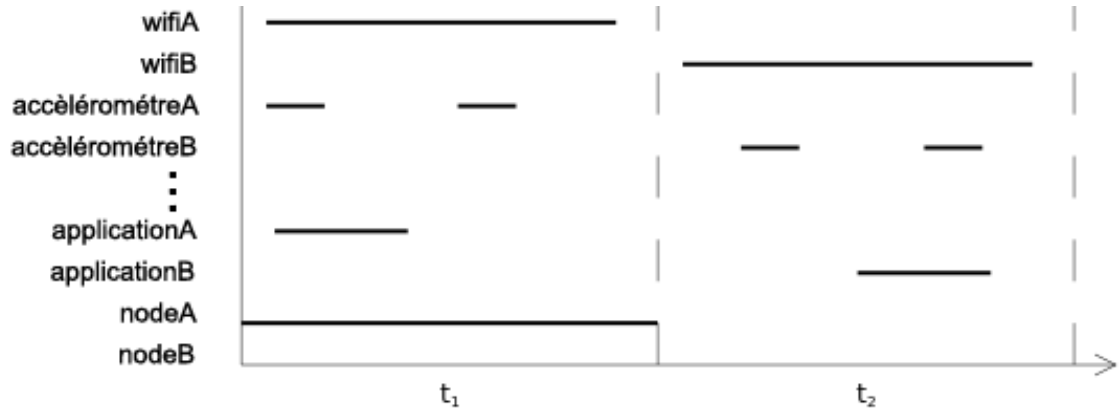

FIGURE 4.1 – Représentation d'association d'événements

Par exemple, sur la figure 4.1, pendant la période t_1, la même mesure d'accéléromètre apparait deux fois mais est considérée comme un seul événement.

Une association d'événements typique peut être : "Je suis connecté au wifi de ma maison ainsi qu'à l'antenne GSM proche de ma maison et j'utilise mon application bancaire". Concrètement, si nous observons la figure 4.1, pour la période t_1, les trois associations d'événements observables sont :

— {wifiA, accéléromètreA, applicationA}
— {wifiA, applicationA, nodeA}

— {accéléromètreA, applicationA, nodeA}

4.2.3 Transmission

Chaque association d'événements doit alors être masquée afin de pouvoir le transmettre au serveur de vérification. Pour cela, seule une signature de l'association des événements est envoyée :

$$h_i = HASH(e_i|K_u) \tag{4.1}$$

où e_i est une association d'événements, K_u est un secret qui n'est connu que par l'élément sécurisé u, et HASH, une fonction de hachage.

Ce calcul peut être réalisé dans un élément sécurisé du téléphone. La combinaison de l'architecture proposée et de l'équation 4.1 assure que les données de l'utilisateur restent secrètes. De plus, pour éviter les attaques par rejeu, il est nécessaire d'établir un canal de transmission sécurisé avec le serveur. En fonction du niveau de sécurité requis, il est possible d'utiliser l'élément sécurisé pour réaliser ce canal de communication sécurisé.

4.2.4 Enrôlement

Le vérifieur reçoit périodiquement les signatures d'associations d'événements h_i. Il stocke alors le nombre d'occurrences O_i pour chaque signature h_i dans une collection. Le vérifieur connait aussi le nombre total d'occurrences $T = \sum_{i=1}^{n} O_i$. Cette collection ainsi que le nombre total d'occurences qu'elle contient constitue le template de l'utilisateur. Il est possible de représenter ce template comme un histogramme comme illustré avec la figure 4.2.

Grâce à la fonction de hachage, de l'équation 4.1, il est impossible pour le serveur de récupérer de l'information sur les valeurs originales. En effet, le serveur ne connaissant pas K_u, il ne peut pas réaliser d'attaques par force brute pour deviner les valeurs de l'utilisateur. De cette manière, les données dont dispose le serveur ne peuvent pas être utilisées dans un autre but que l'authentification.

Les avantages de cette solution sont que le template peut s'adapter à n'importe quel téléphone mobile quelle que soit la précision ou la disponibilité des capteurs et que les profils enrôlés peuvent être révoqués simplement en changeant K_u, en cas de vol du mobile ou d'une fuite de la base de donnée.

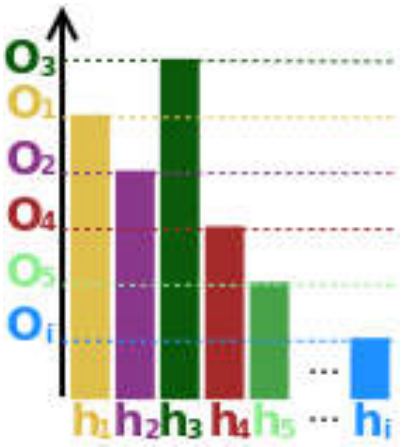

FIGURE 4.2 – Template utilisateur vu comme un histogramme

4.2.5 Authentification

Une fois que suffisamment de données aient été enregistrées pour construire un template consistant, le système passe en mode authentification. Cette transition peut se faire en fonction :

— Du nombre de signatures recueillis par le serveur
— Après un temps donné
— En testant le template afin d'obtenir un seuil de faux rejet donné

Du côté client, aucun changement n'est nécessaire pour passer en mode authentification. Le smartphone continue d'envoyer à intervalle régulier t les signatures des associations d'événements h_i au vérifieur.

Un système de bonus/malus est alors mis en place afin de récompenser les associations d'événements habituels. Le vérifieur extrait Q_i depuis le template, tel que :

$$Q_i = \begin{cases} O_i/T, & \text{si } O_i > 0 \\ P, & \text{sinon} \end{cases}$$

Nous rappelons que O_i est le nombre d'occurrences h_i dans le template de l'utilisateur, T est le nombre total d'occurrences dans la collection. P est une valeur arbitraire de pénalité, dont le rôle est de pénaliser les comportement inhabituels.

La valeur de P doit être fixée expérimentalement. Plus la valeur de P est grande, plus la confiance dans l'utilisateur va décroître rapidement. Durant la période d'échantillonnage t, le vérifieur obtient m signatures h_i et calcule le Δ à ajouter au score de confiance, en fonction de l'algorithme 1.

Algorithme 1 Algorithme Δ_L

Données :
B $\rightarrow$ Bonus de compensation
$Q_i \rightarrow$ Score de cohérence avec le template
$m \rightarrow$ nombre de signatures récupérées durant l'échantillonnage
$\Delta_{L-1} \rightarrow$ Delta précédemment ajouté au score de confiance

Résultat :
$\Delta_L \rightarrow$ Delta à ajouter au score de confiance

début
$\Delta_L = \left(B + \sum_{i=1}^{m} \frac{Q_i}{m}\right)$

si $sign(\Delta_L) == sign(\Delta_{L-1})$ **alors**
    $\Delta_L = \Delta_L + \Delta_{L-1}$
fin si
$\Delta_{L-1} = \Delta_L$
fin

Dans l'algorithme 1, B est une récompense qui permet d'accélérer la croissance de la courbe. Cela permet de compenser le décalage qui apparait dans le comportement de l'utilisateur entre l'enrôlement et les faits réels.

$$L_t = min(max(0, L_{t-1} + \Delta_L), 1) \tag{4.2}$$

Le score L_t calculé dans l'équation 4.2 et est borné dans l'intervalle $[0, 1]$ afin d'empêcher le score de confiance de croître ou de décroître à une valeur qui ne serait pas compensable.

Un exemple des scores de confiance obtenus avec cette méthode sur une période de dix jours est présenté dans la figure 4.3.

4.3 Méthode 2 basée sur la cryptographie homomorphe

Le chiffrement homomorphe permet de réaliser des calculs dans le domaine du chiffré. Ainsi, un serveur peut effectuer des calculs et retourner le résultat au client, sans déchiffrer les données. Ceci ne permet pas d'authentifier tel quel puisque le serveur ne peut pas récupérer le résultat. Il est possible cependant de réaliser un schéma permettant de vérifier la solution proposée par le client depuis le serveur, nous détaillons la solution retenue dans ce manuscrit dans le paragraphe suivant.

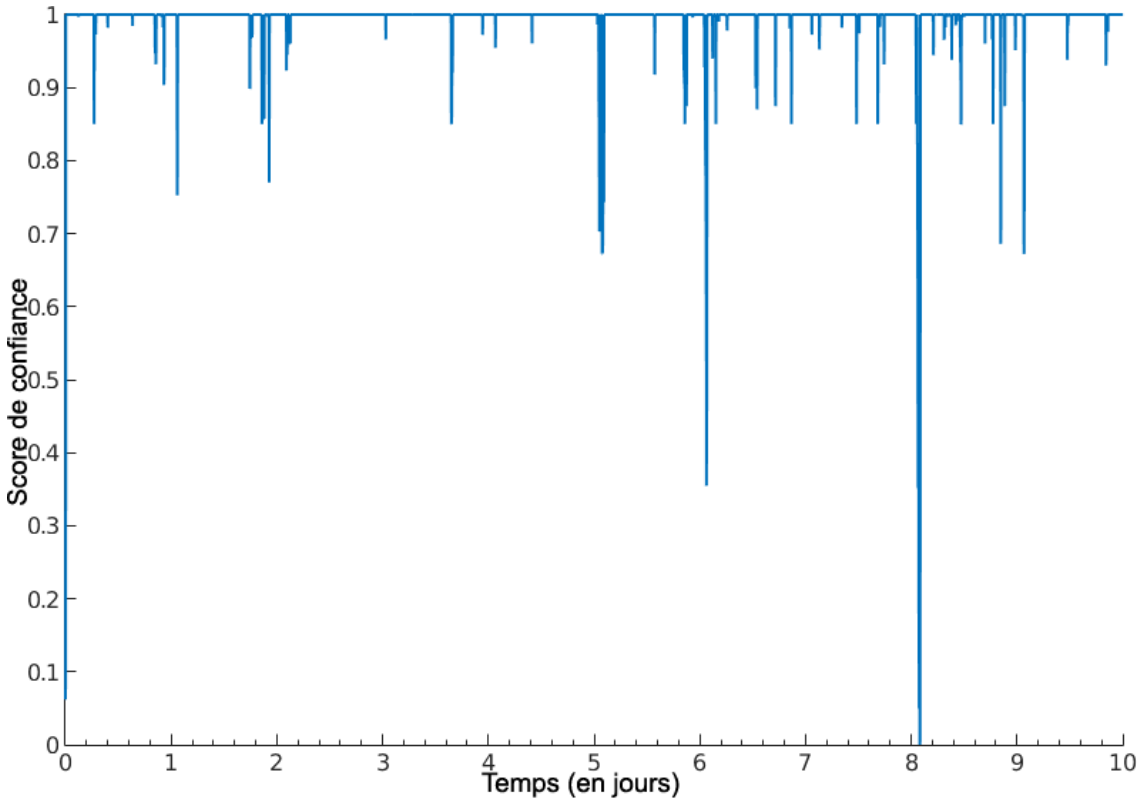

FIGURE 4.3 – Evolution deu score de confiance sur 10 jours

4.3.1 Le schéma de chiffrement de Goldwasser-Micali

Le cryptosystème Goldwasser-Micali (GM) est un schéma à clé publique développé par Shafi Goldwasser et Silvio Micali en 1982 [48, 49]. Il s'agit du premier schéma de chiffrement probabiliste.

Algorithme 1 : Génération des clés : K.

Cette algorithme se compose de quatre étapes :

1. Choisir deux nombres premiers de k bits, p_1 et p_2,
2. Calculer $n = p_1 p_2$,
3. Choisir $y \in \mathbb{Z}/n\mathbb{Z}^*$ de manière à ce que y est un non résidu modulo n (plus précisément, y a pour symbole de Jacobi $+1$, par exemple, nous avons $\left(\frac{y}{n}\right) = 1$),
4. La clé publique est (n, y) et la clé secrète, la paire (p_1, p_2).

Nous remarquons que le y choisi à l'étape 3 de l'algorithme de génération des clés peut être obtenu avec un algorithme probabiliste. En effet, nous choisissons un aléa y jusqu'à ce qu'il vérifie $\left(\frac{y}{p_1}\right) = -1$ et $\left(\frac{y}{p_2}\right) = -1$.

Algorithme 2 : Chiffrement : E.

Supposons que l'utilisateur B souhaite envoyer le texte chiffré c. Soit $m = (m_1, \ldots, \ldots, m_l) \in \{0,1\}^l$ le texte en clair et (n, y) la clé publique. Alors :

1. Pour $i = 1$ jusqu'à l, B tire un nombre aléatoire $x_i \in \mathbb{Z}/n\mathbb{Z}^*$, et en fonction de la valeur de m_i, il calcule :
   — Si $m_i = 0$, alors B calcul $c_i = x_i^2 \mod n$,
   — Sinon, il calcule $c_i = yx_i^2 \mod n$.
2. Le texte chiffré est alors $c = (c_1, \ldots, \ldots, c_l)$.

Algorithme 3 : Déchiffrement : D.

Soit (p_1, p_2) la clé privée, supposons qu'un utilisateur C souhaite déchiffrer le message chiffré $c = (c_1, \ldots, \ldots, c_l)$, alors,

1. Pour chaque $c_i \in c$, avec $i \in [1, \ldots, l]$, C calcul $e_i = \left(\frac{c_i}{p_1}\right)$:
   — Si $e_i = 1$, alors $m_i \leftarrow 0$,
   — Sinon, $m_i \leftarrow 1$.
2. Le message déchiffré est alors $m = (m_1, \ldots, \ldots, m_l)$.

Dans le cas du schéma de Goldwasser Micali, le déchiffrement correspond au calcul d'un symbole de Legendre, réalisé par une exponentiation modulaire.

Sécurité du schéma de Goldwasser Micali :

Assumant la difficulté du problème de la résiduosité quadratique, (par exemple, étant donné un entier $n = p_1 p_2$ et un entier dans $a \in n\mathbb{Z}$ avec $\left(\frac{a}{n}\right) = 1$ décider si a est un résidu quadratique modulo n ou non), le schéma de chiffrement de Goldwasser-Micali est schématiquement sécurisé.

Le schéma de chiffrement de Goldwasser-Micali possède des propriétés homomorphes. Soit c_0, c_1 les chiffrés des bits m_0 et m_1, nous avons alors :

$$\begin{aligned} c_0 c_1 &= x_0^2 y^{m_1} x_1^2 x^{m_2} \\ &= (x_0 x_1)^2 y^{m_1 + m_2}. \end{aligned}$$

Nous obtenons alors, $\mathsf{D}(c_0 c_1) = m_0 \oplus m_1$.

Le protocole que nous proposons est divisé en deux étapes : la phase d'enrôlement et la phase d'authentification. Il est applicable à toutes les données biométriques pour lesquelles une distance de Hamming peut être calculée. Afin d'illustrer ce protocole, nous utilisons par la suite un cas concret.

Pendant la phase d'enrôlement, les informations de chacun des appels téléphoniques (numéro appelé, localisation) du smartphone sont chiffrées en utilisant le

schéma de Goldwasser Micali avant de les envoyer à un serveur d'authentification. Durant la phase d'authentification, chacun des appels est directement envoyé au serveur de manière sécurisée, qui calcule alors la distance en fonction d'un enrôlement réalisé préalablement. Cette distance est calculée à l'aide de propriétés homomorphes.

4.3.2 Vue d'ensemble du protocole

Dans notre protocole, chaque message est un vecteur de 100 bits représentant la localisation depuis laquelle l'appel téléphonique a été passé (latitude et longitude) ainsi que le numéro de téléphone de l'appelé.

La phase d'enrôlement consiste à stocker les appels téléphoniques passés par l'utilisateur. L'utilisateur calcule alors un XOR bit à bit entre ce message et une valeur aléatoire gardée secrète. Afin de protéger la vie privée de l'utilisateur, chacun de ces messages est chiffré à l'aide de la clé publique de l'utilisateur en utilisant le schéma de chiffrement de Goldwasser-Micali.

Après l'enrôlement, chacun des appels déclenche un processus d'authentification. Dans notre protocole, le processus d'authentification est interactif. Tout d'abord, l'utilisateur calcule le XOR bit à bit entre le message généré à partir des informations de l'appel et l'aléa. Le résultat est chiffré en utilisant le schéma de Goldwasser-Micali avec sa clé publique. Ce chiffré est alors envoyé au serveur.

En utilisant les propriétés homomorphes du schéma de Goldwasser-Micali , le serveur est capable de calculer le XOR bit à bit du chiffré fraichement reçu avec chacun des messages liés à l'utilisateur dans sa base de données obtenant ainsi une distance de Hamming entre les informations de ce nouvel appel et chacun des templates stockés en base de données.

Le serveur a alors la garantie que le protocole d'authentification s'est déroulé convenablement, en utilisant les templates appropriés. Cependant, puisque le résultat est chiffré avec la clé de l'utilisateur, le serveur ne peut pas obtenir le résultat directement.

Le serveur a besoin que l'utilisateur déchiffre et retourne le résultat. Néanmoins, avant de renvoyer la distance chiffrée à l'utilisateur, le serveur chiffre avec la clé publique de l'utilisateur une valeur aléatoire de la même taille que le message. Il effectue alors un XOR bit à bit entre l'aléa et le message. L'utilisateur déchiffre ces valeurs, applique le XOR avec sa propre valeur aléatoire et retourne le résultat au serveur en utilisant un canal sécurisé comme TLS [35].

Ces valeurs aléatoires (de l'utilisateur et du serveur) agissent comme des protections "One Time Pad". Celui de l'utilisateur permet d'éviter au serveur de rejeter

le message envoyé par l'utilisateur et d'utiliser un message de son choix. La valeur aléatoire ajoutée par le serveur prévient l'utilisateur de récupérer de l'information sur les templates préalablement enrôlés ou de manipuler les distances. En effet, au moment de déchiffrer les distances, l'utilisateur ne récupère aucune connaissance à leurs sujets.

Dans le cas d'un utilisateur malicieux qui aurait volé un téléphone enregistré dans le système, ceci l'empêche de récupérer de l'information sur les templates enregistrés dans la base de données du serveur.

En particulier, il ne peut pas savoir si ses essais d'authentification sont proches de réussir ou non. Il ne peut ainsi pas réaliser d'attaque adaptative.

Le serveur récupère finalement, les distances en effectuant un XOR des valeurs envoyées par le client avec ses propres valeurs aléatoires. Il détermine le minimum qui sera la distance d'authentification. En fonction du seuil demandé par le service, cela permet d'authentifier ou non l'utilisateur.

Ce protocole garantie un processus d'authentification rigoureux, dans le sens où le serveur est assuré que la méthode d'authentification a été correctement appliquée sur les données reçues et que la distance servant à authentifier est celle qui a été calculée et enfin qu'un imposteur n'a aucun avantage à modifier ce score.

4.3.3 Description formelle

Nous décrivons d'abords la phase d'enrôlement. Cette étape est divisée en deux algorithmes : la génération des clés (Génération des clés) et la génération des templates et leur stockage (Template).

Génération des clés.

Cet algorithme est utilisé par l'utilisateur et génère une paire de clé de Goldwasser-Micali. Il retourne la paire de clés de l'utilisateur (n, y). La clé privée associée, la paire (p_1, p_2), est stockée de manière sécurisée dans le client mobile, par exemple, dans le keystore Android, qui est de plus en plus souvent protégé à l'aide d'un TEE (Trusted Execution Environment).

Template.

Cet algorithme génère k modèles chiffrés qui constituent le template envoyé au serveur. Soit M_i le message construit à partir des informations de l'appel téléphonique i par l'utilisateur pendant la phase d'enrôlement. Chacun de ces messages M_i contient : latitude, longitude, numéro appelé. Chaque M_i est un vecteur binaire de longueur

l. Ces $M_i = (M_{i,0}, \dots M_{i,l})$ sont chiffrés bit à bit avec le schéma Goldwasser-Micali, avec $T_i = (x_0^2 y^{M_{i,0}}, \dots, x_l^2 y^{M_{i,l}})$. Les chiffrés T_i sont alors envoyés au serveur. Le template pour cet utilisateur est $T = (T_1, \dots, T_k)$, où k est le nombre de templates nécessaires pour l'enrôlement.

Après la phase d'enrôlement, chaque appel déclenche le processus d'authentification. Ce processus est composé de quatre étapes : Mobile 1, Serveur 1, Mobile 2 et Serveur 2. Toutes les intéractions entre le téléphone mobile et le serveur sont réalisées en utilisant un protocole d'échange sécurisé comme TLS.

Mobile 1.

Les informations des appels (latitude, longitude, numéro de l'appelé) sont agencés dans un vecteur binaire de longueur l, $M = (M_0, \dots M_l)$. Le mobile génère un vecteur aléatoire r_u de longueur l et calcule un XOR bit à bit $\widetilde{M} = M \oplus r_u$. Ces $\widetilde{M}$ sont chiffrés bit à bit avec la clé publique du téléphone avec le schéma de chiffrement de Goldwasser-Micali, avec $C = (x_0^2 y^{\widetilde{M}_0}, \dots, x_l^2 y^{\widetilde{M}_l})$. Le texte chiffré C est ensuite envoyé au serveur.

Serveur 1.

Le serveur utilise la procédure homomorphe Eval pour réaliser un XOR entre le message reçu et chacun des templates stockés pour cet utilisateur. Pour chaque i, il calcule la distance chiffrée $D_i = (D_{i,0}, \dots, D_{i,l})$ comme suit :

$$\begin{aligned} D_i &= (D_{i,0}, \dots, D_{i,l}) \\ &= (T_{i,0} C_0, \dots T_{i,l} C_l). \end{aligned}$$

Grâce à la propriété homomorphe du schéma de Goldwasser-Micali, ce vecteur chiffre le XOR de $M_i \oplus M \oplus r_u$. Le serveur choisit alors k vecteurs aléatoires de l bits : r_i^{serv}. Il chiffre chacun de ces vecteurs bit à bit en utilisant la clé publique du téléphone mobile pour obtenir les vecteurs : R_i^{serv}. Le serveur utilise l'algorithme Eval pour calculer le vecteur de distance S_i de la manière suivante :

$$\begin{aligned} S_i &= (S_{i,0}, \dots, S_{i,l}) \\ &= (D_{i,0} R_0^{\text{serv}}, \dots D_{i,l} R_l^{\text{serv}}). \end{aligned}$$

Grâce à la propriété homomorphe du schéma de Goldwasser-Micali, ce vecteur est le chiffré de $M_i \oplus M \oplus r_u \oplus r_i^{\text{serv}}$. Ces vecteurs de distance chiffrés sont alors envoyés au téléphone mobile.

Mobile 2.

A la réception, l'utilisateur déchiffre les vecteurs de distance chiffrés S_i. Il obtient les valeurs $M_i \oplus M \oplus r_u \oplus r_i^{\text{serv}}$. Il supprime alors ses valeurs aléatoires r_u et renvoie $M_i \oplus M \oplus r_i^{\text{serv}}$ au serveur. Nous rappelons que ces valeurs doivent être envoyées via un protocole de communication sécurisée afin d'éviter les éventuels rejeux.

Serveur 2.

Le serveur récupère les distances $M_i \oplus M$, retire ses aléas r_i^{serv}. Il détermine la distance minimale parmi les $M_i \oplus M$ et génère une distance d'authentification. En fonction de la politique d'accès, il peut alors décider si l'authentification est fructueuse ou non.

4.4 Méthode 3 basée sur le biohashing

La partie suivante détaille l'architecture proposée, qui inclut l'algorithme du BioHashing.

4.4.1 Architecture

L'architecture globale est composée d'un client (le téléphone mobile) et d'un serveur d'authentification. Dans cet article, le téléphone mobile collecte les données comportementales de l'utilisateur quand celui-ci passe un appel ou envoie un sms. L'approche utilisée peut être étendue à d'autres échantillons issus de capteurs différents de ceux utilisés dans nos travaux. Ceci s'adapte particulièrement pour combiner des informations de géolocalisation avec n'importe quel autre groupe de capteurs.

Architecture cliente

Afin d'évaluer notre proposition, nous avons accès aux données suivantes :

— La position géographique de l'antenne depuis laquelle l'appel a été réalisé

— Le numéro de téléphone de l'appelé

Le processus de vérification est réalisé en ligne. Avant d'être envoyées au serveur, les données doivent d'abord être protégées avec l'algorithme de BioHashing. La figure 4.4 détaille cette architecture.

Une fois sous forme de Biocode, les données peuvent alors être envoyées en ligne. Afin d'éviter des attaques par rejeu, il est nécessaire d'utiliser un canal sécurisé. Ceci peut être fait en utilisant une connexion TLS [33].

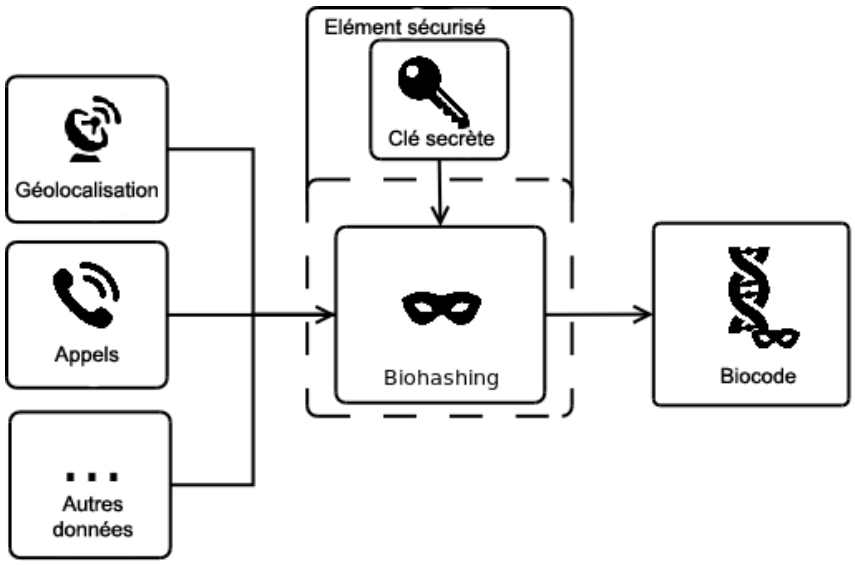

FIGURE 4.4 – Architecture côté client

Architecture serveur

Le serveur reçoit des BioCodes en continu, à chaque fois qu'un utilisateur passe un appel. La première étape est de stocker le BioCode en base. Les données stockées permettent de créer un template. Lorsque suffisamment de données sont enrôlées pour un utilisateur, le serveur passe alors en mode vérification. La figure 4.5 illustre cette architecture.

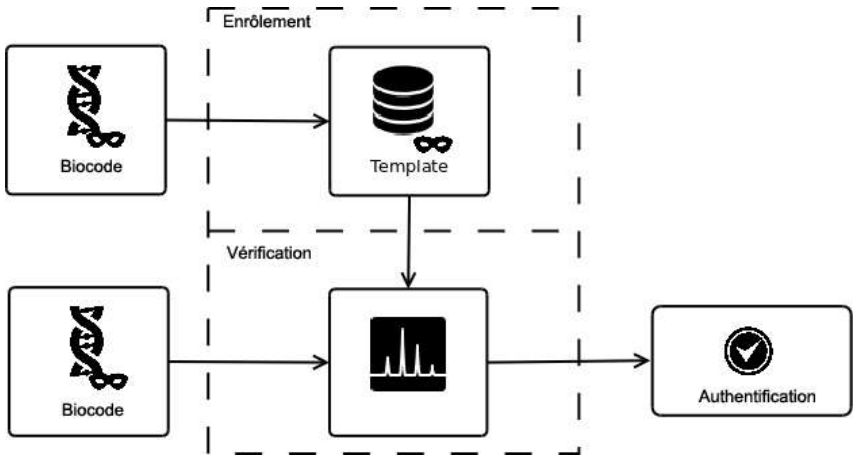

FIGURE 4.5 – Architecture côté serveur

4.5 Evaluation

Afin de comparer les différentes méthodes de protection de la vie privée proposées et choisir celles qui peuvent être adaptées à nos usages, nous avons réalisée une étude spécifique en fonction des caractéristiques propres de chacune des solutions. Nous évaluons ainsi :

— la pertinence des bases de données de tests
— les performances biométriques
— les contraintes d'implémentation
— la rapidité d'exécution

Dans la suite de cette section, nous exposons nos bases de données avant d'évaluer chacun des protocoles proposés un à un.

4.5.1 Base de données

Nous avons vu que les téléphones mobiles étaient équipés de nombreux capteurs. Afin de pouvoir utiliser ces données, nous devons procéder à une transformation de la donnée.

Cette transformation doit être effectuée sur le téléphone afin de garantir la protection des données personnelles de l'utilisateur. Afin d'atteindre cette objectif, il est nécessaire de préparer les données. Ce pré-traitement permet d'homogénéiser les données après leur collecte. Il est ainsi possible de dissocier la collecte des données de leur traitement et protection.

Dans un premier temps, nous listons les bases de données disponibles dans le contexte de la thèse.

MIT

Les paramètres et les utilisateurs sélectionnés dans la base de données du MIT pour conduire les expérimentations sont sélectionnés d'après la référence [76]. L'expérimentation sélectionne 76 participants qui ont le plus grand nombre d'entrées dans le log dont l'activité a eu lieu durant la période du 24/10/2004 au 20/11/2004. Cette restriction est mise en place afin de palier au manque observé d'activité chez les autres participants. Les données sélectionnées sont :

— les appels téléphoniques : numéro appelé,
— SMS : numéro appelé,
— bluetooth : addresse mac,
— localisation : identifiant de l'antenne GSM.

Collecte mobile

Nous avons réalisé une collecte de données sur téléphone mobile. La base de données consiste en un sous-ensemble de 11 personnes sur une collecte initialement réalisée sur 34 personnes durant trois semaines entre Juin et Juillet 2015.

La durée de collecte varie de deux semaines jusqu'à un mois dépendant de l'utilisateur. Le volume de données enregistrées est 10 fois supérieur à la base de données proposé par le MIT. La collecte a été réalisée sur smartphone Android avec une version supérieure à 4.0.2 et contient des données comme :

— Les application utilisées
— Le bluetooth
— Les identifiants des antennes GSM
— L'état de l'écran (allumé, éteint)
— Les appels téléphoniques
— Les SMS
— Les données d'accéléromètre
— Les données du gyroscope
— L'historique de navigation web
— Les points d'accès wifi

Les seuls critères demandés aux volontaires pour rejoindre l'expérimentation est d'installer l'application sur leurs téléphones Android. Une fois installée, l'application fonctionne en arrière plan et le processus de collecte de données est totalement transparent pour l'utilisateur. Si un capteur n'est pas activé ou n'est pas présent sur le téléphone mobile, alors les données associées ne sont pas enregistrées.

Cela garantie la faisabilité de l'implémentation d'un tel module de collecte de données sur n'importe quel smartphone. Aucun utilisateur n'a remarqué une consommation d'énergie plus importante durant la collecte.

Chacune de ces données est récoltée à l'aide d'un capteur logiciel. Les capteurs disponibles sont séparés en deux groupes :

— Les capteurs périodiques
— Les capteurs événementiels

Les capteurs périodiques enregistrent une valeur d'un capteur qui opère en continue une fois activé. Par exemple, le capteur récupérant l'identifiant des données GSM ou celui du champs magnétique sont périodiques. Les capteurs périodiques capturent une valeur toutes les trois minutes.

Les capteurs événementiels n'envoient une valeur que lorsqu'une action spécifique

est réalisée. Par exemple, le capteur chargé de récupérer les applications utilisées sur le téléphone, ne déclenche un événement que lorsqu'une action est réalisée par l'utilisateur (ouverture et fermeture d'une application)

Les données collectées sur le téléphone mobile sont alors envoyées à un serveur au travers d'une connexion TLS. Les données sont alors hachées sur le serveur en utilisant un sel aléatoire. Afin d'éviter des attaques par force brute, le sel a été jeté une fois la collecte de données réalisée. Ceci permet d'avoir un sel similaire pour tous les utilisateurs et ainsi de pouvoir simuler des impostures

Les données collectées ont permis d'obtenir un jeu de données complet et plus moderne que ceux disponibles. En effet, les interactions avec les appareils modernes sont plus nombreuses et les capteurs disponibles sont plus nombreux.

CRA

La base données utilisée pour évaluer ces travaux contient l'historique des communications de 100 personnes pendant un mois. Ces données sont extraites des informations du réseau d'un opérateur téléphonique. Les données présentes dans cette base sont :

— La latitude et la longitude de l'antenne
— Le numéro de l'appelant
— Le numéro de l'appelé
— Le type (Appel ou SMS)

TABLE 4.1 – Taille de la base d'expérimentation

Données	Enrôlement	Vérification
Max	666	666
Min	16	14
Moy	157.5	109

4.5.2 Méthode 1 basée sur les fonctions de hachage

Afin d'évaluer ce framework, une comparaison est réalisée à la fois sur la base de données que nous avons collectée et sur la base de données publique du MIT.

Les paramètres suivants ont été adoptés afin de conduire cette évaluation :
— $P = -10\%$,
— t de 2 à 5 minutes pour notre base de données,

— t de 5 à 15 minutes pour la base de données du MIT.

Afin d'assurer la meilleure usabilité possible, la valeur de B est fixée de manière automatique pour chaque utilisateur en fonction du score de confiance moyen obtenu avec les données d'entraînement. Ainsi, B est égal à $B = 1 - \bar{L}_{training}$

Usabilité

Parmi les paramètres disponibles dans notre base de données, nous sélectionnons :
— application : nom de l'application,
— appels : appels sortants (le numéro de téléphone n'est pas enregistré),
— SMS : messages sortants (le numéro de téléphone n'est pas enregistré),
— web : adresses des sites web,
— antenne GSM : Identifiant de l'antenne,
— bluetooth : adresse MAC,
— wifi : adresse MAC des réseaux.

La courbe de FNMR est représentée dans la figure 4.6 pour différentes périodes d'échantillonnage. L'ensemble des expérimentateurs n'avait pas activé les mêmes capteurs et certaines données sont par nature plus discriminantes que d'autre lorsqu'elles sont disponibles : par exemple, le capteur bluetooth ou bien les identifiants des antennes GSM. Cependant, ces données ne sont pas toujours disponibles, en particulier car les utilisateurs souhaitent masquer ces informations aux autre applications du téléphone mobile ou bien pour réaliser une économie d'énergie.

Afin de pouvoir comparer et valider la pertinence de notre base de données, nous avons aussi évalué nos travaux sur une base de données publique et largement utilisé dans la littérature. Les paramètres et les utilisateurs sélectionnés dans la base de données du MIT pour conduire ces expérimentations ont était choisis en se référent au protocole détaillé dans la référence [76]. L'expérimentation a sélectionné 76 participants qui possède le plus grand nombre d'entrées dont l'activité a été enregistrée durant la période du 24/10/2004 au 20/11/2004.

Les informations sélectionnées sont :
— appels téléphoniques : numéro appelé,
— SMS : numéro appelé,
— bluetooth : adresse mac,
— localisation : identifiant de la cellule GSM.

Les courbes de FMR obtenues sont exposés dans la figure 4.7. Le niveau de confiance

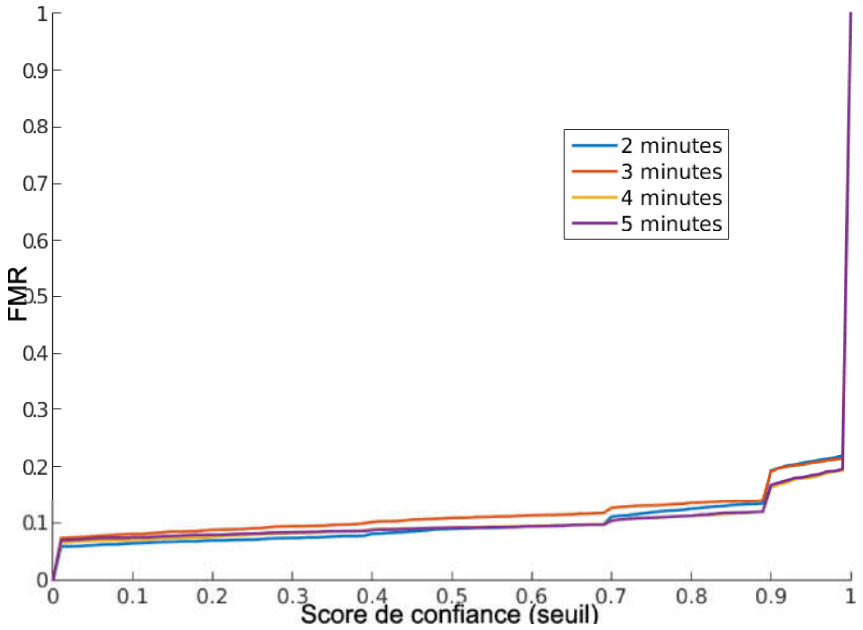

FIGURE 4.6 – FMR pour notre jeux de données

requis pour accéder à une application (qui peut être vu comme un seuil) dépends du temps d'intrusion acceptable pour un utilisateur. Nous exposons cette détection dans la section suivante

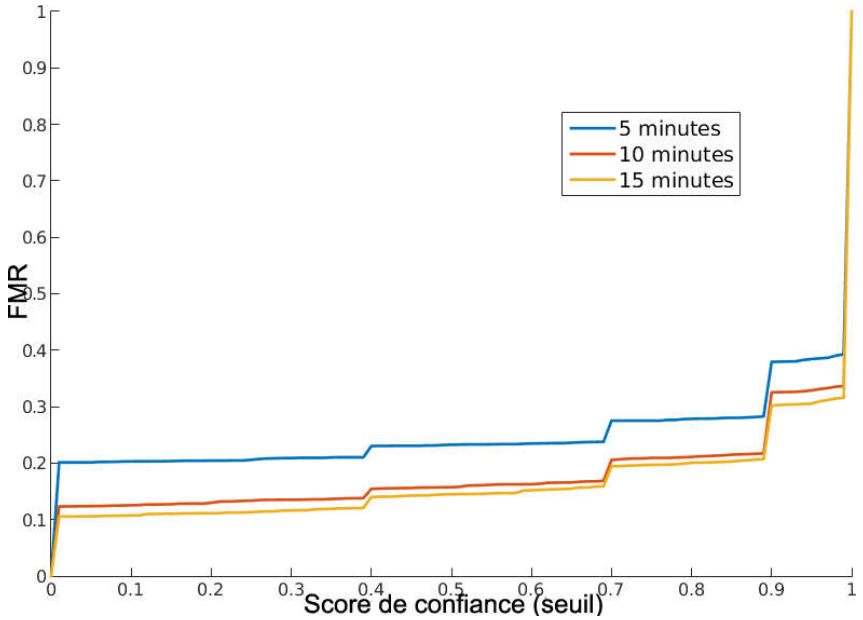

FIGURE 4.7 – FMR pour la base de données du MIT

TABLE 4.2 – Taux de détection et nombres d'actions explicites pour différents intervalles de temps (Collecte mobile)

Score de confiance	2min		3min		4min		5min	
	Taux	Actions	Taux	Actions	Taux	Actions	Taux	Actions
100%	100%	4.4	100%	1.0	100%	1.7	100%	2.0
90%	100%	5.7	100%	4.2	100%	5.8	100%	8.9
75%	100%	6.2	99%	4.4	99%	6.3	99%	9.3
50%	100%	7.8	99%	6.2	99%	10.4	99%	14.3
25%	100%	9.0	98%	8.2	98%	12.5	98%	16.9
10%	100%	9.2	99%	8.4	98%	12.8	98%	17.6

TABLE 4.3 – Taux de détection et nombres d'actions explicites pour différents intervalles de temps (base du MIT)

Score de confiance	5min		10min		15min	
	Taux	Actions	Taux	Actions	Taux	Actions
100%	100%	1.77	100%	0.2	100%	0.3
90%	89%	11.9	94%	1.4	90%	1.3
75%	89%	12.6	94%	1.4	90%	1.4
50%	88%	18.9	94%	1.8	90%	1.7
25%	86%	22.3	92%	2.1	90%	2.1
10%	85%	23.2	92%	2.1	90%	2.1

Détection des intrusions

Afin d'évaluer la résistance aux impostures, les données de l'utilisateur légitime sont substituées au milieu de la semaine par les données d'un imposteur. Nous supposons que l'imposteur possède le sel secret combiné aux données avant d'appliquer la fonction de hachage. Dans le cas contraire, le problème se résume à trouver des collisions à une fonction de hachage et garantie donc qu'il ne peut y avoir d'imposture.

Un exemple du protocole de test est proposé dans la figure 4.8. Bien sûr, la rapidité de la détection des imposteurs est directement corrélée à la période d'échantillonnage t. Cette approche est similaire à celle utilisée dans [82].

Nous observons logiquement que le nombre d'actions nécessaire à la détection augmente avec l'augmentation de t. Deux tables sont proposées, la table 4.2 compte l'ensemble des actions enregistrées par le framework. Ceci inclut aussi les changements de lieu. La table 4.2 ne compte que les actions réalisées explicitement par l'utilisateur. Cette seconde table permet de montrer le nombre d'actions effectives que réaliserait un imposteur.

Les mêmes résultats sont obtenus avec la base du MIT dans les tables 4.3 et 4.3.

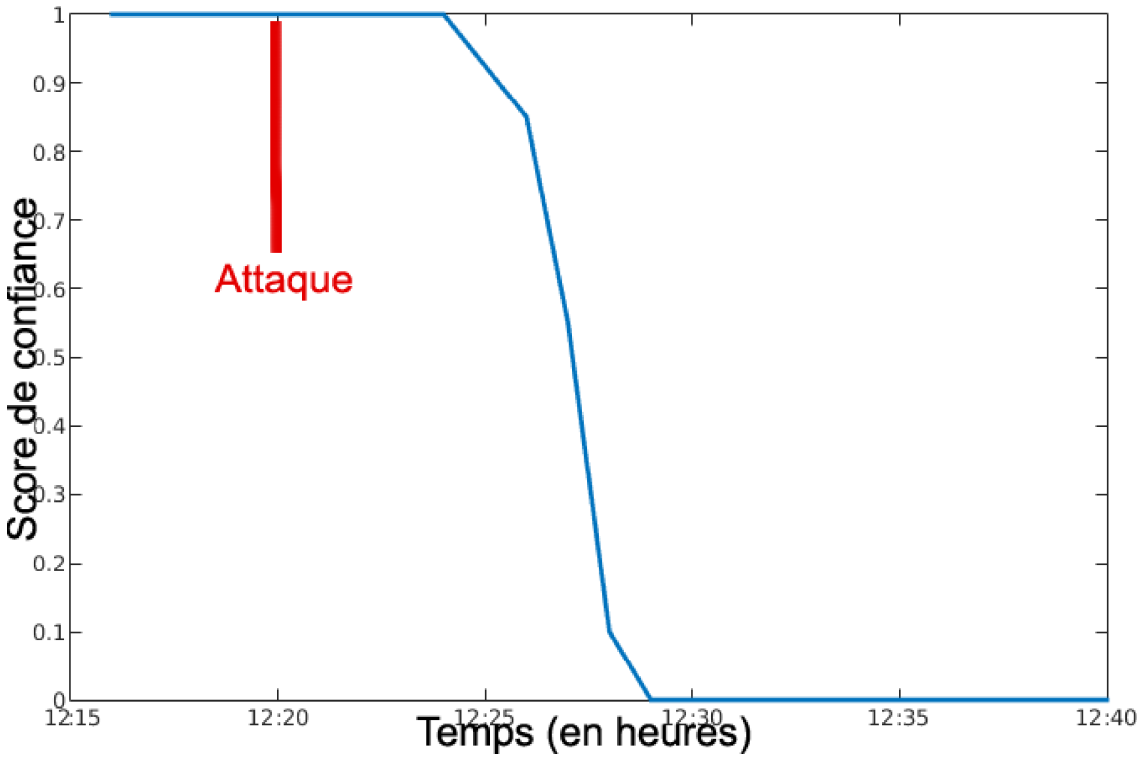

FIGURE 4.8 – Détection des attaques par vol du mobile

Évolution du comportement

Afin d'évaluer la robustesse d'un tel framework au cours du temps, nous avons sélectionné les 5 utilisateurs qui présentent une période de collecte suffisamment longue. Ceci correspond à un intervalle allant de 10 mois jusqu'à 1 ans. Nous avons alors procédé à un enrôlement de deux semaines. Dans tous les cas, nous observons une décroissance nette du score de confiance moyen au cours du temps. Cette décroissance apparaît entre 1 mois et 4 mois après le premiers enrôlement. Un exemple est disponible pour l'utilisateur 25 dans la figure 4.9.

Ce changement est dû à un changement des habitudes de l'utilisateur. Il est intéressant de remarquer que le fait d'augmenter le temps d'enrôlement n'a pas modifié l'ampleur ni reculé le début de cette dégradation des performances biométriques. Cependant, un simple ré-enrôlement du profil de l'utilisateur à un intervalle régulier permet de palier à ce problème (par exemple, tous les mois). La figure 4.10 expose un ré-enrôlement périodique pour le même utilisateur 25. Une amélioration significative est observable.

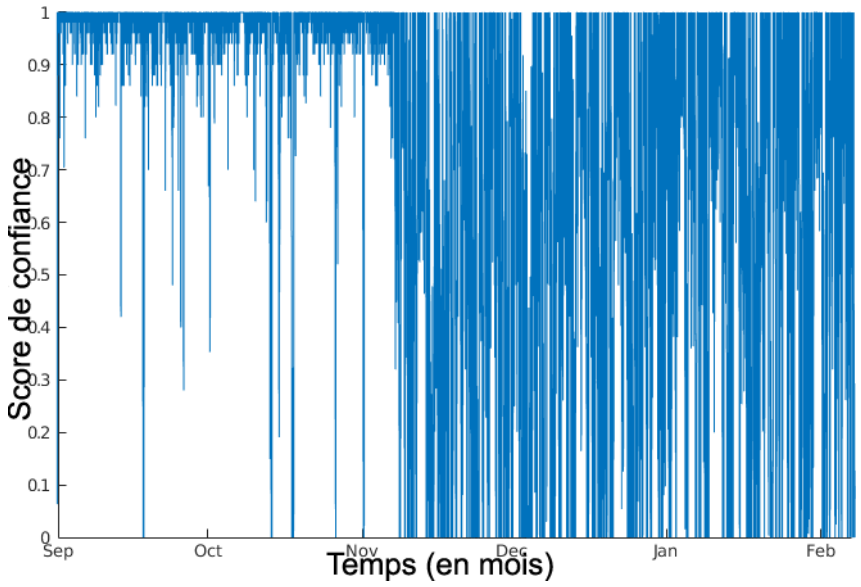

FIGURE 4.9 – Évolution du score de confiance pour l'utilisateur 25 de la base de données MIT sans réentrainement

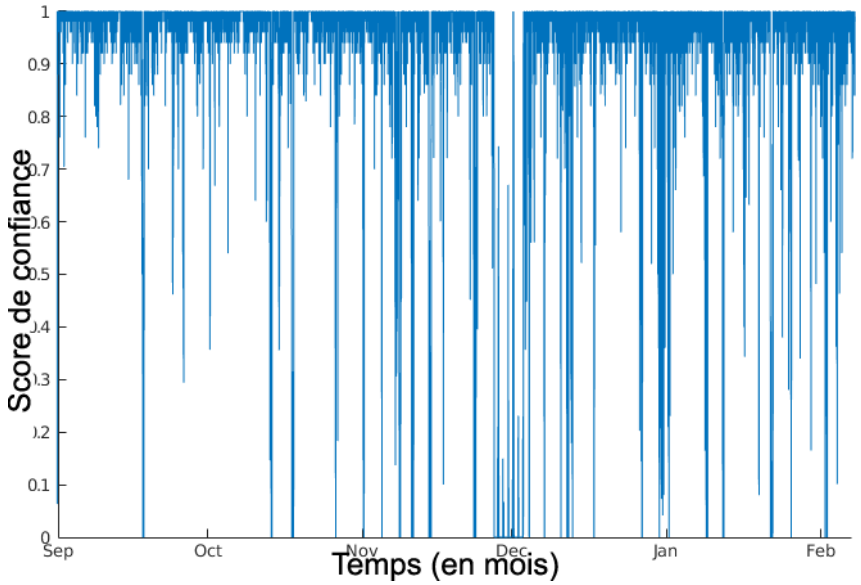

FIGURE 4.10 – Évolution du score de confiance pour l'utilisateur 25 de la base de données MIT sans réentrainement périodique

4.5.3 Méthode 2 basée sur la cryptographie homomorphe

La solution que nous avons élaborée et qui se base sur du chiffrement homomorphe nécessite deux points d'attentions particuliers :

— Le temps de calculs

— La taille du template

En effet, afin de réaliser une authentification comportementale, les templates sont généralement assez conséquents. Le temps de calcul et la taille des chiffrés nécessitent de réduire au maximum ces templates. Pour ces raisons, nous avons divisés en deux parties l'évaluation. La première partie évalue la qualité de l'authentification biométrique et la seconde évalue les performances de l'algorithme une fois porté sur mobile.

La mesure des performances a été réalisée avec la base de données CRA. Nous proposons aussi une évaluation biométrique avec la base de données du MIT. Cette évaluation est cependant à prendre avec précaution puisque les numéros de téléphone ont été remplacés par des pseudonymes dans la base de données du MIT. Lors de cette substitution, ils ont été attribués dans un ordre croissant ce qui améliore les résultats biométriques de la base. Les résultats sont résumés dans la table 4.4 en fonction de la taille de template utilisé.

Evaluation biométrique

TABLE 4.4 – Performance biométrique de la solution

Taille du template	EER	
	CRA	MIT
16	14.64%	13.36%
32	13.03%	12.21%
64	11.41%	10.78%
128	9.82%	9.04%

On peut remarquer que les résultats sont similaires à [112], [76], ou encore à ceux obtenus avec le biohashing dans le cas d'un vol de mobile [56]. De plus, la vérification est réalisée sur le serveur sans qu'il n'y ait de possibilité d'une fuite de données concernant le template des utilisateurs ou les éléments d'intérêt.

Performances de calcul

Le protocole est composé d'un client (le téléphone mobile) et d'un serveur d'authentification. Les ressources disponibles sont limitées sur le téléphone mobile. Nous avons évalué le protocole proposé sur un Galaxy S7 (ARM Cortex-A72 - 2.3 GHz avec 4Go de RAM) pour le téléphone mobile et sur un Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz avec 20Go de RAM pour le serveur. Une fois un appel téléphonique passé par l'utilisateur, la localisation et le numéro appelé sont extraits et chiffrés. Afin d'accélérer les calculs, deux tampons de 0 et de 1 chiffrés sont pré-calculés. Ce

pré-calcul prend environ 300ms pour des clés de 2048 bits et peut être réalisé quand le téléphone est dans un état d'attente ou en veille.

Cela demande un cache de 50ko dans le cas de clé de taille 2048 bits. C'est une taille relativement faible comparée aux ressources actuellement utilisées par les smartphones pour chacune des applications. La problématique liée à la puissance de calcul apparait au moment du déchiffrement du côté du téléphone mobile.

Cependant, les performances que nous avons observées permettent d'imaginer des applications industrielles basées sur ces recherches.

Il est intéressant de noter que dans le cas d'une authentification comportementale basée sur des éléments du contexte mobile, la taille des données envoyées reste fixe et relativement faible dans notre protocole. La plus grosse partie des données étant les templates à déchiffrer sont téléchargées par le téléphone. C'est un avantage du protocole car, les vitesses d'envoi sont généralement inférieures aux vitesses de réception. De plus, la taille des données à télécharger est comparable à un usage normal d'internet sur smarphone (la taille d'une image). Les résultats pour différentes tailles de clés et différentes tailles de templates sont affichés dans les tables 4.5 et 4.6.

TABLE 4.5 – Calculs sur le smartphone avec une clé de 1024 bits

Nombre de templates	Temps de chiffrement	Temps de déchiffrement	Taille de l'upload	Taille du download
16	<1ms	306ms	12.5Ko	200Ko
32	<1ms	580ms	12.5Ko	400Ko
64	<1ms	952ms	12.5Ko	800Ko
128	<1ms	2,147s	12.5Ko	1.563Mo

TABLE 4.6 – Calculs sur le smartphone avec une clé de 2048 bits

Nombre de templates	Temps de chiffrement	Temps de déchiffrement	Taille de l'upload	Taille du download
16	<1ms	1.324s	25Ko	400Ko
32	<1ms	2.525s	25ko	800Ko
64	<1ms	4.280s	25ko	1.563Mo
128	<1ms	8.351s	25ko	3.125Mo

Du côté serveur, des vecteurs aléatoires sont calculés pour chacune des entrées du template en utilisant la clé publique du client. De nouveau, ces données peuvent être précalculées afin d'accélérer le processus d'authentification. Le serveur n'a alors plus qu'à calculer une multiplication sur chacun des templates avec les données de l'utilisateur. Les performances et les ressources mémoires nécessaires pour le serveur sont exposées dans la table 4.7.

TABLE 4.7 – Calculs sur le serveur

Nombre de templates	Temps de XOR		Tailles des templates	
	1024 bits	2048 bits	1024 bits	2048 bits
16	1ms	1ms	256Ko	512Ko
32	1ms	2ms	512Ko	1Mo
64	2ms	4ms	1Mo	2Mo
128	3ms	5ms	2Mo	4Mo

La partie la plus coûteuse de ce protocole est le déchiffrement mobile. Cependant, même avec une clé de 2048 bits, et un template de 128 entrées, une authentification peut être réalisée en moins de 10 secondes.

4.5.4 Méthode 3 basée sur le biohashing

Afin d'évaluer un framework qui utiliserait le biohashing comme méthode de protection de la vie privée, nous utilisons la base de données recensant les habitudes d'appels des utilisateurs. Pour évaluer les performances biométriques de notre application, nous utilisons ici le FMR et le FNMR. Afin d'évaluer la protection de la vie privée, nous utilisons une métrique basée sur la mesure d'entropie de Shannon[106].

Un biocode parfait n'est pas sensé divulguer d'informations quant au contenu des éléments qu'il contient. Autrement dit, aucune information sur les éléments d'intérêts ne peut être obtenue à partir d'un biocode.

Ceci signifie qu'il n'est pas distinguable d'un aléa et que par conséquent son entropie doit être égale à sa longueur en bits. Cependant, nous observons que l'entropie réelle mesurée est inférieure à l'entropie théorique parfaite. Nous appellons cette différence le *privacy leakage*. Cette valeur est exprimée en bits. Afin de caractériser le biohashing sur les données issues du contexte mobile, deux classifieurs sont utilisés. Toujours dans le but d'être en cohérence avec les applications industrielles, nous utilisons des méthodes de classifications ne nécessitant de posséder que les données d'un seul utilisateur, c'est à dire pas de classifieur multi-classes. Le premier classifieur utilisé est un One Class SVM. L'implémentation utilise la librairie libSVM [20] pour Matlab.

Nous évaluons tout d'abords les performances de notre base de données sans appliquer l'algorithme de biohashing. Ceci a pour but de comparer nos résultats avec ceux de la littérature.

Validation de la base de données

Lors de ces calculs ; nous utilisons les données issues de la base CRA. Nous utilisons le numéro de téléphone de l'abonné comme un identifiant et nous concaténons les données du numéro appelé avec la position GPS de l'antenne GMS depuis laquelle est passé l'appel téléphonique ou depuis laquelle est envoyé le SMS. En utilisant un SVM à une classe, nous obtenons les résultats décrits dans les tables 4.8 et4.9.

TABLE 4.8 – SVM à une classe sans protection de la vie privée

FMR (%)	FNMR (%)
29.54	1.23

TABLE 4.9 – KNN sans protection de la vie privée

Nombres de voisins	EER (%)	Seuil correspondant
1	8.39	0.15
2	8.39	0.30
3	9.15	0.49
4	9.28	0.67
5	9.77	0.86

Les résultats sont nettement améliorés en utilisant la distance aux plus proches voisins. Avec cette distance, nous obtenons des résultats similaires à ceux obtenus par [77]. Une légère différence peut être remarquée, elle provient :

— 1) d'un biais dans la base du MIT utilisée dans [77] concernant l'attribution des numéro de téléphone

— 2) de l'utilisation d'une seule classe contre un classifieur multi-classe.

Ces résultats nous permettent d'utiliser sereinement la base de données CRA afin de poursuivre les expérimentations.

Scénario du meilleur cas

Contrairement aux autres solutions permettant de préserver l'anonymat, le Biohashing par son algorithme conserve une partie des propriétés biométriques. Il est donc possible d'obtenir avec un biocode des faux positifs et des vrais négatifs. Autrement dit, le FMR et le FNMR ne sont pas égaux à zéro même si les utilisateurs ne possèdent pas la même clé. Lorsque nous utilisons le Biohashing sur des données de contexte, nous définissons deux cas de tests :

— **Le meilleurs cas :** Dans ce scénario, les attaquants ne possèdent pas la clé secrète.

— **Le pire cas** : Dans ce scénario, l'attaquant a volé la clé secrète. Puisque la clé doit être protégée de manière architecturale, cela signifie que le téléphone a été dérobé.

Afin d'obtenir un biocode de 100 bits, nous concaténons une représentation binaire de la latitude et de la longitude, auquel est concaténée une représentation binaire du numéro de téléphone mobile appelé. Ceci nous donne un vecteur de 104 bits qui une fois transformé via l'algorithme de Biohashing, nous obtenons un Biocode de 100 bits. Le résultat obtenu en classifiant les données à l'aide d'un SVM à une seule classe sont exposés dans la table 4.10. La encore, l'utilisation d'une simple distance au voisins les plus proches améliore grandement les résultats. On peut observer ces résultats dans la table 4.11.

TABLE 4.10 – SVM à une classe dans le meilleur des cas

FMR (%)	FNMR (%)
35.19	0

TABLE 4.11 – KNN dans le meilleur des cas

Nombre de voisins	EER (%)	seuil correspondant
1	1.04	0.30
2	1.10	0.62
3	1.09	0.95
4	1.16	1.29
5	1.19	1.63

Afin d'évaluer la protection de la vie privée avec l'algorithme de Biohashing sur des informations issues du contexte mobile, nous mesurons la fuite de données sensibles (privacy leakage). Lorsque nous mesurons la différence entre un biocode idéal et les biocodes obtenus, nous obtenons un privacy leakage de 4 bits. Même si cela représente une bonne avancée en terme de protection des éléments d'intérêts, il faut garder à l'esprit que ce n'est pas une protection parfaite.

Pire des cas

Dans le pire des cas, l'utilisateur possède le téléphone mobile et peut donc générer des données avec la clé de l'utilisateur. Les performances sont alors nettement dégradées. Les résultats en utilisant un SVM à une classe et la distance aux plus proches voisins sont représentés respectivement dans les table 4.12 et table 4.13.

TABLE 4.12 – SVM à une classe dans le pire des cas

FMR (%)	FNMR (%)
34.60	2.68

TABLE 4.13 – KNN dans le pire des cas

Nombre de voisins	EER (%)	Seuil correspondant
1	10.45	0.23
2	10.16	0.47
3	10.65	0.72
4	10.69	0.98
5	10.76	1.24

4.5.5 Discussion

Chacune des méthodes proposées dans ce manuscrit possède des atouts qui lui sont propres. Il est néanmoins nécessaire de réaliser des compromis afin de choisir les méthodes répondant aux contraintes techniques de notre projet.

La méthode naïve d'utiliser des hash offre l'avantage de pouvoir être continue et de ne pas révéler d'informations autres que des fréquences d'utilisation. Cependant, face à un adversaire avancé, la simple fréquence permettra de déduire les données émises. A titre d'exemple, dans [63] les auteurs exposent que la position géographique la plus probable est le lieu de domiciliation. Les performances de cette méthode reste relativement faible et la combiner avec d'autres solutions augmenterait les informations disponibles pour un attaquant.

L'algorithme de Biohashing est une solution intéressante car elle permet un calcul rapide et une transformation non réversible de la donnée initiale. Cependant, les données disponibles en entrée de l'algorithme ne sont pas suffisantes et des transformations sont nécessaires. Après cette transformation, nous obtenons 5% de privacy leakage ce qui révèle qu'une partie de l'information liée à la donnée initiale est présente dans les Biocode. Cependant, les résultats biométriques sont satisfaisants offrant jusqu'à 1% d'EER dans le meilleur des cas et 8% dans le cas d'un vol du mobile. Dans le cas d'un vol du mobile et du template, il est toujours impossible de récupérer la donnée initiale grâce aux propriétés de diffusion de l'algorithme. De plus, les capacités de calcul nécessaires à l'utilisation de cet algorithme sont très modestes ce qui est particulièrement adapté au calcul sur téléphone mobile.

Le chiffrement homorphe est incontestablement le champion des performances biométriques. Il offre un EER égal à 0% dans le meilleur des cas et est aussi performant que la donnée biométrique seule en cas de vol du téléphone mobile. Cependant, s'il protège parfaitement la donnée vis à vis de la vie privée, en cas de vol du template et

de la clé, il devient possible de récupérer les données d'origine. La principale difficulté est lié à la complexité de l'algorithme qui demande des moyens importants et par conséquent est mois performant sur téléphone mobile.

Enfin, une dernière alternative peut être d'utiliser le Biohashing et le chiffrement homomorphe. Dans ce cas, nous obtenons les même performances biométriques et la même résistance au vol que pour le Biohashing avec la même protection des informations personnelles et le même temps de calcul que pour le chiffrement homomorphe.

Les trois dernières solutions offrent des compromis intéressants d'un point de vue industriel. Afin de réaliser nos prototypes (voir section 6) nous avons choisie d'utiliser l'algorithme de Biohashing pour son temps d'exécution. Ce choix a été réalisé afin de favoriser l'usabilité. Dans d'autre scénario, il est envisageable d'utiliser d'autres méthodes.

La table 4.14 situe nos méthodes avec les autres méthodes présentes dans la littérature.

TABLE 4.14 – Comparaison avec les autres solutions

Référence	No. d'utilisateurs	Paramètres	Vie privée	Performances	Révocabilité
Li et al.[77]	71	Localisation & Appels	Aucun	EER :8.8% avec 1 appel et EER :5.3% avec 6 appels	✗
Savaenee et al.[101]	30	analyse linguistique, dynamique de frappe et analyse comportementale	Aucun	EER :3.3%.	✗
Tanviruzzaman et al.[112]	13	GPS et démarche	Aucun	EER :10%	✗
Fridman et al.[44]	200	GPS	Aucun	FAR :11% and FRR :6%	✗
Fridman et al.[44]	200	contenu des SMS, GPS, applications, historique web	Aucun	ERR :5% après 1 minute et EER :1% après 30 minutes	✗
Chow et al.[22]	50	appels, historique web, sms, gps	Donnée stocké sur le télphone	Les utilisateurs légitimes exécutes 90 actions avant d'être rejeté quand un imposteur n'en réalise que 10	✗
Safa et al.[103]	Non fourni	appels, localisation, Wi-Fi, historique web	protocole interactif en trois échanges entre client et serveur	Non fourni	✓
Méthode 1	100	appels, localisation	Hachage des données	EER :10.45%	✓
Méthode 2	100	appels, localisation	Cryptographie homomorphe	EER :0% dans le meilleurs cas, EER :8.5% dans le pire cas	✓
Méthode 3	100	appels, localisation	Biohashing	EER :1.04% dans le meilleurs cas, EER :10.45% dans le pire cas	✓

On remarque que les méthodes proposées dans ce manuscrit permettent d'améliorer grandement la prise en compte de la protection de la vie privée pour l'authentification transparente. De plus, ces méthodes sont applicables et ont été testées dans un environnement réaliste (calculs effectués sur téléphones mobiles).

Les figures suivantes permettent de comparer visuellement les atouts de chaque méthode. Quatre axes sont proposés :

— **Information :** La capacité à protéger les éléments d'intérêt.
— **Temps de calcul :** Le temps de calcul nécessaire dans un système complet (mobile et serveur).
— **Résistance au vol :** Capacité à protéger les données une fois la base de données et/ou la clé secrète divulguée.
— **Performance :** Les performances biométriques du système.

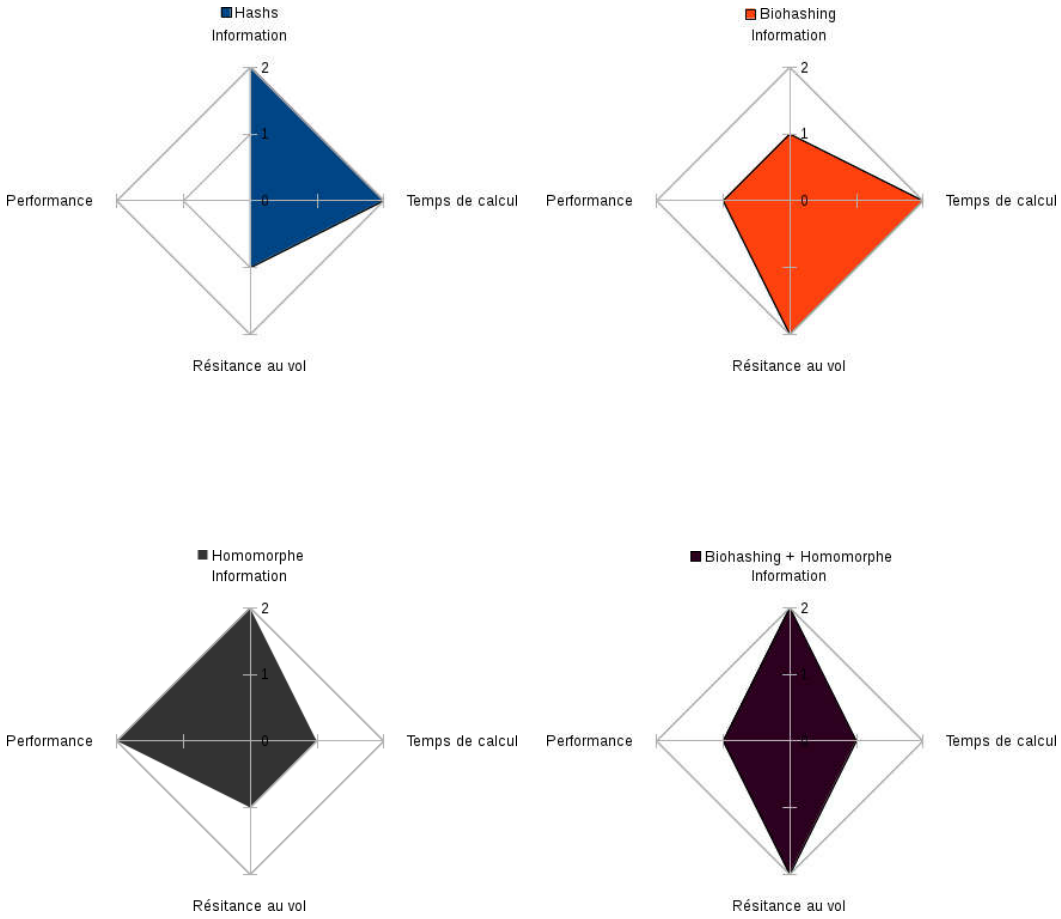

FIGURE 4.11 – Comparaison des méthodes de protection

4.6 Conclusion

Ce chapitre propose trois solutions d'authentification qui utilise des éléments du contexte de l'utilisateur. Afin de protéger les données à caractère personnel, les solutions proposées s'appliquent sur le mobile de l'utilisateur avant d'être transmis au serveur d'authentification.

Ce choix permet une évolution rapide des systèmes d'authentification en fonction des donnée reçues. En effet, l'architecture client serveur permet d'étendre le recueil des données à de multiples appareils assez simplement en gardant les même solution d'authentifications. Cette architecture permet aussi très simplement de combiner plusieurs facteurs d'authentification issus du même appareil.

Cependant, avant de pouvoir utiliser différents appareils et différentes modalités, nous verrons dans le chapitre suivant des solutions pour combiner les scores d'authentifications obtenus dans le but d'une application industrielle.

Chapitre 5

Etablir un seuil de confiance pour l'authentification

Ce chapitre étudie l'état de l'art de l'établissement des niveaux de confiances dans les systèmes d'authentification. Une fois les normes et les articles de la littérature présentés, nous proposons une méthode permettant à la fois de combiner différente méthodes d'authentification et d'analyser les méthodes proposées dans la littérature.

Sommaire

5.1 Introduction

La généralisation des services web a entraîné la généralisation des mots de passe. Avec la croissance des transactions numériques, les utilisateurs sont ainsi sollicités périodiquement afin de vérifier leur identité.

La méthode d'authentification idéale devrait être sécurisée, et non intrusive et permettre différents niveaux de sécurité. Avec la multiplication des services, il est nécessaire de pouvoir adapter le niveau de sécurité au service proposé et aux risques

associés à l'usage de ce service. Cependant, l'assurance dans l'identité de l'utilisateur ne devrait pas ajouter de charge supplémentaire à l'utilisateur.

Une solution pour éviter une re-authentification constante est d'utiliser un service d'authentification unique (Single Sign On) comme *Google Sign On* ou *Facebook connect*. Dans les environement SSO, la vérification d'identité n'est pas réalisée directement par le fournisseur de service. Un fournisseur d'identité (IdP) gère à la fois l'enrôlement et l'authentification pour le fournisseur de services (SP) afin d'assurer un niveau de confiance suffisamment élevé.

Dans un environnement SSO, la vérification de l'identité n'est pas directement réalisée par le fournisseur de service. Un fournisseur d'identité (IdP) gère les étapes d'enrôlement et d'authentification pour le fournisseur de services(SP). Pour avoir suffisamment confiance dans l'identité numérique de son client, le fournisseur de service demande un certain niveau d'assurance au fournisseur d'identité. Le choix de ce niveau d'assurance est réalisé en fonction des risques qui sont associés aux services proposés par le fournisseur de services.

Le problème actuel qui réside dans cette forme d'authentification, provient du fait que la confiance repose sur des règles pré-établies qui correspondent à un niveau de sécurité maximum.

L'authentification devrait être définie en prenant en compte un niveau de confiance et les besoins actuels d'un mécanisme d'authentification continue.

5.2 Estimateurs de la confiance

LA notion de confiance dans l'identité numérique, a été abordée sous plusieurs aspects dans la littérature. La norme veut que l'on utilise les textes gouvernementaux (normés dans l'ISO 29115). Il existe cependant d'autres approches afin d'évaluer le niveau de confiance dans l'identité d'un individu.

5.2.1 Les niveaux d'assurance

Les niveaux d'assurance (AAL) sont à mettre en relation avec le niveau de risque en cas d'une authentification frauduleuse. Plusieurs gouvernements ont établi des exigences dans le but de confronter les risques par rapport à l'assurance sur l'authentification. Les gouvernements utilisent des structures d'authentification (frameworks) pour les services publics basées sur des niveaux d'assurance [64].

Jøsang [64] listent les frameworks suivants [1] :

— **US EAG** *Electronic Authentication Guideline* (NIST SP800-63-1) [115] : Décrit les exigences pour les niveaux d'assurance pour accéder aux services des agences fédérales des États Unis.

— **EU IDABC** *eID Interoperability for PEGS* (Pan-European eGovernment services) [41] : une proposition pour une authentification multi-niveaux et une cartographie des mécanismes d'authentification existants.

— **Australian NeAF** *National e-Authentication Framework* [4] : Une structure pour l'authentification bien agencée, publiée par le gouvernement fédéral Australien. Elle inclut de manière explicite un niveau d'assurance 0, dont le but est de fournir un accès et une authentification anonyme.

— **Indian ePramaan** *ePramaan : Framework for e-Authentication* [51] : Un modèle concis publié par le gouvernement Indien. Il inclut lui aussi un niveau d'assurance 0 similaire à l'Australian NeAF.

Framework	Niveau d'assurance				
EAG (USA) 2006	peu ou pas d'assurance (1)		quelque (2)	élevé (3)	très élevé (4)
IDABC (EU) 2007		minimum (1)	faible (2)	substantiel (3)	élevé (4)
NeAF (Australie) 2009	aucun (0)	minimal (1)	faible (2)	modéré (3)	élevé (4)
ePramaan (Inde) 2012	aucun (0)	minimal (1)	modéré (2)	fort (3)	très fort (4)

TABLE 5.1 – Correspondance entre les niveaux d'assurances

Les recommandations émises par le NIST ont été normalisées dans [61]. Les différents niveaux d'assurance ont une construction analogue. Nous retrouvons d'ailleurs une similitude dans le degré d'assurance correspondant au niveau qui lui est associé. Par exemple, le niveau d'assurance 3 (AAL-3) a une signification très proche dans chacun des frameworks.

Jøsang [64] décrit les trois facteurs suivants comme contribuant à la robustesse d'un système d'authentification pris dans son ensemble :

Force de la méthode d'authentification : la robustesse intrinsèque de la solution d'authentification utilisée par la solution. Cell-ci est donnée par la force

1. Josang cite aussi le framework Norvégien. Celui ci étant écris en Norvégien, nous n'avons pas été en mesure de l'examiner

des facteurs d'authentification et de leurs combinaisons.

Gestion des revendications d'authentification : la fiabilité estimée des revendications utilisées pour l'authentification. Par exemple, un mot de passe changé fréquemment est plus fiable que le même mot de passe qui n'est pas changé de manière régulière.

Assurance dans l'enregistrement des identités : la précision avec laquelle les attributs d'identité sont enregistrés.

Les niveaux d'assurance ont l'avantage de proposer une solution pratique pour déterminer la force de l'authentification en fonction des facteurs d'authentification, des politiques de gestion en place pour ces facteurs d'authentification et des attributs d'identité connus du système d'authentification.

Récemment, le NIST a proposé une nouvelle version du framework dans [1]. Les niveaux d'assurances y sont réduits à trois sans y apporter cependant de changements majeurs. Il est toutefois important de noter la dégradation des mots de passe à usage unique par SMS dans ce document.

Cependant, pour atteindre un niveau d'assurance donné, le système requiert qu'un certains nombre d'exigences soit satisfait. Ces exigences demandent des contraintes techniques fortes et ne permettent pas de comparer les méthodes d'authentification entre elles. Les niveaux d'assurance ne permettent donc pas de décider parmi plusieurs agencements des facteurs d'authentification lequel est le plus adapté en fonction de contraintes telles que la protection de la vie privée, la sécurité et l'usage. De plus, le manque de granularité dans les niveaux d'assurances, au maximum cinq niveaux, ne permet pas d'ajuster finement le processus d'authentification à la tâche en cours.

5.2.2 Notion de confiance dans l'identité

D'autres approches existent dans l'étude de l'authentification. La confiance dans l'utilisateur peut être attribuée en fonction des actions réalisées par un individu. Si une action effectuée est profitable au système alors le niveau de confiance augmente. Dans le cadre de l'authentification, une action usuelle serait une action profitable au système. Par exemple, dans [76], un numéro de téléphone connu est considéré comme une action profitable au système.

A notre connaissance, les travaux de Marsh [78] sont considérés comme étant les premiers (1994) à formaliser numériquement la confiance. Marsh évalue la confiance

de manière continue dans l'intervalle $[-1, +1]$ à partir d'un ensemble de variables. Il évalue +1 comme étant une confiance aveugle, et -1 comme étant une méfiance totale.

Ranganathan *et al.* [94] proposent d'évaluer la confiance dans l'authentification comme étant la confiance dans les éléments contextuels. Les auteurs proposent que si un utilisateur essaie de s'authentifier en utilisant n méthodes d'authentification différentes et que tous les essais sont réussis, alors $V_1, V_2, ..., V_n$ sont les valeurs de confiance associées. La confiance issue du réseau des méthodes d'authentification V_{net} est donnée par la formule (dérivée de la théorie des probabilité)

$V_{net} = 1 - (1 - V_1)(1 - V_2)...(1 - V_n)$ Où V_i sont des probabilités.

Cette approche a un avantage conséquent qui est la possibilité de combiner aisément des méthodes d'authentification entre elles, et ce qu'elles soient implicites ou explicites. La seule problématique est de pouvoir calculer une valeur de confiance pour chacune des méthodes d'authentification. Cette valeur peut être assez triviale dans le cas de l'utilisation de la biométrie puisqu'il peut alors s'agir du nombre de faux positifs (EER ou FMR). Elle l'est beaucoup moins pour des authentifications basées sur les autres facteurs. De plus, la formule suppose que toutes les authentifications soient réussies, qu'advient-il si une des authentifications échoue ?

Le framework NICA [24] (No Intrusive Continuous Authentication) propose une évaluation de la confiance dans l'authentification de l'utilisateur allant de -5 à +5 (voir figure 5.1).

Cette vision de la confiance dans le cadre de l'authentification continue permet de valider l'accès à une ressource. Ainsi, le niveau de confiance dans l'authentification requis pour utiliser un service d'écoute de musique en ligne n'est pas le même que celui requis pour payer l'abonnement à ce service.

Afin de calculer la valeur à incrémenter, ou à décrémenter, les facteurs biométriques ainsi que des facteurs mémoriels sont pris en compte. Les facteurs d'authentification utilisés sont alors liés à un score en fonction de leur robustesse. Enfin, le code PIN permet de remettre le score du mécanisme d'authentification à 0 lorsque que la session a été verrouillée.

Il peut être reproché à ce système de n'être qu'une fusion élaborée des facteurs

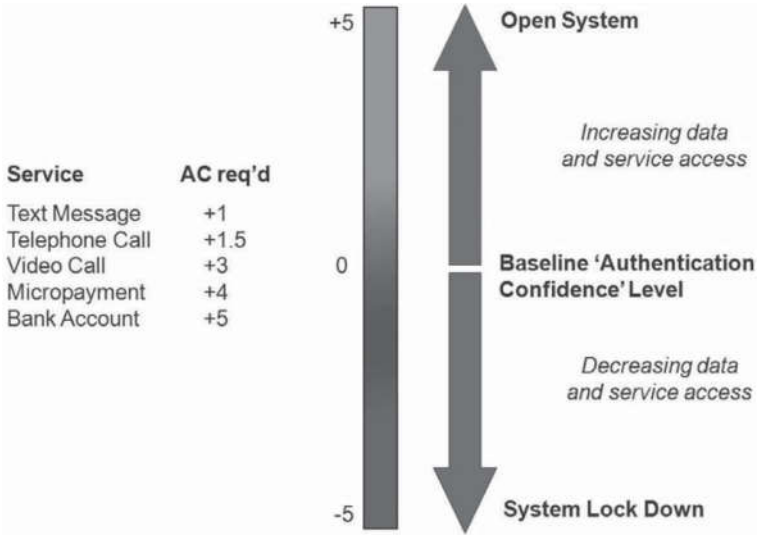

FIGURE 5.1 – Niveau de confiance dans l'authentification de NICA [24]

biométriques. Dans le cas d'une authentification en continue, cela permet aussi de ne pas exclure un utilisateur directement après qu'une authentification biométrique a échoué. Dans le cas d'un ERR de 10%, cela arrive en moyenne une fois sur dix, et par conséquent, nuit à l'usabilité du système d'authentification continue. Le terme *authentification périodique* est d'ailleurs plus adapté pour décrire ce type de système.

Crawford *et al.* [26] proposent de combiner des systèmes d'authentification biométriques et des éléments issus du contexte avec des authentifications explicites issues d'un facteur mémoriel. Pour cela, l'auteur utilise un tampon pour chaque mécanisme d'authentification dans lequel l'auteur stocke les résultats des authentifications précédentes. L'auteur calcule alors la moyenne pour chacun de ces tampons et réalise une décision en choisissant que la confiance soit le maximum parmi toutes les moyennes ainsi générées. Ceci privilégie l'usabilité.

Malheureusement, si un utilisateur se fait voler un de ses moyens d'authentification, qu'il soit biométrique ou mémoriel alors l'ensemble du système est compromis. De plus, ces deux frameworks sont destinés à vérouiller le téléphone mobile et non l'accès à des services connectés. Ceci règle les problèmes de sécurité en cas de perte ou de vol du téléphone mobile mais ne permet pas de simplifier l'usage auprès des applications et services connectés d'aujourd'hui. Pour cela, il est nécessaire qu'un processus d'authentification permette de propager la confiance entre les différentes parties.

Une solution alternative proposée par Chow *et al.* [22] est d'utiliser des données contextuelles tel que la position GPS ou l'historique de navigation, afin de s'authentifier auprès d'un serveur permettant l'authentification à l'aide du protocole OpenID Connect. Dans cette architecture, afin de faciliter l'usabilité, si une authentification, grâce aux données contextuelles échoue, alors le système bascule sur un mécanisme d'authentification classique. En terme de protection de la vie privée, les données sont stockées sur un serveur d'authentification implicite.

Mustafić *et al.* [83] évoquent un système d'authentification continue utilisé en plus d'une authentification explicite dans le cadre d'une installation domotique. Ce système d'authentification continue utilise la dynamique de frappe au clavier. Cependant, la confiance accordée à l'utilisateur est directement proportionnelle au résultat de l'authentification par dynamique de frappe au clavier. Cette solution est relativement contraignante pour l'utilisateur, l'obligeant à se reconnecter régulièrement, tout du moins aussi souvent que la modalité biométrique comportementale choisie pour l'authentification continue a de vrais négatifs (EER ou FNMR).

Une autre notion importante est l'évolution de la confiance au cours du temps. Clarke [24] considère que les actions usuelles ou non usuelles s'effectuent en continu. Ceci permet dont d'évaluer la confiance en temps réel. Cependant, cette hypothèse ne s'applique pas aux services connectés qui peuvent envoyer du contenu suite à une authentification correcte et attendre un certains temps pour faire de nouveau un lien avec l'utilisateur. Il est nécessaire pour cela d'évaluer la confiance au cours du temps. Crawford *et al.* [26] appliquent un poids proportionnel à l'ancienneté de chacun des scores d'authentification. Ainsi, une preuve d'authentification récente a plus de valeur qu'une preuve plus ancienne.

Dans les solutions commerciales, la société Biometry [2] est la plus proche de proposer une solution d'authentification se basant sur la mesure de la confiance. Elle offre différents frameworks pour agencer des facteurs d'authentification biométriques afin d'en déduire un niveau de confiance dans le temps. Ce niveau de confiance est apparenté à un système de *bons points* qui sont attribués lorsqu'un utilisateur s'authentifie correctement. Plus l'authentification réalisée est forte, plus l'utilisateur gagne de points. Une érosion de la confiance au cours du temps est appliquée par la suite. Enfin, une notion de seuil minimal est introduite. Si on rapporte cela à une authentification multi-facteurs, on peut voir la possession du mobile comme

2. www.biometry.com/

un facteur d'authentification apportant un score minimum, ensuite en fonction des authentifications réussies, il est nécessaire de fournir d'avantage de points pour l'établissement de la confiance dans l'authentification de l'utilisateur.

Dans la référence [58], les auteurs décrivent 6 niveaux d'assurances en utilisant l'entropie ainsi qu'une entropie biométrique équivalente définie dans [89]. L'entropie est calculée en considérant différents vecteurs d'attaque comme un mot de passe facile à deviner. Ces niveaux sont décrits dans la table 5.2. Dans ce framework de comparaison, la règle est de combiner les facteurs en additionnant leur entropie.

TABLE 5.2 – [58] Levels of assurances

Entropy	Level	ISO equivalent level
$128 < H$	Extreme security	LoA4+
$56 < H < 128$	High security	LoA4
$20 < H < 56$	Higher medium security	LoA3
$14 < H < 20$	Medium security	LoA2
$10 < H < 14$	Low security	LoA1
$H < 10$	No security	

Dans [92], les auteurs proposent de regrouper toutes les informations qui peuvent aider pour authentifier un utilisateur. Ensuite, l'idée principale est de laisser un être humain décider de l'authentification en fonction des faisceaux d'indices qui ont été collectés. Ce cas s'applique à des scénarios de haute sécurité.

Pour des raisons de coût et de temps, ce système ne peut évidemment pas être adapté à tous les systèmes d'authentification. Pour la majorité des services web, les utilisateurs doivent être authentifiés de manière massive et immédiate.

Afin de palier à ce manque de granularité et de prendre le contexte en compte, nous proposons des outils mathématiques permettant d'imaginer de nouveaux système de mesure de la confiance.

5.3 Critère de conception

Dans cette section, nous nous concentrons sur les propriétés nécessaires pour définir un système d'authentification transparent et adaptable à l'écosystème actuel en matière de confiance numérique.

Prouver que l'utilisateur possède l'identité dont il se réclame est l'élément central dans les différents frameworks d'authentification. Il est alors possible de représenter formellement un framework d'authentification en considérant les preuves fournies

par l'utilisateur au système d'authentification. Cette preuve résulte alors du facteur d'authentification.

Le degrés de confiance dans la preuve fournie par un facteur d'authentification va justifier son utilisation. Les frameworks d'authentification établissent une liste de critères afin d'assurer la robustesse d'une preuve donnée. Plus concrètement, si nous considérons le framework Indien, une preuve biométrique fournie au travers d'un facteur d'authentification biométrique est considérée plus sécurisée que les autres preuves. De la même manière, dans le framework Européen et le framework ISO, les éléments sécurisés sont considérés comme plus sécurisés, leur accordant une place obligatoire afin d'obtenir le niveau d'assurance le plus élevé.

Nous proposons les propriétés suivantes qu'un framework d'authentification doit remplir pour pouvoir s'adapter au contexte mobile tout en conservant les authentifications plus traditionelles.

5.3.1 Etat neutre

Le framework doit pouvoir représenter un état neutre dans lequel aucune preuve n'a encore été fournie. Dans cet état, le niveau d'assurance ne doit ni afficher de la confiance ni de méfiance à propos de l'utilisateur.

5.3.2 Corrélation entre les facteurs

Deux facteurs peuvent être corrélés. Dans les modèles actuels, afin de réaliser une authentification à deux facteurs et d'éviter une corrélation entre les facteurs, le second facteur doit être d'une autre catégorie (biométrique, savoir, possession). Si cette condition permet d'exprimer ce que deux facteurs corrélés sont, il ne permet d'exprimer ce que sont deux facteurs non corrélés et n'implique pas que deux facteurs puissent être partiellement corrélés. Deux preuves peuvent être corrélées pour deux raisons :

— Le protocole utilisé pour les transmettre
— Les facteurs sont corrélés (par exemple : utiliser un code PIN pour réaliser un paiement nécessite de posséder la carte à puce)

5.3.3 Preuves ordonnées

Certaine preuves sont plus fortes que d'autre. De plus, il est possible d'augmenter la force d'une authentification en combinant les facteurs. Le principe utilisé dans une authentification multifacteur est d'augmenter le niveau de confiance dans l'authentification si une nouvelle preuve concluante est présenté. Afin de cconserver ce

principe trivial, le niveau de confiance dans l'identité de l'utilisateur doit augmenter si les authentifications respectives sont concluantes.

5.3.4 Preuves imbriquées

Une preuve imbriquée est une preuve qui n'est pas directement présentée au vérifieur. En lieu et place, la vérification est réalisée au travers de la vérification d'une autre preuve. Un exemple concret est l'authentification par carte à puce avec un code PIN. Si on se réfère à [41], il s'agit d'une authentification à deux facteurs qui garantie le niveau de sécurité maximal (LOA4). Il est important de prendre en compte le fait que la vérification du code PIN est déléguée à la carte à puce. En effet, celui-ci est vérifié localement sur la carte à puce. Cela signifie que le vérificateur final ne peut pas savoir si le code PIN a été entré correctement ou non. Il doit se résoudre à faire confiance à la carte. Il y a alors une dépendance entre la vérification du code PIN en regard de la carte à puce. Il est nécessaire d'exprimer cette dépendance de manière formelle.

5.3.5 Représentation du contexte

Le contexte offre des informations supplémentaires à propos de l'utilisateur qui caractérise sa situation. Les systèmes d'authentification comportementaux se basent sur le fait que la plupart des être humains ont des habitudes de vie avec une faible entropie[120]. Cela permet de construire un modèle pour chaque utilisateur. Dans ce cas, le contexte peut être décrit comme une cohérence avec un modèle d'utilisateur prédéfini. De cette manière, le contexte peut être représenté comme un score de similitude entre 0 et 1. Bien sûr, les limites de cet intervalle peuvent être pondérées en fonction de l'importance à donner au contexte dans le résultat d'authentification.

5.3.6 Érosion de la confiance

Une fois qu'une session a été ouverte en utilisant un moyen d'authentification ponctuel comme un mot de passe, l'utilisateur est authentifié avec un certain niveau de confiance. Si l'utilisateur quitte son poste, la session est toujours ouverte avec le même niveau de confiance. Ceci ne devrait pas arriver si nous considérons un niveau d'authentification en temps réel, dans lequel la confiance offerte par une preuve doit décroître avec le temps. Nous appelons ce phénomène érosion de la confiance. Cette érosion peut être modifiée en fonction du contexte. Par exemple, si un utilisateur utilise un appareil sans aucune interruption, il n'y a pas besoin de décroître le niveau de confiance.

5.3.7 Représentation de la confiance

Le niveau de confiance doit être représentée avec une valeur sur une échelle continue et pas seulement avec quatre ou cinq niveaux. Un framework d'authentification doit être capable de donner un score et de comparer deux méthodes d'authentification lorsqu'elles combinent plusieurs facteurs d'authentification.

5.4 Élaboration du modèle

5.4.1 Théorie de Dempster Shaffer

La vision de l'authentification est généralement basée sur un scénario probabiliste. Afin d'authentifier un utilisateur, le système essaie de répondre à l'assertion : Est-ce l'utilisateur légitime ? Nous obtenons alors un ensemble de solutions possibles : $\Theta = \{g, a\}$ où g est l'utilisateur légitime et a est un imposteur.

Si nous appliquons les probabilités classiques à ce problème, nous obtenons alors une solution où $P(g) = 1 - P(a)$. Autrement dit, si l'utilisateur n'est pas reconnu, il est automatiquement considéré comme un imposteur. Cette vision ne laisse pas de place au doute ou à l'incertitude.

Pour cette raison, les probabilités classiques ne sont pas capables de manipuler correctement le niveau de confiance lié à une authentification. Nous souhaiterions obtenir une vision plus proche du modèle humain. Nous proposons une vision qui introduit la notion de doute et d'incertitude sur l'identité de l'utilisateur.

La théorie de l'évidence de Dempster Shaffer [105] prend en compte cet état d'incertitude. La théorie de la croyance peut être vue comme une extension de la théorie des probabilités classiques en y ajoutant une expression explicite de l'ignorance.

La théorie de l'évidence a été originellement proposée par Dempster en 1960 [105] comme un modèle prenant en compte une probabilitée inférieure et supérieure. Shaffer a proposé par la suite un modèle pour exprimer la fusion des croyances.

La théorie de l'évidence abandonne le principe d'additivité des probabilités classiques et autorise l'utilisateur à assigner des masses d'évidence à l'ensemble des sous-ensembles de l'espace des états.

Soit θ l'ensemble de discernement, cet ensemble contient une liste exhaustive des éléments. Par exemple, $\theta = \{g, a\}$. La proposition $\wp(\theta)$ sera l'ensemble des parties de θ incluent l'ensemble vide $\emptyset$. Dans notre exemple, $\wp(\theta) = \{\emptyset, g, a, \{g, a\}\}$.

Lorsqu'un capteur réalise une mesure d'un état X, il assigne une croyance basique, aussi appelée fonction de masse ou juste masse $m(X)$.

Cette masse vérifie les équations suivantes [105] :

$$m(\emptyset) = 0 \qquad \text{and} \qquad \sum_{X \in \wp(\theta)} m(X) = 1 \tag{5.1}$$

A partir de là, la fonction de croyance $Bel()$ et la fonction de plausibilité $Pl()$ sont définies comme :

$$Bel(A) = \sum_{B|B \subseteq A} m(B) \tag{5.2}$$

$$Pl(A) = \sum_{B|B \cap A \neq \emptyset} m(B) \tag{5.3}$$

L'équation (5.2) représente la limite inférieure et l'équation (5.3) la limite supérieure pour un état A. Cette théorie de l'évidence est utilisée dans notre contribution pour définir un modèle d'authentification. La théorie de Dempster Shaffer propose une règle pour combiner deux masses m_1 et m_2.

$$m_{1,2}(\emptyset) = 0 \tag{5.4}$$

$$m_{1,2}(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B).m_2(C) \tag{5.5}$$

où

$$K = \sum_{B \cap C = \emptyset} m_1(B).m_2(C) \tag{5.6}$$

5.4.2 Confiance dans une preuve

Nous basons ce modèle d'évaluation de l'authentification sur l'attribution de masses aux preuves. Nous obtenons alors les masses $m(g)$ et $m(\theta)$ assignées à une preuve. Pour calculer ces masses, deux critères peuvent être pris en compte :

— La force de la preuve

— La corrélation avec les preuves précédentes

Nous exposons ces deux critères à la suite.

Force d'une preuve

La force d'une preuve S est subjective. Elle dépend de la robustesse de la preuve et d'autres propriétés comme la possibilité de détection d'un vol par exemple. Il est usuel et pratique de classer la force des preuves en trois catégories allant de faible

à bon. Nous fixons l'hypothèse que la force des preuves peut être classée en trois catégories : Faible, Moyen, Bon. On peut alors donner une valeur numérique à chaque catégorie pour S en se référant à la figure 5.2. Seule la partie supérieure ($[\frac{1}{2}; 1]$) a été gardée. En effet, une preuve d'authentification ajoute de la confiance.

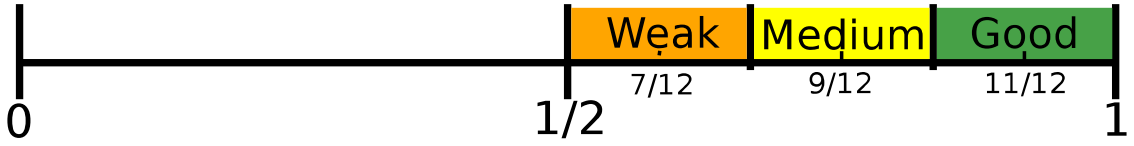

FIGURE 5.2 – Strength of the proofs.

Corrélation

Cette force est alors pondérée en fonction des corrélations avec les preuves exposées précédemment. La corrélation $Corr$ est déterminée par la corrélation à la fois entre les modalités (même catégorie de facteur) et par le protocole utilisé (même média...). On lui attribue les valeurs 0, $\frac{1}{2}$ ou 1 pour $Corr$ si les modalités sont corrélées ou partiellement corrélées.

Nous pouvons alors attribuer une masse $m(g) = S \times (1 - Corr)$. Dans le cas d'une authentification infructueuse, un processus équivalent est réalisé en attribuant une masse à $m(a)$ cette fois ci. Il faut cependant prendre en compte non plus la force de l'authentification mais son usabilité.

5.4.3 Calcul d'un score de confiance

En combinant les preuves, on obtient un intervalle de valeurs possibles entre $Bel(g)$ et $Pl(g)$. Une solution pour prendre en compte le score de l'authentification continue C et de ne ressortir qu'un scalaire est d'utiliser une fonction pignistique [**?**]. Ceci peut être vu comme le placement d'une mise sur le score de confiance de l'individu en utilisant Bel(g), Pl(g) et le score d'authentification continue. Le score de confiance L est alors calculé de la manière suivante :

$$L = Bel(g) + C \times (Pl(g) - Bel(g)) \tag{5.7}$$

5.4.4 Force d'une preuve imbriquée

Dans le cas d'une preuve imbriquée, la vérification d'une première preuve est nécessaire pour déduire que la seconde preuve a été vérifiée. Il y a donc une dépendance

entre les preuves. Cette dépendance peut s'écrire formellement $S = S_{handler} \times S_{nested}$. Une corrélation apparaît entre les preuves. On attribue la valeur de $\frac{1}{2}$ à la corrélation.

5.4.5 Combinaison des preuves

Si les preuves sont indépendantes, il est possible d'appliquer la règle de Dempster Shaffer. La combinaison peut alors être définie comme $\alpha_{1,2} = \alpha_1 \oplus \alpha_2$ pour évaluer une preuve 1 combinée avec une preuve 2, où α est défini dans l'équation (5.9). Si les preuves résultent d'une authentification réussie, nous obtenons [105] :

$$\begin{cases} m_{1,2}(g) = m_1(g) + m_2(g) - m_1(g)m_2(g) \\ m_{1,2}(\theta) = m_1(\theta)m_2(\theta) \end{cases} \tag{5.8}$$

5.4.6 Érosion de la confiance

L'érosion de la confiance peut apparaître à cause de l'inactivité. La confiance résultante à propos de l'état g peut être calculée de la manière suivante :

$$^{\gamma}\alpha = \begin{cases} m(g) = \gamma \cdot m(g) \\ m(\theta) = \gamma \cdot m(\theta) + (1 - \gamma) \end{cases} \tag{5.9}$$

Par exemple, γ peut être déterminé en utilisant les équations suivantes [26] : $\gamma = \frac{1}{(t_{now} - t/\rho)^r}$, où r est le coefficient de vieillissement. Augmenter r implique que la confiance dans une preuve va décroître plus rapidement. ρ est la granularité : une granularité plus grande permet de regrouper les événements qui apparaissent en même temps.

5.4.7 Élément neutre

Un élément neutre peut être introduit pour définir l'état initial du système dans lequel il n'y a pas de preuve. Cette valeur particulière $\alpha = (0, 1)$ représente un état d'ignorance totale. Dans la section suivante, nous simulons un scénario d'usage et attribuons des valeurs équivalentes aux niveaux d'assurance afin de comparer les deux approches.

5.5 Utilisation du modèle

Nous démontrons deux cas d'usage de notre modèle. Dans un premier temps, nous convertissons les niveaux d'assurance afin d'obtenir des seuils équivalents. Puis, un

niveau d'assurance est calculé en fusionnant une authentification continue avec des authentifications ponctuelles plus classiques. C'est illustré au travers d'un scénario d'usage et montre comment ce modèle peut être utilisé dans des conditions réelles, où les niveaux d'assurances sont actuellement utilisés.

5.5.1 Niveaux d'assurance

Le modèle proposé peut permettre d'être utilisé pour calculer des valeurs pour les niveaux d'assurances. Cela permet de transformer les niveaux discrets en un seuil à atteindre. Dans ce chapitre, nous avons choisi de suivre le framework ISO qui est implémenté dans de nombreux systèmes de gestion d'identités. Notre système repose sur les facteurs d'autentification présentés dans la table 5.3, classés dans les catégories exposées dans la section 5.4.2.

TABLE 5.3 – Force de chaque facteur d'authentification.

Faible	Moyen	Bon
Mot de passe code PIN	Empreinte OTP	Carte à Puce

Les niveaux d'assurance ne prennent pas en compte les méthodes d'authentification futures telles que l'authentification continue. Afin d'évaluer une équivalence, nous prenons alors un niveau d'authentification continue neutre : $C = 0$. Il n'y a pas de donnée sur les attaquants dans l'établissement des niveaux d'assurance $m(a)$ est toujours égal à 0. Donc, $m(\theta) = 1 - m(g)$. La force des différents facteurs est établie avec la figure 5.2. Les résultats sont exposés dans la table 5.4.

On remarque la faible granularité des niveaux d'assurance. Cependant, ces seuils calculés nous permettent d'obtenir des équivalents dans notre framework. Ces seuils seront alors utilisés pour assurer une rétro-compatibilité. Nous allons maintenant illustrer le modèle de manière pratique en l'exposant au travers d'un cas d'usage.

Scénario d'usage

La figure 5.3 montre l'évolution du niveau de confiance durant une journée. Les données de l'authentification continue ont été réalisées à l'aide des données issues de la base du MIT Reality Mining Dataset [39]. Pour simplifier l'usage et parce que ce n'est

TABLE 5.4 – Seuils équivalents aux LoA.

Niveau	LoA1	LoA2	LoA3	LoA4
Score équivalent	0.58	0.75	0.89	0.96

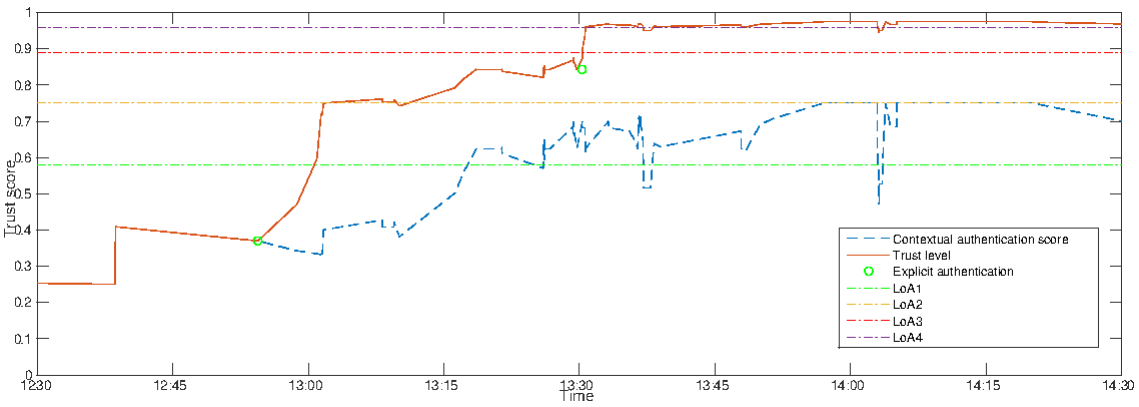

FIGURE 5.3 – Évolution au cours du temps pour une authentification continue donnée.

pas le but de ce chapitre que de se concentrer sur le mécanisme d'authentification continue, le score d'authentification continue est une simple probabilité d'utiliser une application donnée à une heure donnée pour un utilisateur donnée. Prenons notre utilisatrice Alice. Si elle lance une application donnée à une heure habituelle alors son score d'authentification va augmenter dans ce scénario. Même s'il ne s'agit pas d'un mécanisme optimal d'authentification continue, il permet d'exposé simplement notre modèle.

Au début de la simulation (Point A sur la figure), le score d'authentification est en dessous de 0.5. On peut en déduire que notre utilisatrice Alice n'est pas dans un contexte habituel. Dans le début d'après midi, elle souhaite accéder à son compte mail professionnel : elle entre alors son mot de passe au point B. Par conséquent, le niveau de confiance est augmenté et fini par atteindre un niveau équivalent au LoA2. En plus de cela, un comportement usuel est demandé, le niveau d'authentification continue augmente alors au cours du temps (point C). Logiquement, le score de confiance est lui aussi augmenté. A 13h30, Alice souhaite accéder à son compte en banque. Même si l'authentification continue offre un meilleur score, c'est toujours insuffisant pour atteindre le niveau LoA3.

Un mot de passe à usage unique est entré par Alice et le niveau d'assurance devient suffisamment conséquent pour accéder à son compte bancaire (point D).

On observe que, plus il y a de facteurs, moins le niveau de confiance est influencé par le score de l'authentification continue. Ceci permet d'éviter les effets d'un score trop faible et de contrer les éventuels effets de bords lié aux techniques d'authentification biométriques. A l'opposé, une authentification continue forte améliore grandement l'usabilité d'un système et évite des interruptions régulières pour saisir un facteur d'authentification.

Un effet du modèle proposé implique que le niveau de confiance associé à une authentification multifacteur décroît plus rapidement que celles basées sur une authentification simple facteur. Cela est dû à l'érosion de la confiance qui est appliquée de manière indépendante sur chacune des preuves. Le fait de prendre en compte l'authentification continue, permet de contrer cette effet au cours du temps.

Cependant, si le niveau de confiance L permet de toujours obtenir un niveau de confiance qui inclut l'authentification continue, lorsqu'il n'y a aucune autre preuve de disponible, le niveau global n'est donnée que par l'authentification continue à cause de l'équation (5.7).

Les améliorations proposées dans notre solution sont comparées aux modèles de l'état de l'art dans la table 5.5.

Framework	*Élément neutre*	*Corrélation*	*Ordre des preuves*	*Preuves imbriquées*	*Authentification continue*	*Érosion*	*Présentation de la confiance*
ISO			✓				
eID			✓				
NeAF	✓		✓				
ePramaan	✓		✓				
[45]	✓				✓		
[26]	✓				✓	✓	✓
[84]		✓			✓	✓	
[58]			✓				
[92]	✓	✓	✓	✓	✓	✓	
Contribution	✓	✓	✓	✓	✓	✓	✓

TABLE 5.5 – Propriété de la solution proposée en comparaison des autres systèmes de l'état de l'art.

5.6 Conclusion

Dans ce chapitre, nous avons vu comment était évalué le niveau d'une authentification dans les documents institutionnels. Nous remarquons que cette évaluation n'est pas satisfaisante et très empirique. Elle ne prend pas en compte les forces des

systèmes d'authentification et se limites aux systèmes d'authentification statiques. De plus, la vision de ces frameworks est biaisée par l'aspect culturel qui incite les gouvernements à accorder plus de confiance à certaine méthodes d'authentification en fonction de leur histoire plutôt qu'en fonction des faits.

C'est pourquoi il est nécessaire d'introduire d'autre métriques pour évaluer le niveau d'une authentification. Dans l'état de l'art, aucun travaux ne prenant en compte l'ensemble des aspects nécessaires aux recherches proposées dans ce manuscrit, nous avons développé notre propre framework pour évaluer le niveau d'une authentification.

Bien que celui ci offre plus de liberté et une certaine rationalité dans l'évaluation des scénarios d'authentification, ce framework reste un framework expérimental qui ne permet pas d'être appliqué tel quel dans un contexte industriel. Il permet cependant de pointer du doigt des incohérences dans l'établissement des niveaux d'assurance traditionnels.

Chapitre 6

Intégration des recherches dans un environnement DevOps

Ce chapitre expose le transfert des travaux de cette thèse dans un environnement industriel. La proposition faite , actuellement en cours d'implémentation, permet de prendre en compte les problématiques issue de la recherche et de les combiner avec les propositions industrielles.

Sommaire

6.1 Introduction

Les méthodes classiques de management sont difficilement adaptables aux processus de recherche. L'exemple de la société 3M illustre la difficulté d'intégrer des méthodes de management modernes basées sur l'amélioration continue[59]. En effet, la Société 3M a fortement compromis son processus créatif suite à l'application de méthodes de production rationnelles basées sur l'amélioration continue.

Cependant, dans des structures importantes comme la société Orange, qui héberge ce travail de doctorat, le temps de passage d'un produit du stade de recherche au stade de production peut s'avérer trop long. Afin de contourner cette difficulté,

certaines sociétés placent la recherche au centre de leur fonctionnement, comme en témoigne l'exemple de Google [109].

Afin de comprendre comment intégrer dans un processus industriel d'amélioration continue les contributions de recherche défendues dans cette thèse, il est nécessaire de définir ce qu'est le développement opérationnel.

6.2 Le développement opérationnel

Dans les entreprises modernes, les équipes sont organisées par compétences qui correspondent à des tâches bien définies. Ceci amène à des objectifs spécifiques pour chaque équipe.

Ainsi, la productivité des équipes de développement logiciel peut se mesurer par le ratio entre les nouveaux points de fonctionnalité produits et l'unité de temps par homme, tandis que celle des équipes dédiées à la maintenance des serveurs sera mieux évaluée par des indicateurs de qualité tels que la robustesse et la fiabilité du service (uptime, crashes, latence).

Cependant, ces deux objectifs peuvent apparaître contradictoires. Les équipes opérationnelles oeuvrent en effet pour la stabilité des services qu'elles opèrent, ce qui peut aller à l'encontre de l'introduction de nouvelles fonctionnalités : une mise en production d'une nouvelle version, une migration ou un patch correctif sont autant de risques que souhaite minimiser la production. A l'inverse, les équipes de développement seront jugées performantes si le logiciel évolue fréquemment : leur management souhaite voir le plus de changements possibles. On observe alors un fonctionnement en silo comme l'illustre la figure 6.1.

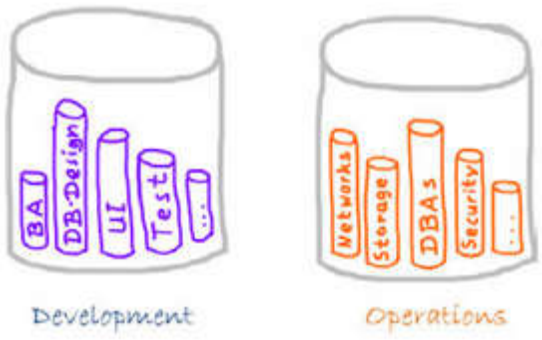

FIGURE 6.1 – Fonctionnement en silo des équipes [60]

C'est essentiellement pour casser ce fonctionnement en silo que l'idée de Développement Opérationel a fait son apparition. On peut le définir de la manière

suivante.

Definition 11 (DevOps[38] [1]). *Le Développement Opérationel (DevOps) est une approche organisationnelle qui se focalise sur l'empathie et les collaborations interfonctionelles au sein et entre les équipes - spécialement le développement et l'opérationnel - dans les sociétés de développement logiciel, dans le but d'opérer des systèmes résiliants et d'accélérer la livraison des changements.*

La résilience est la propriété qui permet de contenter les deux objectifs apparemment antagonistes présentés plus haut. Un système résilient possède en effet la propriété de pouvoir s'auto-réparer rapidement en cas de panne, ce qui contente les opérationnels. En même temps, il est hautement adptatif à son environnement, ce qui permet aux développeurs de contenter leur besoin de créativité. On déduit de cette définition que le DevOps, inclut :

— des outils d'automatisation ;
— une culture commune ;
— des outils de mesures ;

Le Devops a permis d'accélérer la mise en production des développements. En appliquant le même paradigme, nous allons voir comment les travaux de cette thèse ont été intégrés dans un environnement acclimaté à la culture Devops et aux contraintes afférentes.

6.3 De la preuve de concept au processus d'intégration

Avant de pouvoir intégrer une technologie, comme celle de l'authentification transparente, il est nécessaire de pouvoir la faire migrer des équipes en charge de la recherche vers les équipes en charge du développement. Dans la pratique, cette migration est le plus souvent un processus compliqué où les échecs sont nombreux. Dans un environnement industriel comme celui d'Orange, de manière caricaturale et uniquement dans le but de poser la problématique, nous avons observé que :

— les chercheurs ne disposent pas du temps suffisant pour acculturer les développeurs parce qu'ils sont objectivés sur des nouvelles recherches, et,

1. Traduit de l'anglais : DevOps is an organizational approach that stresses empathy and cross-functional collaboration within and between teams – especially development and IT operations – in software development organizations, in order to operate resilient systems and accelerate delivery of changes.

— les développeurs négligent la difficulté de prise en main des résultats de recherche, parce que la démonstration qu'il leur en a été faite peut laisser croire que le système sous-jacent est simple.

Nous proposons ici une passerelle naturelle pour passer d'un monde à l'autre, basée sur la conception d'un prototype et de son transfert dans un environnement d'industrialisation.

A l'issue des expérimentations en laboratoire, nous avons proposé une preuve de concept fonctionnel de bout en bout. En effet, l'intégration est une étape complexe du processus de production de l'industrie du logiciel ; l'intégration des résultats de recherche dans un cas d'usage développé de bout en bout présente à cet effet deux avantages majeurs :

— la possibilité d'identifier les problèmes d'intégration avant même la spécification des premiers développements ;
— la mise en place d'une expérience professionnelle commune entre les développeurs et les chercheurs.

Ainsi le prototype, qui était jusque-là destiné à la communication externe de l'équipe de recherche, devient en plus un outil de communication interne avec l'équipe de développement. Afin d'accélérer le processus de transmission et d'augmenter des interactions fructueuses entre les équipes, il est intéressant d'intégrer les équipes de développement à la réalisation de la preuve de concept, jusque-là strictement réservée aux équipes de recherche.

6.3.1 Développement de la preuve de concept

Les objectifs d'une preuve de concept sont multiples :

— prouver la faisabilité technique ;
— communiquer sur les intérêts industriels des recherches amont ;
— monter en compétence sur une technologie.

Dans un processus industriel, il est indispensable de s'assurer au plus tôt que les applications des concepts de recherche seront potentiellement commercialisables. Dès la phase de recherche, la visibilité interne à l'entreprise est donc primordiale pour faire évoluer le concept avec les retours des métiers ciblés mais aussi pour susciter l'adhésion du management. Idéalement, le concept est validé quand il est intégré à la stratégie d'entreprise. A cette fin, le prototype devrait donc présenter de manière pédagogique et ludique des solutions capables de créer de nouveaux besoins ou bien de répondre à des besoins existants non couverts par l'état de l'art. Une photographie de la mise en place de ce prototype est présentée sur la figure 6.2.

L'intégration de l'équipe de développement dès la fabrication de la preuve de concept devrait être un avantage pour les objectifs pédagogiques et esthétiques du projet. Parrallèlement, elle peut apporter des compétences sur les technologies et méthodes spécifiques du développement logiciel que les équipes de recherche ne possèdent généralement pas.

De plus, cette nouvelle organisation du travail devrait permettre de de fluidifier le processus de transmission du savoir. Les développeurs peuvent acquérir à leur rythme les nouvelles technolgies issues de la recherche (algorithmes, outils, disciplines scientifiques et techniques) tout en imposant un certain nombre de standards issus de la culture DevOps.

Dans le cadre de ces recherches, les compétences transmisent aux équipes de développement ont été les connaissances liées à la caractérisation des systèmes biométriques par exemple tandis que les équipes de recherches ont pu se confronter aux problématiques du terrain et découvrir des solutions industrielles de déploiement de services ou encore des librairies spécifiques de traitement de donnée temps réel.

Le prototype devient alors un outil d'échange bilatéral permettant de prendre en compte aussi bien les contraintes des équipes de recherche que celles des équipes de développement.

6.3.2 Rayonnement de la preuve de concept

Pour qu'une preuve de concept soit efficace, il est nécessaire qu'elle obtienne un rayonnement important, du moins au sein de l'entreprise, afin de susciter un intérêt commercial. Intégrer les équipes de développement à l'élaboration favorise ce rayonnement interne et permet de soulever des contraintes techniques dès le commencement du projet.

Prenons l'exemple suivant, une technologie mise au point par les équipes de recherche doit être intégrée dans un environnement existant. Cependant, cette technologie nécessite d'utiliser une base de données particulière qui n'est pas maîtrisée par les équipes de développement.

Dans un processus d'intégration classique, cette difficulté est un frein majeur. Dans l'environnement de transfert des connaissances proposé et expérimenté avec les travaux présentés dans ce manuscrit, le point est pris en charge beaucoup plus rapidement et l'acquisition du savoir se fait en même temps que la conception du prototype. Ceci permet d'améliorer le spectre de compétences de l'entreprise tout en accélérant la mise sur le marché.

Enfin, il ne faut pas négliger les bénéfices de la communication externe dans le processus de transfert des résultats de recherche. Dans notre projet, le prototype

FIGURE 6.2 – Photographie de la démonstration mise en place au Salon de la recherche d'Orange Labs sur l'évolution de l'indice de confiance dans une authentification transparente.

a été cité plusieurs fois dans les journaux [27, 40, 29]. Recités via les outils de communication internes de l'entreprise, ces communications amènent des nouveaux avis, permettent de convaincre d'autres directions de l'entreprise et peuvent être un levier pour motiver le management en faveur du projet dans une situation d'arbitrage des ressources.

6.4 L'intégration industrielle

Afin de pouvoir évaluer cette preuve de concept, il est aussi nécessaire de développer des outils. En effet, les résultats obtenus dans un scénario d'usage doivent être comparables à ceux obtenus lors des recherches en laboratoire. Les laboratoires développent généralement des outils d'évaluation [117].

6.4.1 Outil de test

Dans le cadre de ces recherches, nous avons fait le choix de développer notre propre outil d'évaluation. Les équipes de recherches doivent se comparer à l'état de l'art. Il est donc indispensable d'évaluer les algorithmes.

Cependant, il est important de séparer le travail de recherche qui nécessite un environnement créatif et dans lequel l'échec est fréquent de celui du développement

habitué à des améliorations incrémentales dans lequel les aléas liés au processus créatif doivent avoir le moins d'impact possible.

Afin de répondre à cette problématique, nous avons choisi une solution technique basée sur un élément commun aux deux équipes et qui leur permettent de tester leurs solutions. Dans le cas de l'équipe de recherche, le but est d'évaluer leur solution par rapport à l'état de l'art. Pour l'équipe de développement, l'objectif est différent, il est de ne pas entraîner de régression et de valider que les développements offrent bien les résultats attendus. Les tests deviennent donc un espace d'échange privilégié entre les deux équipes.

De cette constatation, nous pouvons déduire que même si les objectifs de tests sont différents, les interfaces logiciels nécessaires sont similaires. L'architecture de l'outil développé est présentée dans la figure 6.3.

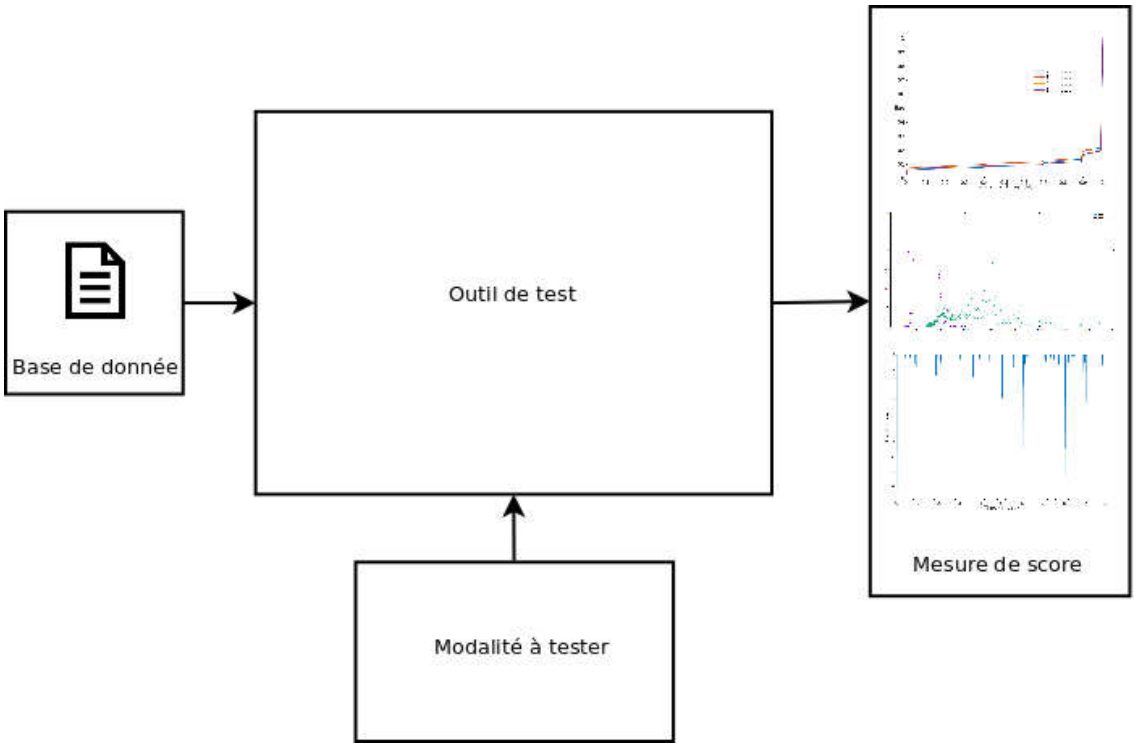

FIGURE 6.3 – Architecture de l'outil d'évaluation des modalités

Il est donc possible d'imaginer un cadre logiciel commun au développement et à l'évaluation des propositions réalisées en recherche. Nous avons ainsi implémenté et évalué nos recherches sous forme de blocs modulaires qui peuvent être confrontés à l'état de l'art. De plus, elles peuvent être récupérées facilement dans un usage plus général par les équipes de développement pour être intégrées avec des changements très mineurs dans un produit de qualité industriel plus global.

6.4.2 Bloc modulaire

Dans le cas d'étude qui est abordé, nous intégrons des solutions d'authentification transparentes à un environnement industriel. Le bloc élémentaire que les équipes de recherches souhaitent développer se découpe naturellement par modalités d'authentification.

Afin de pouvoir être intégré de manière globale, dans un processus industriel, on peut séparer cette modalité en une succession de blocs. La figure 6.4 reprend ces blocs.

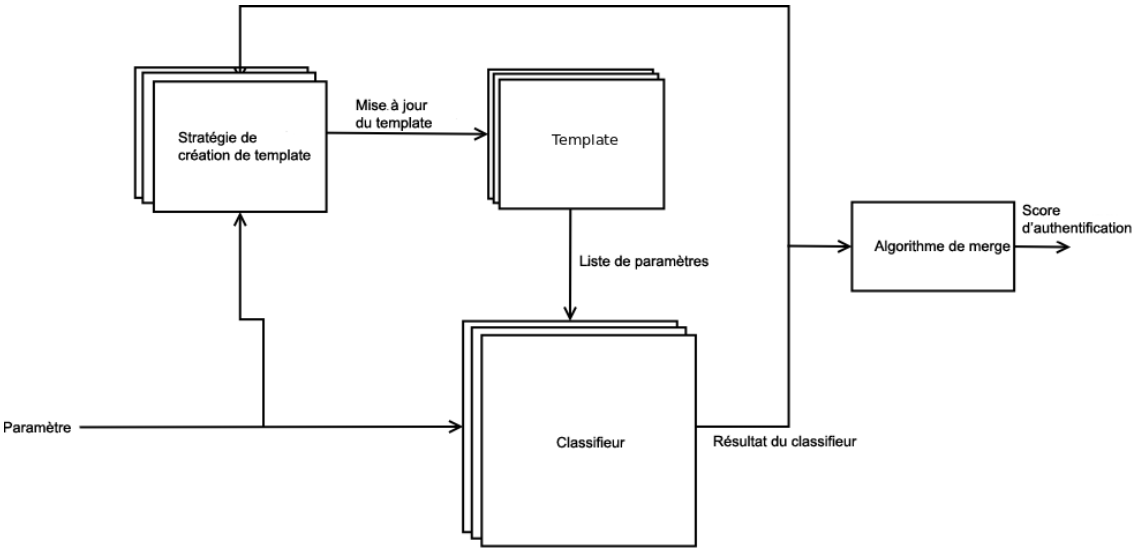

FIGURE 6.4 – Framework d'authentification transparent

Il est intéressant de discuter les aspects modulaires de ce cadre logiciel. Chacun des blocs est utilisé de manière indépendante et permet une réutilisation. Le schéma classique d'évaluation d'une nouvelle modalité d'authentification biométrique ou bio-comportemental est globalement toujours similaire et peut se découper de la manière suivante :

1. Acquisition de données
2. Choix des caractéritiques (features)
3. Construction du modèle de référence (template)
4. Evaluation des classifieurs
5. Evaluation des mises à jour du modèle de référence (template)

Afin d'évaluer les différents classifieurs, il est nécessaire de comparer plusieurs solutions. La mise en place d'un cadre logiciel de ce type permet à la fois d'ac-

célérer la recherche puisque les classifieurs peuvent être réutilisés d'une modalité d'authentification à l'autre et les évaluations peuvent être menées en parallèle.

Cela n'enlève rien au travail créatif du chercheur qui doit continuer à évaluer de nouvelles solutions et à fournir de nouvelles techniques de classification. Cependant, l'outil devient un outil de communication des travaux de recherche puisqu'il lui permet de partager de manière concrète et fonctionnelle de nouvelles idées sous forme d'implémentation logicielle. De plus, cet outil devient un cadre qui permet de valoriser les travaux tout en assurant un suivi des évolutions de la recherche.

Implémentation d'une nouvelle modalité d'authentification

Une difficulté majeure dans l'implémentation d'une nouvelle modalité d'authentification au produit construit par l'équipe de développement réside dans la dissociation des rôles liés à la recherche tout en associant les travaux des équipes de recherche et des équipes de développement. A l'issue des évaluations, le travail réalisé peut être comparé à l'état de l'art.

Une fois les travaux de recherche aboutis, la nouvelle méthodologie appliquée vise à les injecter dans un environnement industriel. Cette étape consiste à livrer l'ensemble des modules nécessaires et de les intégrer directement dans les tâches des développeurs.

L'architecture de test étant calquée sur l'architecture de développement, l'intégration est très simple et consiste à un ajout de code simple et efficace. L'utilisation d'un modèle client-serveur comme il a été évoqué dans ce manuscrit facilite encore le développement.

Les développeurs peuvent alors appliquer les principes classiques du DevOps à ces nouveaux modules et les tester selon leurs critères (temps de réponse, robustesse, ...). Les résultats des tests sont autant de nouvelles données qui permettent d'aider les équipes de recherche et de nourrir leurs réflexions avec des problématiques industrielles.

Les tests passés, l'outil est passé en production et devient un nouvel outil de recherche puisqu'il permet de collecter des données à l'échelle qui sont nécessaires pour la qualification scientifique des algorithmes implémentés.

6.4.3 Amélioration continue

Les solutions industrielles sont souvent en décalage par rapport à l'état de l'art académique. Dans le domaine de la sécurité informatique, ceci peut être dangereux puisque des failles peuvent apparaître dans les environnements de production.

Il semble possible d'utiliser les outils développés pour la recherche afin de diminuer ce délai entre recherche et production. L'outil de tests proposé dans la section 6.4.1 peut prendre les mêmes données en entrée que le serveur d'authentification transparent. Il est alors possible de continuer à améliorer les modalités d'authentification avec des données représentatives de la réalité. Pour effacer les biais liés à la collecte de données, il suffit de dupliquer le flux d'entrée du service de production vers l'outil de test.

Cette dernière proposition est indispensable à l'évolution de l'ensemble des services qui nécessite des techniques d'apprentissage. En effet, comme nous l'avons vu dans la section 4.5.2 un biais apparaît avec le temps. Même si nous avons expérimenté des solutions palliatives, nous ne pouvons avoir aucune certitude sur l'évolution des comportements des utilisateurs sur une période de plusieurs années puisque à notre connaissance, aucune base de données publiques ne possèdent d'informations de ce type.

6.5 Conclusion

Ce chapitre expose les méthodes utilisées lors de cette thèse pour intégrer les travaux de recherche. L'interface d'échange entre les équipes chargées du développement et les équipes de recherche a pu se réaliser en utilisant un framework commun.

En effet, les méthodes de développements modernes et la recherche s'appuient sur les données (fouille de données, archivage...) et sur le test logiciel (évaluation des performances, robustesses). Ces point communs ont permis de mettre en place une interface d'échange qui permet non seulement d'offrir des services basés sur les innovations de l'état de l'art mais aussi d'apporter aux équipes de recherches des données issues des environnements de productions dans un futur proche.

Chapitre 7

Conclusion et perspectives

Au cours de cette thèse, nous nous sommes intéressés aux solutions d'authentifications transparentes et à leurs intégration dans un environnement industriel. Après avoir introduit cette thèse dans le chapitre 1, nous avons présenté le contexte dans le chapitre 2.

Une définition de l'authentification est proposée et les différents types de facteurs d'authentification sont présentés. Les concepts fondamentaux de l'authentification sont abordés et nous définissons en particulier les exigences que doit respecter un système d'authentification. Ces exigences peuvent être regroupées en trois familles :

— **Vie privée :** Ceci est imposé par des organismes de régulation comme la CNIL ou par des volontés des utilisateurs. L'objectif est de garantir les libertés fondamentales d'un individu.

— **Sécurité :** Qui permet avec l'expérience d'évaluer la robustesse d'un système d'authentification. Elle inclue aussi bien les mécanismes cryptographiques que les scénario d'usages.

— **L'usabilité :** Qui est défini selon la norme ISO 9241-11 et permet d'évaluer un mécanisme par rapport à l'aisance avec laquelle un utilisateur peut l'utiliser.

A la suite des exigences, nous avons présenté les principaux travaux d'authentification transparente sur mobile. Le chapitre 2 reprend les méthodes de collecte de données sur téléphone mobile et l'évaluation des solutions d'authentification. Le chapitre 3 propose 3 contributions pour authentifier un utilisateur tout en respectant sa vie privée. L'intérêt majeur réside dans la proposition de solutions concrètes pour implémenter des mécanismes d'authentification transparente dans une architecture client serveur. Nous décrivons une solution basée sur la théorie de Dempster Shaffer pour évaluer la confiance dans l'authentification. Enfin, le chapitre 6 propose une solution effective pour industrialiser les travaux de cette thèse dans un environnement

DevOps.

7.1 Bilan

Dans ce manuscrit, nous avons exposé trois contributions majeures.

La première contribution est la mise au point et l'évaluation de différentes méthodes d'authentification transparente respectueuse de la vie privée. Cette contribution a permis de lever un verrou technologique quant à l'usage des méthodes d'authentification transparente et comportementale. En effet, un des principaux freins au déploiement et à l'adoption de ces technologies est lié à la sécurité des données personnelles de l'utilisateur. Nous avons ainsi réalisé une étude et déduire les compromis à réaliser pour garantir la protections des données personnelles de l'utilisateur. L'ensemble des solutions proposées a été implémenté sous forme de prototypes fonctionnels ce qui a permis d'établir la faisabilité technique.

La seconde contribution majeure de cette thèse est l'élaboration d'un modèle permettant de combiner des techniques d'authentification continue avec des méthodes classiques. Cette solution améliore considérablement l'usabilité des systèmes d'authentification tout en permettant une rétro compatibilité avec les méthodes actuelles basées sur des niveaux d'assurance.

Enfin, une solution permettant de transférer les acquis de cette thèse dans un environnement industriel constitue la dernière contribution majeure de cette thèse. Cette méthodologie permet actuellement de développer un système d'authentification transparent dans un but d'industrialisation.

7.2 Perspectives

La première perspective provient des avancés des performances des téléphones mobiles. Ceci ouvrira de nouvelles perspectives quant à l'implémentation de solutions de chiffrement plus rapide ainsi que de nouvelles données biométriques à utiliser dans les algorithmes d'authentification.

L'évolution des standards qui régissent l'évaluation de la confiance dans l'authentification repose aujourd'hui sur les techniques d'authentification actuelles. Ceci freine l'adoption de ces nouvelles métriques d'authentification dans les systèmes de gestion d'identités. Un large champ est encore à explorer pour normaliser une authentification en continue.

L'implémentation de telles solutions d'authentification transparente dans des environnements industriels doit être évaluée sur le long terme et adaptée. Cependant, l'utilisation de données issues des bases de données de production permet de prévoir les évolutions et d'étudier plus largement de nouvelles métriques en prenant un ensemble très complet de données. De plus, si ces données sont récoltées après y avoir appliqué un mécanisme de protection de la vie privée, il est possible d'utiliser des architectures comme celles proposées dans le chapitre 6 afin d'évaluer des nouvelles métriques d'authentification à très large échelle.

La sécurité sur téléphone mobile dépends encore d'un élément sécurisé qui peut être embarqué ou sur un serveur. Ceci implique de nombreuses attaques liées aux vulnérabilités des téléphones mobiles. Il est cependant possible d'évaluer la véracité des données en profitant des propriétés homomorphe ou de combinaisons du Biohashing lorsque le vérificateur dispose de données sûres. C'est le cas pour un opérateur téléphonique comme Orange qui peut compter sur son réseau pour vérifier ces données qui sont usurpables.

Publications

Conférences internationales avec comité de relecture

Thomas Souvignet, Julien Hatin, Fabrice Maqua, Damien Tesnière, Pierre Léger et al. Payment card forensic analysis : From concepts to desktop and mobile analysis tools Digital Investigation, Elsevier, 2014, 11 (3), pp.143-153

Hatin, J., Cherrier, E., Schwartzmann, J. J., Frey, V., & Rosenberger, C. (2016, February). A Continuous LoA Compliant Trust Evaluation Method. In International Conference on Information Systems Security and Privacy (ICISSP).

Hatin, J., Cherrier, E., Schwartzmann, J. J., & Rosenberger, C. (2017). Privacy Preserving Transparent Mobile Authentication. In ICISSP (pp. 354-361).

Conférence nationale avec comité de relecture

Hatin, J., Cherrier, E., Schwartzmann, J. J., & Rosenberger, C. (2017). Authentification basée sur des habitudes d'appel et garante de la vie privée. In COmpression et REpreésentation des Signaux Audiovisuels (CORESA).

Bibliographie

[1] Draft nist special publication 800-63b digital identity guidelines authentication and lifecycle management. [Cité pages 19 et 90.]

[2] Gergely Alpár, Jaap-Henk Hoepman et Johanneke Siljee : The identity crisis. security, privacy and usability issues in identity management. *arXiv preprint arXiv :1101.0427*, 2011. [Cité page 23.]

[3] Patricia Arias, Florina Almenárez Cabarcos, Rubén Trapero, Daniel Díaz et Andrés Marín Sánchez : Blended identity : Pervasive idm for continuous authentication. *IEEE Security & Privacy Magazine*, 2014. [Cité pages 17 et 23.]

[4] Australian governement : National e-authentication framework, 2009. [Cité pages 19 et 89.]

[5] Jakob E Bardram, Rasmus E Kjær et Michael Ø Pedersen : Context-aware user authentication–supporting proximity-based login in pervasive computing. *In UbiComp 2003 : Ubiquitous Computing*, pages 107–123. Springer, 2003. [Cité page 17.]

[6] Guido Bertoni, Joan Daemen, Michaël Peeters et Gilles Van Assche : Keccak. *In Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 313–314. Springer, 2013. [Cité page 41.]

[7] Åke Björck : Solving linear least squares problems by gram-schmidt orthogonalization. *BIT Numerical Mathematics*, 7(1):1–21, 1967. [Cité page 44.]

[8] Marina Blanton et Paolo Gasti : Secure and efficient protocols for iris and fingerprint identification. *In* Springer, éditeur : *16th European Symposium on Research in Computer Security (ESORICS)*, volume 6879 de *Lecture Notes in Computer Science*, pages 190–209, 2011. [Cité page 42.]

[9] R.M. Bolle, J.H. Connell et N.K. Ratha : Biometric perils and patches. *Pattern Recognition*, 35(12):2727–2738, 2002. [Cité page 44.]

[10] Joseph Bonneau, Cormac Jerley, Paul C. van Oorschot et Frank Stajano : The quest to replace passwords :a framework for comparative evaluation of web authentication schemes. Rapport technique, University of Cambridge, Computer Laboratory, 2012. [Cité page 23.]

[11] Bernhard E Boser, Isabelle M Guyon et Vladimir N Vapnik : A training algorithm for optimal margin classifiers. *In Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. [Cité page 46.]

[12] Patrick Bours : Continuous keystroke dynamics : A different perspective towards biometric evaluation. *Information Security Technical Report*, 17(1):36–43, 2012. [Cité pages 50 et 51.]

[13] Patrick Bours et Soumik Mondal : Performance evaluation of continuous authentication systems. *IET Biometrics*, 4(4):220–226, 2015. [Cité pages 51 et 52.]

[14] John Brainard, Ari Juels, Ronald L. Rivest, Michael Szydlo et Moti Yung : Fourth-factor authentication : Somebody you know. *In Proceedings of the 13th ACM conference on Computer and communications security*, 2006. [Cité page 17.]

[15] Pam Briggs et Patrick Olivier : Biometric daemons : Authentication via electronic pets. *In CHI '08 Extended Abstracts on Human Factors in Computing Systems*, 2008. [Cité page 23.]

[16] Julien Bringer, Hervé Chabanne, Malika Izabachène, David Pointcheval, Qiang Tang et Sébastien Zimmer : An application of the goldwasser-micali cryptosystem to biometric authentication. *In* Springer, éditeur : *12th Australasian Conference on Information Security and Privacy (ACISP)*, volume 4586 de *Lecture Notes in Computer Science*, pages 96–106, 2007. [Cité page 42.]

[17] Ines Brosso, Alessandro La Neve, Graça Bressan et Wilson Vicente Ruggiero : A continuous authentication system based on user behavior analysis. *In International Conference on Availability, Reliability and Security*, 2010. [Cité page 22.]

[18] Kim Cameron, Reinhard Posch et Kai Rannenberg : Appendix d. proposal for a common identity framework : A user-centric identity metasystem. *The Future of Identity in the Information Society*, page 477, 2009. [Cité page 16.]

[19] H. Carter, C. Lever et P. Traynor : Whitewash : Outsourcing garbled circuit generation for mobile devices. *In 30th Annual Computer Security Applications Conference (ACSAC)*, pages 266–275, 2014. [Cité page 43.]

[20] Chih-Chung Chang et Chih-Jen Lin : Libsvm : A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27 :1–27 :27, mai 2011. ISSN 2157-6904. URL http://doi.acm.org/10.1145/1961189.1961199. [Cité page 78.]

[21] Michał Choraś et Rafał Kozik : Contactless palmprint and knuckle biometrics for mobile devices. *Pattern Analysis and Applications*, 15(1):73–85, 2012. [Cité page 38.]

[22] Richard Chow, Markus Jakobsson, Ryusuke Masuoka, Jesus Molina, Yuan Niu et Zhexuan Song : Authentication in the clouds : A framework and its application to mobile users. *In Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, 2010. [Cité pages 83 et 93.]

[23] Nathan Clarke : *Advanced user Authentication for mobile devices.* Thèse de doctorat, School of Computing, Communication & Electronics, 2004. [Cité page 22.]

[24] Nathan Clarke : *Transparent User Authentication Biometrics, RFID and Behavioural Profiling.* Springer, 2011. [Cité pages 28, 38, 91, 92, et 93.]

[25] Heather Crawford : *A Framework For Continuous ,Transparent Authentication on Mobile Devices.* Thèse de doctorat, SCHOOL OF COMPUTING SCIENCE COLLEGE OF SCIENCE AND ENGINEERING UNIVERSITY OF GLASGOW, 2012. [Cité page 10.]

[26] Heather Crawford, Karen Renaud et Tim Storer : A framework for continuous, transparent mobile device authentication. *computers & security elsevier*, 2013. [Cité pages 31, 92, 93, 100, et 103.]

[27] Philippe Créhange : Orange. quand le smartphone vous reconnaîtra grâce à vos faits et gestes. Le Télégramme. URL http://www.letelegramme.fr/economie/orange-quand-le-smartphone-vous-reconnaitra-grace-a-vos-faits-et-gestes-25-04-2017-php. [Cité page 110.]

[28] David Crouse, Hu Han, Deepak Chandra, Brandon Barbello et Anil K Jain : Continuous authentication of mobile user : Fusion of face image and inertial measurement unit data. *In 2015 International Conference on Biometrics (ICB)*, pages 135–142. IEEE, 2015. [Cité page 29.]

[29] Yann Daoulas : Mobile : comment orange prépare la fin du mot de passe. Degroupnews. [Cité page 110.]

[30] Alberto de Santos-Sierra, Carmen Sanchez-Avila, Javier Guerra-Casanova et Aitor Mendaza-Ormaza : *Hand Biometrics in Mobile Devices*, chapitre 18, pages 367–382. InTech, 2011. [Cité page 38.]

[31] Mohammad Derawi et Patrick Bours : Gait and activity recognition using commercial phones. *Computers & Security*, 2013. [Cité page 30.]

[32] Ingo Deutschmann, Peder Nordstrom et Linus Nilsson : Continuous authentication using behavioral biometrics. *IT Professional*, 4(15):12–15, 2013. [Cité page 51.]

[33] Tim Dierks : The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, octobre 2015. URL https://rfc-editor.org/rfc/rfc5246.txt. [Cité page 66.]

[34] Tim Dierks et Eric Rescorla : The transport layer security (tls) protocol version 1.2. RFC 5246 (Proposed Standard), août 2008. URL http://www.ietf.org/rfc/rfc5246.txt. Updated by RFCs 5746, 5878, 6176. [Cité page 20.]

[35] Tim Dierks et Eric Rescorla : The transport layer security (tls) protocol version 1.2. RFC 5246 (Proposed Standard), août 2008. URL http://www.ietf.org/rfc/rfc5246.txt. Updated by RFCs 5746, 5878, 6176. [Cité page 63.]

[36] Josep Domingo-Ferrer, Qianhong Wu et Alberto Blanco-Justicia : Flexible and robust privacy-preserving implicit authentication. *In 29th IFIP 11 International Conference on Systems Security and Privacy Protection (SEC)*, pages 18–34, 2015. [Cité page 43.]

[37] F. Betül Durak, Thomas M. DuBuisson et David Cash : What else is revealed by order-revealing encryption ? *In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1155–1166, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4139-4. URL http://doi.acm.org/10.1145/2976749.2978379. [Cité page 43.]

[38] Andrej Dyck, Ralf Penners et Horst Lichter : Towards definitions for release engineering and devops. *In Proceedings of the Third International Workshop on Release Engineering*, pages 3–3. IEEE Press, 2015. [Cité page 107.]

[39] Nathan Eagle et Alex Pentland : Reality mining : sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006. [Cité page 101.]

[40] Delphine Escure : Orange labs présente deux innovations sur mobile. [Cité page 110.]

[41] Europe : eid interoperability for pegs. Rapport technique, iDABC European eGovernement services, 2007. [Cité pages 89 et 96.]

[42] PUB FIPS : 197, advanced encryption standard (aes), national institute of standards and technology, us department of commerce, november 2001. *Link in : http ://csrc. nist. gov/publications/fips/fips197/fips-197. pdf.* [Cité page 41.]

[43] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic et Dawn Song : Touchalytics : On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *CoRR*, abs/1207.6231, 2012. URL http://arxiv.org/abs/1207.6231. [Cité page 30.]

[44] Lex Fridman, Steven Weber, Rachel Greenstadt et Moshe Kam : Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location. *arXiv preprint arXiv :1503.08479*, 2015. [Cité pages 31, 46, et 83.]

[45] Steven Furnell, Nathan Clarke et Sevasti Karatzouni : Beyond the pin : Enhancing user authentication for mobile devices. *Computer Fraud & Security*, 2008. [Cité page 103.]

[46] Paolo Gasti, Jaroslav Sedenka, Qing Yang, Gang Zhou et Kiran S. Balagani : Secure, fast, and energy-efficient outsourced authentication for smartphones. *IEEE Trans. Information Forensics and Security*, 11(11):2556–2571, 2016. [Cité page 43.]

[47] A. Goh et D. Ngo : Computation of cryptographic keys from face biometrics. *In Communications and Multimedia Security*, pages 1–13. LNCS 2828, 2003. [Cité page 43.]

[48] Shafi Goldwasser et Silvio Micali : Probabilistic encryption and how to play mental poker keeping secret all partial information. *In Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 365–377, 1982. URL http://doi.acm.org/10.1145/800070.802212. [Cité page 61.]

[49] Shafi Goldwasser et Silvio Micali : Probabilistic encryption. *J. Comput. Syst. Sci.*, 28:270–299, 1984. URL http://dx.doi.org/10.1016/0022-0000(84)90070-9. [Cité page 61.]

[50] Marta Gomez-Barrero, Julian Fierrez, Javier Galbally, Emanuele Maiorana et Patrizio Campisi : Implementation of fixed-length template protection based on homomorphic encryption with application to signature biometrics. *In CVPR Workshops*, pages 259–266, 2016. [Cité page 42.]

[51] Government of India : e-pramaan : Framework for e-authentication. Rapport technique, Ministry of Communications and Information Technology, 2012. [Cité pages 19 et 89.]

[52] Eric Grosse et Mayank Upadhyay : Authentication at scale. *Security & Privacy, IEEE*, 11(1):15–22, 2013. [Cité page 14.]

[53] Richard P Guidorizzi : Security : Active authentication. *IT Professional*, 15(4):4–7, 2013. [Cité page 28.]

[54] Kimmo Halunen et Antti Evesti : Context-aware systems and adaptive user authentication. *Evolving Ambient Intelligence*, 2013. [Cité page 17.]

[55] Kimmo Halunen et Visa Antero Vallivaara : Secure, usable and privacy-friendly user authentication from keystroke dynamics. *In* Springer, éditeur : *21st Nordic*

Conference on secure IT systems (nordsec), volume 10014 de *Lecture Notes in Computer Sciences*, pages 256–268, 2016. [Cité page 43.]

[56] Julien Hatin, Estelle Cherrier, Jean-Jacques Schwartzmann et Christophe Rosenberger : Privacy preserving transparent mobile authentication. *In Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1 : ICISSP,*, pages 354–361, 2017. ISBN 978-989-758-209-7. [Cité pages 44 et 76.]

[57] Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong et Ian Oakley : Casa : Context-aware scalable authentication. *In SOUPS '13 Proceedings of the Ninth Symposium on Usable Privacy and Security*, 2013. [Cité page 22.]

[58] Kirsi Helkala et Einar Snekkenes : Formalizing the ranking of authentication products. *Information Management & Computer Security*, 17(1):30–43, 2009. [Cité pages 94 et 103.]

[59] Brian Hindo : At 3m, a struggle between efficiency and creativity. 2007. [Cité page 105.]

[60] Michael Httermann : *DevOps for developers*. Apress, 2012. [Cité page 106.]

[61] ISO : Information technology — security techniques — entity authentication assurance framework (iso 29115), 2013. [Cité pages 15, 16, 19, 31, et 89.]

[62] Anil K Jain, Arun Ross et Salil Prabhakar : An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14 (1):4–20, 2004. [Cité pages 49 et 50.]

[63] Markus Jakobsson, Elaine Shi, Philippe Golle et Richard Chow : Implicit authentication for mobile devices. *In HotSec'09 Proceedings of the 4th USENIX conference on Hot topics in security*, 2009. [Cité page 81.]

[64] Audun Jøsang : Identity management and trusted interaction in internet and mobile computing. *Information Security, IET*, 2013. [Cité pages 15, 16, 20, 21, 23, 88, et 89.]

[65] Audun Jøsang et Simon Pope : User centric identity management. *In AusCERT Asia Pacific Information Technology Security Conference*, 2005. [Cité pages 20 et 21.]

[66] Ronald Kainda, Ivan Flechais et AW Roscoe : Security and usability : Analysis and evaluation. *In Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 275–282. IEEE, 2010. [Cité page 26.]

[67] Burt Kaliski : PKCS #5 : Password-Based Cryptography Specification Version 2.0. RFC 2898, septembre 2000. URL https://rfc-editor.org/rfc/rfc2898.txt. [Cité page 56.]

[68] Jin-Suk Kang : Mobile iris recognition systems : An emerging biometric technology. *Procedia Computer Science*, 1(1):475–484, 2010. [Cité page 38.]

[69] Hilmi Günes Kayacik, Mike Just, Lynne Baillie, David Aspinall et Nicholas Micallef : Data driven authentication : On the effectiveness of user behaviour modelling with mobile device sensors. *CoRR*, abs/1410.7743, 2014. URL http://arxiv.org/abs/1410.7743. [Cité pages 31 et 45.]

[70] Hassan Khan, Aaron Atwater et Urs Hengartner : A comparative evaluation of implicit authentication schemes. *Research in Attacks, Intrusions and Defenses*, 2014. URL https://crysp.uwaterloo.ca/software/ia/. [Cité pages 22 et 30.]

[71] Hassan Khan, Aaron Atwater et Urs Hengartner : Itus : an implicit authentication framework for android. *In MobiCom '14 Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014. [Cité page 31.]

[72] Hassan Khan et Urs Hengartner : Towards application-centric implicit authentication on smartphones. *In HotMobile '14 Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, 2014. [Cité page 22.]

[73] Elie Khoury, Bostjan Vesnicer, Javier Franco-Pedroso, Ricardo Violato, Z Boulkcnafet, Luis-Miguel Mazaira Fernandez, Mireia Diez, Justina Kosmala, Houssemeddine Khemiri, Tomas Cipr *et al.* : The 2013 speaker recognition evaluation in mobile environment. *In Biometrics (ICB), 2013 International Conference on*, pages 1–8. IEEE, 2013. [Cité page 38.]

[74] Wei-Han Lee et Ruby Lee : Implicit smartphone user authentication with sensors and contextual machine learning. *In International Conference on Dependable Systems and Networks*, 2017. [Cité page 31.]

[75] Fudong Li, Nathan Clarke, Maria Papadaki et Paul Dowland : Behaviour profiling on mobile devices. *In 2010 International Conference on Emerging Security Technologies*, 2010. [Cité page 45.]

[76] Fudong Li, Nathan Clarke, Maria Papadaki et Paul Dowland : Behaviour profiling for transparent authentication for mobile devices. *In 10th European Conference on Information Warfare and Security*, 2011. [Cité pages 22, 31, 68, 71, 76, et 90.]

[77] Fudong Li, Nathan Clarke, Maria Papadaki et Paul Dowland : Active authentication for mobile devices utilising behaviour profiling. *International Journal of Information Security*, 2013. [Cité pages 31, 48, 79, et 83.]

[78] Stephen Paul Marsh : *Formalising Trust as a Computational Concept.* Thèse de doctorat, University of Stirling, Department of Science Computing and Mathematics, 1994. [Cité pages 15 et 90.]

[79] MASTERCARD : Mastercard securecode merchant implementation guide. 2011. [Cité page 18.]

[80] Alfred J MENEZES, Paul C VAN OORSCHOT et Scott A VANSTONE : *Handbook of Applied Cryptography*. Discrete Mathematics and Its Applications. CRC Press, 2010. ISBN 9781439821916. [Cité page 26.]

[81] Nicolas MICALLEF, Hilmi Günes KAYACIK, Mike JUST, Lynne BAILLIE et David ASPINALL : Sensor use and usefulness : Trade-offs for data-driven authentication on mobile devices. *Not yet published*, 2015. [Cité page 41.]

[82] Sudipta MONDAL et Patrick BOURS : Continuous authentication using mouse dynamics. *In Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*, pages 1–12. IEEE, 2013. [Cité pages 50 et 73.]

[83] Tarik MUSTAFIĆ, Arik MESSERMAN, Seyit Ahmet CAMTEPE, Aubrey-Derrick SCHMIDT et Sahin ALBAYRAK : Behavioral biometrics for persistent single sign-on. *In Proceedings of the 7th ACM workshop on Digital identity management*, pages 73–82. ACM, 2011. [Cité page 93.]

[84] Abhijit Kumar NAG, Dipankar DASGUPTA et Kalyanmoy DEB : An adaptive approach for active multi-factor authentication. *In 9th Annual Symposium on Information Assurance (ASIA'14)*, page 39, 2014. [Cité page 103.]

[85] Tempestt J NEAL, Damon L WOODARD et Aaron D STRIEGEL : Mobile device application, bluetooth, and wi-fi usage data as behavioral biometric traits. *In Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*, pages 1–6. IEEE, 2015. [Cité page 31.]

[86] Jakob NIELSEN : *Usability engineering*. Elsevier, 1994. [Cité page 27.]

[87] FIPS PUB NIST : 180-1 : Secure hash standard, 1995. [Cité page 41.]

[88] OASIS : Glossary for the oasis security assertion markup language (saml) v2.0, 2005. [Cité page 15.]

[89] Lawrence O'GORMAN : Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040, 2003. [Cité pages 50 et 94.]

[90] Margarita OSADCHY, Benny PINKAS, Ayman JARROUS et Boaz MOSKOVICH : SCiFI - a system for secure face identification. *In 31st IEEE Symposium on Security and Privacy (S&P)*, pages 239–254, 2010. [Cité page 42.]

[91] V. M. Patel, R. Chellappa, D. Chandra et B. Barbello : Continuous user authentication on mobile devices : Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, July 2016. ISSN 1053-5888. [Cité page 41.]

[92] Sean Peisert, Ed Talbot et Tom Kroeger : Principles of authentication. *In Proceedings of the 2013 workshop on New security paradigms workshop*, pages 47–56. ACM, 2013. [Cité pages 94 et 103.]

[93] Andreas Pfitzmann et Marit Hansen : A terminology for talking about privacy by data minimization : Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, 2010. [Cité pages 25 et 40.]

[94] Anand Ranganathan, Jalal Al-Muhtadi et Roy H Campbell : Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2):62–70, 2004. [Cité page 91.]

[95] N.K. Ratha, J.H. Connell et R. Bolle : Enhancing security and privacy in biometrics-based authentication system. *IBM Systems J.*, 37(11):2245–2255, 2001. [Cité page 44.]

[96] C. Rathgeb et A. Uhl : A survey on biometric cryptosystems and cancelable biometrics. *EURASIP J. on Information Security*, 3, 2011. [Cité page 43.]

[97] Oriana Riva, Chuan Qin, Karin Strauss et Dimitrios Lymberopoulos : Progressive authentication : Deciding when to authenticate on mobile phones. *In USENIX Security Symposium*, pages 301–316, 2012. URL https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/riva. [Cité page 22.]

[98] Cristiano C Rocha, Joao Carlos D Lima, MAR Dantas et Iara Augustin : A2best : An adaptive authentication service based on mobile user's behavior and spatio-temporal context. *In Computers and Communications (ISCC), 2011 IEEE Symposium on*, pages 771–774. IEEE, 2011. [Cité page 22.]

[99] République Française : Référentiel général de sécurité, 2010. [Cité page 14.]

[100] Napa Sae-Bae et Markus Jakobsson : Hand authentication on multi-touch tablets. *In Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, page 8. ACM, 2014. [Cité page 38.]

[101] Hataichanok Saevanee, Nathan Clarke, Steven Furnell et Valerio Biscione : Text-based active authentication for mobile devices. *In ICT Systems Security and Privacy Protection*, pages 99–112. Springer, 2014. [Cité pages 31 et 83.]

[102] Hataichanok Saevanee, Nathan L Clarke et Steven M Furnell : Multi-modal behavioural biometric authentication for mobile devices. *In Information Security and Privacy Research*, pages 465–474. Springer, 2012. [Cité page 22.]

[103] Nashad Ahmed Safa, Reihaneh Safavi-Naini et Siamak F Shahandashti : Privacy-preserving implicit authentication. *In IFIP International Information Security Conference*, pages 471–484. Springer, 2014. [Cité page 83.]

[104] Nashad Ahmed Safa, Reihaneh Safavi-Naini et Siamak Fayyaz Shahandashti : Privacy-preserving implicit authentication. *In 29th IFIP 11 International Conference on Systems Security and Privacy Protection (SEC)*, pages 471–484, 2014. [Cité page 43.]

[105] Glenn Shafer *et al.* : *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976. [Cité pages 97, 98, et 100.]

[106] C. E. Shannon et W. Weaver : *The Mathematical Theory of Information*. University of Illinois Press, Urbana, Illinois, 1949. [Cité page 78.]

[107] Elaine Shi, Yuan Niu, Markus Jakobsson et Richard Chow : Implicit authentication through learning user behavior. *In Information Security*, pages 99–113. Springer, 2011. [Cité pages 22, 23, et 28.]

[108] Sophos : Mobile usage. https://www.sophos.com/en-us/press-office/press-releases/2013/03/mobile-security-survey.aspx. [Online ; accessed 10-July-2016]. [Cité page 55.]

[109] Alfred Spector, Peter Norvig et Slav Petrov : Google's hybrid approach to research. *Communications of the ACM*, 55(7):34–37, 2012. [Cité page 106.]

[110] Zahid Syed, Sean Banerjee et Bojan Cukic : Continual authentication. *Biometric Technology Today*, 2014. [Cité page 23.]

[111] Mohammad Tanviruzzaman : *Towards Usable End-User Authentcation*. Thèse de doctorat, Faculty of the Graduate School, Marquette University, 2014. [Cité page 23.]

[112] Mohammad Tanviruzzaman et Sheikh Iqbal Ahamed : Your phone knows you : Almost transparent authentication for smartphones. *In Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual*, pages 374–383. IEEE, 2014. [Cité pages 23, 76, et 83.]

[113] A.B.J. Teoh, D. Ngo et A. Goh : Biohashing : two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition*, 40, 2004. [Cité page 43.]

[114] Juan Ramón Troncoso-Pastoriza, Daniel González-Jiménez et Fernando Pérez-González : Fully private noninteractive face verification. *IEEE Transactions on Information Forensics and Security*, 8(7):1101–1114, 2013. [Cité page 42.]

[115] United State gouvernement : Electronic authentication guideline. Rapport technique, NIST, 2006. [Cité pages 19 et 89.]

[116] U.S. Department of State Foreign Affairs : The privacy act and personally identifiable information, 2010. [Cité page 24.]

[117] Benoit Vibert, Zhigang Yao, Sylvain Vernois, Jean-Marie Le Bars, Christophe Charrier et Christophe Rosenberger : Evabio a new modular platform to evaluate biometric system. *In International Conference on Information Systems Security and Privacy*, pages 234–250. Springer, 2015. [Cité page 110.]

[118] Rui Wang, Shuo Chen et XiaoFeng Wang : Signing me onto your accounts through facebook and google : A traffic-guided security study of commercially deployed single-sign-on web services. *In Security and Privacy (SP), 2012 IEEE Symposium on*, pages 365–379. IEEE, 2012. [Cité page 21.]

[119] Heiko Witte, Christian Rathgeb et Christoph Busch : Context-aware mobile biometric authentication based on support vector machines. *In Emerging Security Technologies (EST), 2013 Fourth International Conference on*, pages 29–32. IEEE, 2013. [Cité page 22.]

[120] Jiangchuan Zheng et Lionel M Ni : An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data. *In Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 153–162. ACM, 2012. [Cité page 96.]

Titre : Evaluation de la confiance dans un processus d'authentification
Mots clés : Authentification Transparente, Confiance, Protection de la vie privée

Résumé : Dans notre quotidien, le smartphone est devenu un outil indispensable pour effectuer nos tâches courantes. Accéder à des services en ligne depuis son téléphone mobile est devenu une action commune. Afin de s'authentifier à ces services parfois sensibles, la seule protection est généralement l'usage d'un mot de passe. Ces mots de passe pour être robustes doivent être de plus en plus longs. Ceci représente, sur les téléphones mobiles, une contrainte plus forte que pour les ordinateurs de bureau puisque les claviers tactiles disposent de moins de touches. D'autres méthodes d'authentification ont vu le jour sur téléphones mobiles comme la reconnaissance faciale sur les appareils android ou bien l'empreinte digitale qui gagne le marché des smartphones et même le domaine bancaire avec Apple Pay.

Afin de simplifier l'authentification, la biométrie prend une part de plus en plus importante dans l'usage des téléphones mobiles. Au delà des capteurs dédiés à l'acquisition de données biométriques, il est aussi possible d'utiliser l'environnement du téléphone mobile pour authentifier les utilisateurs. Si les méthodes d'authentification tendent à se transformer pour devenir de plus en plus transparentes, cela amène deux questions :

- Comment utiliser ces nouvelles techniques d'authentification dans les processus actuels d'authentification ?
- Quels impacts ces nouvelles méthodes peuvent avoir sur la vie privée des utilisateurs ?

L'objectif de cette thèse est de proposer des méthodes d'authentification transparentes qui soient respectueuses de la vie privée des utilisateurs tout en permettant leur intégration dans les systèmes actuels d'authentification.

Dans le manuscrit de thèse, nous abordons ces deux questions en analysant tout d'abord les travaux existants sur la collecte des données permettant l'authentification sur téléphone mobile. Puis, une fois les données collectées, nous verrons les processus permettant la mise en place d'une authentification respectueuse de la vie privée. Enfin, nous évaluons concrètement ces méthodes d'authentification par la réalisation de prototypes à l'échelle industrielle.

Abstract: In our daily life, the smartphone became an unavoidable tool to perform our common tasks.
Accessing to online services from its mobile phone is an usual action.
In order to authenticate to those services, that might be sensitive, the one and only protection is usually a password.
Those passwords must be longer and longer to stay robust.
This is a bigger constraint on mobile phones than on desktop computers.
Other authentication solutions are dedicated to smartphones, like facial recognition on android and now Apple smartphones or the fingerprint that conquier new phones.

To ease the authentication process, biometrics is more and more often used on mobile phones. In addition to the dedicated biometric sensors, it is also possible to use the phone environement to authenticate users.
However, if authentication methods are becoming more and more transparent, it brings two questions:

- How to integrate those new methods within the actual authentication framework ?
- What is the impact of those new methods on users' privacy ?

The main goal of the phD is to offers privacy compliant transparent authentication methods while integrating them in current authentication systems.

In this document, we evaluates those two questions by first analyzing existing works on the data collection for transparent authentication on mobile phones. Then, once the data are collected, we will see wich process can enable the privacy protection. To conclude, we will evaluates concretly those solutions by building industrial prototypes.