Contributions to keystroke dynamics for privacy and security on the Internet

Denis Migdal

To cite this version:

Denis Migdal. Contributions to keystroke dynamics for privacy and security on the Internet. Cryptography and Security [cs.CR]. Normandie Université, 2019. English. NNT : 2019NORMC253 . tel-02518436

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'Université de Caen Normandie

Contributions to Keystroke Dynamics for Privacy and Security on the Internet

Présentée et soutenue par
Denis MIGDAL

Thèse soutenue publiquement le 26/11/2019 devant le jury composé de		
M. PATRICK BOURS	Professeur des universités, Norwegian University of Science and Tech	Rapporteur du jury
M. STEVE FURNELL	Professeur des universités, Université de Plymouth - Royaume Uni	Rapporteur du jury
M. JEAN-YVES RAMEL	Professeur des universités, Université de Tours François Rabelais	Rapporteur du jury
M. AUDUN JOSANG	Professeur des universités, Université d'Oslo - Norvège	Membre du jury
M. OLIVIER LEZORAY	Professeur des universités, Université Caen Normandie	Président du jury

Thèse dirigée par CHRISTOPHE ROSENBERGER, Groupe de recherche en informatique, image, automatique et instrumentation

Forewords

I first met Denis Migdal when he came to the Informatics Depatrment at UiO (The University of Oslo) in May 2015 to spend three months as an internship Master's student. At that time UiO and ENSCAEN collaborated on the OffPAD project (Offline Personal Authentication Device), which was a Eurostars project. The task given to Denis focused on developing and implementing solutions for IdM (Identity Management) in the context of the OffPAD device and architecture. Denis did very valuable work for the OffPAD project, so we were extremely happy to see Denis return to UiO for a 6-month internship from March to September in 2016. During this internship, Denis developed a proxy protocol for secure communication between the OffPAD device, the client device and the server. This work resulted in our co-authored paper: Offline Trusted Device and Proxy Architecture based on a new TLS Switching Technique, (Johansen, Jøsang and Migdal), presented at the International Workshop on Secure Internet of Things (SIoT 2017). Oslo, September 2017. After completing his Master's the same year, Denis started as a PhD-student at ENSICAEN and the University of Caen. His PhD research project has focused on biometric keystroke dynamics and methods to strengthen online trust and privacy. We were again extremely happy when Denis decided to return to UiO during August and September 2019 to write up his thesis. He has been part of the Research Group on Information and Cyber Security where he already knew many people from his internships in 2015 and 2016. Denis gave a talk in the Academic Forum on Security on 26 August 2019. The title of his talk was Privacy Threat of Keystroke Profiling on the Web. Many people came to hear his talk, including representatives from the The Norwegian Data Protection Authority.

I am very pleased to see the completion of the PhD thesis of Denis Migdal. It has been exciting to follow his work as a Master's student and as a PhD student, and it is great to see that he has matured to become a researcher with great analytic skills, practical talent to make things work, and a strong capacity to focus and to carry out tasks to successful completion. I wish Denis Migdal all the best for his career.

Oslo, 7 October 2019,
Audun Jøsang

Preface

”We do not research for a living, we are living for research.”

Motivation

This thesis has been the result of my work at the GREYC laboratory where I was asked to bring trust to online transactions with a focus on Keystroke Dynamics.

”Trust in online transactions” is a very broad concept that requires to define the actors involved and the environment in which such interactions take places. From this definition we extracted problematic, some of which we answer in this thesis.

I took the approach of considering the problematic in all its facets, and to provide a global answer, leading to the form of this thesis. This thesis addresses several problematic with contributions in various sub-domains, leading to a complete privacy-compliant authentication scheme.

This thesis focuses on Keystroke Dynamics. Keystroke Dynamics is one of the speciality of the GREYC laboratory with several thesis conducted in this domain [Mhenni et al., 2019, Idrus et al., 2013, Giot, 2012]. It enables profiling of users through their way of typing on a Keyboard. This is a very interesting modality as it does not require neither additional sensors nor additional actions from the user. This is thus a costless and transparent biometric modality, which explains its growing popularity.

However little to no work has been made on the Keystroke Dynamics anonymization, synthetic generation, or usage with BioHashing[1], which this thesis address.

While reading this thesis, I hope you will find as much as pleasure I had while writing it.

[1] a biometric ”hash” used in this thesis. It is also one of the speciality of the GREYC laboratory which made several contributions in this domain [Atighehchi et al., 2019, Ninassi et al., 2017, Plateaux et al., 2014, Barbier and Rosenberger, 2014, Lacharme et al., 2013]

Plan

This thesis in organized as follows:

In a first chapter, we explore the Web environment and its issues to then define the problematic and scope of this thesis.

The second chapter presents information that a Web Server could easily access, and how they pose a threat to user privacy. The third chapter will then present ways of protecting such information against non-consented collection. The forth chapter later proposes a mean of using such information in a multi-modal and privacy-compliant authentication scheme. These chapters focus on Keystroke Dynamics.

The fifth chapter tackles the problem of keystroke modelisation, necessary to evaluate and improve the performances of the contributions presented in the previous chapters.

The sixth and last chapter present some applications of our contributions, as well as their usage with trusted devices in order to increase security and privacy.

Acknowledgements

I would like to take the opportunity to thanks the persons that contributed or supported this work.

I am thankful to the GREYC Laboratory and the Normandy region for the thesis opportunity, as well as the University of Oslo and the University of Reggio Calabria for welcoming me both 1.5 months.

A special thanks to my PhD director, Christophe Rosenberger, who made this thesis possible. I cannot be grateful enough for your unconditional availability when the deadlines were coming close. You like challenges, but I hope I wasn't too much of one.

A special thanks also to Francesco Buccafurri who supervised me for a 1.5 months research collaboration during my PhD thesis. I am looking forward to future collaborations. A third special thanks to Audun Jøsang who supervised me for 2 master degree internships and a 1.5 months research collaboration during my PhD thesis. May the magic chocolate box never come empty.

I also thanks Patrick Bours, Steve Furnell, and Jean-Yves Ramel who accepted to review this PhD manuscript, as well as Audun Jøsang, and Olivier Lezoray for joigning them in my PhD jury.

I truly appreciate working with all my colleagues and co-authors, especially with the GREYC E-Payment & Biometrics team, where I really enjoyed working in such a pleasant atmosphere. I thanks them along with all the OffPAD project members with whom I working during my 2 internships. I do not forget to thank the secretariats, the doctoral school MIIS, and the sysadmins that helped and assisted me during this thesis, as well as all bachelors, master, and engineering students with whom I collaborated through research projects.

I am also grateful to all my family, who supported and helped me during this thesis, although it was not easy to regularly see each other during this thesis. I also take the opportunity to wish the best academic successes to of my little cousins who grew so fast.

Contents

Chapter 1

Thesis position

In this chapter, we present the Internet social environment, i.e. the involved actors, their motivations and issues. We then discuss the thesis scope, objectives, and the issues it addresses.

Keywords: *Internet; Internet actors; Internet issues.*

Contents

1.1 Environment: The Internet and its actors

The Internet is not only a set of techniques and computers, but is also a set of actors with different interests and motivations. The deployed technical solutions are not an end in themselves, but serve actors needs. Understanding this environment, actors interests, needs and motivations, as well as the conflicts arising from divergences in actors interests, are thus required to the conception of suitable technical solutions.

In the following, we thus quickly present Internet actors and explore their aims and motivations. In the next section, we discuss issues inherent to the Internet, to then establish the scope of this PhD thesis. Still, actors presentation remains superficial as deep understanding of the ins and outs of actors interactions requires strong interdisciplinary knowledge mainly in the fields of economics (s.a. game theory), psychology, sociology, and legal science.

1.1.1 Internet actors

The Internet is defined by the Cambridge dictionary as a ”wordwide system of computer networks used to exchange information” [1], which highlights the three main components of the Internet:

- ***computers***, constituting a physical network (of networks);
- ***users***, physical or juridical persons interacting on the Internet;
- ***information***, that are exchanged.

Computers do not act on their own, but on behalf of Internet users. They are tools instrumented by users for them to pursue their needs. Computers are not legal persons, and thus cannot be held responsible of their actions. However, users are accountable of the actions executed by computers on their behalf.

We distinguish between users (*Service Provider*) instrumenting computers (*server*) in order to provide services, and other users (*users*) accessing such services through other computers (*clients*). these roles are non-exclusive, e.g. a computer can be, depending on the context either, a client, a server, or both.

As illustrated in Figure 1.1, we consider the following non-exclusive roles played by Internet actors:

- ***Author***, origin/creator of information;
- ***Subject***, what the information is about;
- ***Viewer***, consumes information;
- ***Service Provider***, hosts and distribute information;
- ***Moderators***, regulate information by enforcing policies;
- ***Society***, defines the policies that should be enforced.

Authors create contributions containing information about a subject. They propose their contributions to Service Providers (SP), which in turn propose them to viewers

[1] `https://dictionary.cambridge.org/dictionary/english-french/internet?q=Internet`

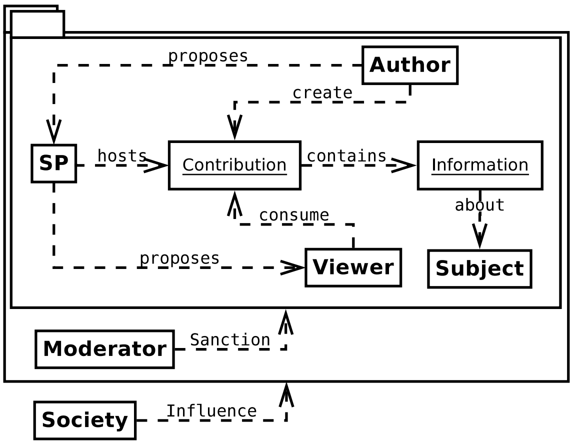


Figure 1.1: Internet actors

for consumption. These interactions are enforced by moderators in a framework influenced and defined by the society. These roles are presented and detailed in the following.

In short: *Internet actors play several non-exclusives roles: they can be author/viewer/subject of information, a provider of service (Service Provider), a moderator, or/and be simply part of the society.*

1.1.2 Subjects

Subjects are what exchanged information is about. Their aim is to control their image and reputation, i.e. how viewers perceive them through information consumption. For a physical or juridical person, this image constitutes their online identity.

Subjects aim at different kinds of interactions depending on the context. For each, they have a different image and identity [al, 2014, Boyd, 2014]. For example, in a professional context, a subject is likely to aim at a more serious and professional image than in a family context. In order to maintain their images, and thus the way they interact with others, subjects aim, to some extends, at controlling information about themselves, i.e. who have access to what, independently of the information truthfulness.

Tristan Nitot[2] identifies in [Nitot, 2016, p.32-37] five causes of information leakage or misuse:

- An entity might voluntary denounce an user;
- An employee might exceed/overstep its functions/duties;
- A server/computer might be hacked;

[2] Founder of Mozilla Europe, member (2013-2015) of the *National Council of the Digital Technology*, a French consultative state institution, and member of the forecasting comity of the French CNIL since 2015 (`https://www.cnil.fr/fr/les-membres-du-comite-de-la-prospective`)

- An entity (e.g. state agency like NSA) might spy on another;
- Authors himself might unintentionally leak information.

A sixth reason can be added: an entity might sell or share information to another for financial gains [Fox et al., 2000].

Even through, in itself, information, or the knowledge of information, are not harmful in anyway, their usage and consequences, might be. Uncontrolled information thus constitute serious threats for subjects:

- Subjects *image* deterioration impacts the way they interact with others, resulting, e.g. to a loss of trust, credibility, or influence (e.g. boycott of a company, or even of a physical person).
- Subjects might *loose power*, e.g. negotiation power (e.g. for banking or insurance fees).
- Another entity might *gain power over* subjects, e.g. blackmailing, extortion, social engineering.
- Subjects might be *attacked*, s.a. thief, identity usurpation, abduction.
- Subjects might be *sanctioned* by moderators, s.a. banned from using a service, depraved of Internet by legal responsible, fined or jailed by state, fired from a job [Fox et al., 2000].
- Subjects might be *sanctioned* by the society, e.g. slanders, harassment, physical threats.
- The *fear* of the consequences above may lead the subject to self-censure or, if sensitive information are leaked, to suicide.

However, controlling information on the Internet is quite hard. First, it is difficult for a subject to be conscious of every disclosed information, as well as all the ins and outs of such disclosures, at short and long terms. Moreover, its identities are very likely to overlap, and might then be linked by third parties, i.e. asserting that they belong to the same person. Identities linkability can lead to a ”collision” of incompatible contexts [Boyd, 2014], i.e. contexts where the subject maintains contradictory images.

Secondly, information can be easily retrieved on the Internet through the use of either standard search engines, or more advanced/dedicated tools. Such tools enable third parties to easily find needles in the humongous haystack that is the Internet. Thus, an information publicly accessible on the Internet is to be assumed known by all.

Finally, information can hardly be fully removed from the Internet as they are easily shared and copied. Any attempt to force removal of a given information is likely to backfire through the Streigand effect [Jansen and Martin, 2015]. Information can also be archived by the SP itself (even if the information has been officially ”deleted”) or by specialized websites s.a. `https://archive.org/web/`. Thus, any information known at a time, is likely to be known afterwards. In [Nitot, 2016], Tristan Nitot recommends the POSSE approach[3] to increase subjects control on the

[3] Publish on your Own Site Syndicate Elsewhere - `https://indieweb.org/POSSE`

information they disclose. However, this does not prevent third parties to copy such information on other websites.

In the light of the above, subjects information control mainly remains on the choice to themselves disclose information.

In short: *Subjects need to control, in some extends, the information about themselves, as they might pose serious threats to them s.a. blackmailing, boycott, identity usurpation, harassment. However, this control is limited as information can easily be found on the Internet, and can hardly be deleted.*

1.1.3 Authors

Authors are physical or juridical person producing information, seeking it to be consumed by a given set of viewers, for a given usage.

However, they can hardly ensure that the created information will be consumed the way they intended, by the viewers they intended. Indeed, once the information is known, viewers and Service Providers are able to share it, or to perform arbitrary computations on their clients (or servers), without authors knowledge and consent. Moreover, any created information also contains scrap information, s.a. authors pseudonym, a timestamp, IP address, a lexical vocabulary used. Some of these give information about the author, which is thus, at the same time, both author and subject. In such cases, authors need, before each creation of information to weight up their interests as authors and as subjects.

As already seen in the previous section, it is quite difficult to fully understand all the ins and outs of information disclosures. Besides, it is as difficult for authors to know the exact information they disclose by contributing. Indeed, even though Service Providers provides Terms of Services (ToS), not all authors are really aware of the service ins and outs, and the ToS might not fully match the reality of the collects and processing made by the Service Provider.

In short: *Authors produce information in order to make it consumed by a given set of viewers. However, as seen in the previous section, information can hardly be controlled. Also, authors are subject of scrap information without necessary being fully aware of it.*

1.1.4 Viewer

Viewers are physical or juridical persons who aim to access and consume information they want, in the way they want, possibly, in order to, or leading to, take action, make a decision, or make up their mind on a given subject.

However, viewers are physically unable to consume all information created in a year. Indeed, in 2014, 300 hours of video were uploaded *per minutes* on Youtube[4], and 220,330 books where published by IPA members[5], i.e. one book every 6 minutes. In 2006, near 1,350,000 research papers were published according to [Bjork et al., 2009], i.e. one paper every 24 seconds. Requiring viewers to select which information they wish, or not, to consume.

Viewers select information under their own criteria, e.g. perceived quality, truthfulness, subjects, popularity, reputation. Depending on the current context, some of information might be as well unwanted by viewers, s.a. spam, porn, off topic, spoils, advertisement. However, it might be difficult for viewers to find what they seek to consume [Cordier, 2015], and to evaluate information quality and relevance before actually consuming it. In a sense, some information might deceive users into consuming it.

The quality, truthfulness, and trust viewers place into information are mandatory to the proper consumption of the information. In case of incomplete, deformed, or false information, viewers might take unlighted decisions, leading to tragic consequences, s.a. financial ruin, absence of medical health, breach of trust.

In short: *Viewers aim to access and consume information. With the plethora of information on the Internet, viewers have to select the information they wish to consume, to the risk of being deceived.*

1.1.5 Service Provider

Service Providers make the link between authors and viewers by distributing information. Their aim is often financial, e.g. to make profit from the distribution of information, but can also be ideologically-driven, and centered around its own values.

Service Providers main issue is to decide how viewers and authors should interact, in particular which information should be highlighted or/and distributed to whom. The Service Providers thus have to build an internal policy that will be translated e.g. into Terms of Services, underlying processes (s.a. automatic censorship of certain type of words, highlighting popular information), the service structure and algorithms, resulting into a framework for authors-viewers interactions.

Building such framework is far from trivial. Obviously, in order to be used, the service must fulfill the needs of its users, i.e. authors and viewers. Moreover, Service Provider might be, depending on the legislation, partly responsible of the information

[4] `https://www.cnet.com/news/youtube-music-key-googles-stab-at-taking-paid-streaming-songs-mainstream/`

[5] `http://www.internationalpublishers.org/images/annual-reports/ipa_ar_online.pdf`, page 17

they host and distribute. Such legal constraints are even more important that service provider are often well-known, without the possibility of being fully anonymous, contrary to viewers or authors that are usually at least pseudonymous on the Internet, thus, making more difficult for service providers to escape legal sanctions.

Such framework should also match its users ethics and societies morals to not be disapproved and condemn e.g. with a boycott. Information hosted and shared by the Service Provider will influence its image, e.g. by association, or by thinking the SP endorse hosted information. All of this, of course, without losing sight of their bests interests.

In short: *Service Providers distribute information from authors to viewers to meet their own interests and goals. However, the interaction framework built by the Service Providers have to meet its users, as well as moderators and society, wishes.*

1.1.6 Society

Society defines which consumptions are lawful, "moral" or "ethic", although several opposed views might be expressed on some moral/ethic questions. Answers are often balanced and shaded depending on the exact context and situation [Allen, 1996]. If rules or laws are not recognized as legitimate and fair, they are likely to be transgressed or not enforced.

Several laws regulate online interactions, we focus on this section on European privacy laws. In Europe, the right of privacy is declared by:

- article 12 of the Universal Declaration of Human Rights (1948):
  "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks."[6]
- article 8 of the European Convention on Human Rights (1953):
  "1: Everyone has the right to respect for his private and family life, his home and his correspondence.
  2. There shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others."[7]
- articles 7 and 8 of the Charter of Fundamental Rights of the European Union (2000):
  Article 7. Respect for private and family life. Everyone has the right to respect for his or her private and family life, home and communications.

[6] https://www.un.org/en/universal-declaration-human-rights/index.html

[7] https://www.echr.coe.int/Pages/home.aspx?p=basictexts

> *Article 8. Protection of personal data 1. Everyone has the right to the protection of personal data concerning him or her. 2. Such data must be processed fairly for specified purposes and on the basis of the consent of the person concerned or some other legitimate basis laid down by law. Everyone has the right of access to data which has been collected concerning him or her, and the right to have it rectified. 3. Compliance with these rules shall be subject to control by an independent authority.*[8]

Privacy has been an issue at the early stage of the history of computers and the Internet, and is still relevant today. Privacy laws were previously defined in France by the "Loi informatique et liberté" (1978), modified by decrees in 1991 and in 2004. This law inspired the European *Convention for the protection of individuals with regard to the processing of personal data*[9] (1981), as well as the *Data Protection Directive*[10] (1995). Privacy laws are now defined at the European level by the *General Data Protection Regulation* (GDPR) which came into effect the 28th of May 2018, during this thesis.

The GDPR requires SP to collect users explicit and positive consent before processing their personal data. In consequences, opt-out options cannot be used anymore to collect users consent. Terms of Service (ToS) are also required to be understandable by anyone, and inform users on the performed personal data processing. ToS were often not adapted to neophyte users who just want to use services quickly, and often used technical or juridical terms incomprehensible by lambda users, and discouraging users from reading them with their length [Nitot, 2016, page 86]. ToS were aimed more to protect the SP juridically than to inform users on the processing carried out.

GDPR also requires privacy by design as well as an obligation of being secure in order to protect the processed personal data. It thus requires to bring new technical solutions to enhance privacy and security on the Internet. GDPR application is extraterritorial, meaning that any entity can be held liable of processing implying personal data of an European citizen. However, enforcement of such laws is difficult in an international context, e.g. for the right to be forgotten. European users might indeed ask search engines to remove results concerning them, but they are only removed from the website versions intended for the European public[11].

In short: *Society defines laws, moral, and ethics. The new European GDRP regulation defines a legal frame to protect users privacy.*

[8] https://www.europarl.europa.eu/charter/pdf/text_en.pdf

[9] https://www.coe.int/en/web/data-protection/convention108/modernised

[10] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:31995L0046

[11] http://curia.europa.eu/juris/document/document.jsf?docid=218105&text=&dir=&doclang=EN&part=1&occ=first&mode=DOC&pageIndex=0&cid=4477289

1.1.7 Moderators

Moderators are physical or juridical persons enforcing rules or/and laws. They can be e.g. the state, a child parents, employer, or the SP moderation service, enforcing rules at the level of a state, a house, a company, or a website. Their responsibility can be defined by laws (e.g. parental responsibilities) or/and by contracts (e.g. ToS).

To happen, entities interactions require a framework to structure such interactions, and enable trust between entities. This framework weight the rights of each entity by establishing boundaries, a.k.a. rules and laws. The role of moderators is to enforce such framework, and to sanction transgressors, s.a. with fines, deprivation of rights, temporary/permanent bans from the service. Moderators have the power to censor and control what viewers can see, for legitimate purposes (e.g. unsuitable content for minors) or not (s.a. driven by political, religious or philosophical ideologies). Giving too much power to moderators might lead to arbitrary law enforcement and abuses, thus being detrimental to users rights, *"Quis custodiet ipsos custodes?*[12] *", Juvenal.*

For example, anonymity and pseudonymity enable authors to escape from the consequences of their actions. This can be desirable, as anonymity allows whistle-blowers, journalistic sources, witnesses, and political dissidents to express themselves without fearing repercussions. But, at the same time, this can be undesirable, as it also enables people to transgress laws, ethics, or moral with impunity. Indeed, e.g. suing an user for its misbehavior requires for the plaintiff, or the moderators, to know the misbehaving user real name and address.

In general, democratic systems cannot exists without secrecy, hence the Human Right of privacy. Indeed, voting processes often require ballot secrecy to prevent vote coercion, vote buying, or other electoral frauds. Secrecy of judicial inquiries are also required to the proper operation of Justice. A too strong law enforcement also deprives citizens from their Human Right of revolution. A too strong surveillance also deprives citizens from their freedom of expression and freedom of thought, due to the panoptic effect [Simon, 2005], where citizens censor themselves, and change their behaviors, when feeling observed or monitored. Other secrecy are also required in order to prevent illegitimate discrimination based, e.g. on the user medical state, sexual orientation, or religion.

Although the private sphere must be protected in order to guarantee citizens rights, moderators still need to enter the private sphere to sanction serious law infringement that also threaten other citizen rights. Moderators thus need a special access to information, but only in the scope of a procedure framed by laws, with safeguards enabling to preserve users fundamental rights.

This has for consequence that moderators often do not have the power to fully enforce the rules and laws, for example, by resorting to users reports of inappropriate

[12] "Who will [moderate] the [moderators] ?"

content or behaviour. This enables to reduce the resources assigned to services monitoring, but on an other hand, unreported misbehaviors remain unsanctioned.

In short: *Moderators enforce rules to enable interactions between users. However, they often lacks of the means to properly enforce such rules, partially to prevent abuses from moderators themselves.*

1.2 Security and privacy

Security and privacy are often presented as opposed concepts, security protecting the system against users, and privacy protecting users against the system, thus, leading to a false-dilemma, asking system designers to choose between security and privacy [Schneier, 2001].

Security ensures the proper operation of the system that must remain *available* and *efficient* in order to serve its users. The system must ensure *authenticity* (i.e. truthfulness of a claim) of user identity (e.g. to grant access to resources), of messages *origin* and *integrity* (i.e. the message has been sent by the claimed user, and has not been modified by another), as well as the authenticity of information or user attributes (e.g. the truthfulness of a statement s.a. users claimed age or gender). Systems also require *identification* of users, whether to detect multi-accounts, or to *prevent repudiation* (i.e. denying of an act) to engage users *accountability*. The system also requires *confidentiality* of information and to remain secure even if it has been compromised at some points, past exchanges must remain secured (*Backward Secrecy*), as well as future exchanges (*Forward Secrecy*).

Privacy ensures the right of users to be left alone. *Confidentiality* of users personal data must be ensured, as well as the *unlikability* of their different information or accounts. Unlikability means that a third party should not be able to determine whether two information/accounts belongs, or not, to a same entity. Unlikability implies users *anonymity* or *pseudonymity*, i.e. not being able to link an information to users real identity. Users has also a *right to be forgotten*, as well as to be able to *repudiate* their own acts. Moreover, they should be able to control their information, i.e. to correct false information, and to choose the information they disclose.

Security and privacy may seem at first opposed, they are not by nature incompatibles. For example, security requires non-repudiation whereas privacy requires repudiation, however, both can be achieved at the same time. Indeed, the entities to which non-repudiation must be achieved in security, e.g. state moderators in the scope of a specific procedure, are not the same in privacy, i.e. all other entities.

1.3 Biometrics

Service Providers through a web page can collect biometric data. Not only they enable identification and profiling of users, they also verify users essence, and not, e.g. the knowledge of a secret. This makes biometric data typically harder to share, copy, and spoof than, e.g. knowledge, or possessions.

Contrary to knowledge and possession-based modalities, biometrics usage is probabilistic, i.e. in an authentication system, legitimate users has some probability to be rejected (FRR), and attackers, to be accepted (FAR). This is due to the variations in users biometric acquisitions (intra-class variations), and similarities between users (inter-class variations).

Biometrics can also cause privacy issues as they are often hardly revocable and renewable, and can be used to deduce personal information about users, s.a. their age, or gender. This makes biometrics a particular authentication modality, as demonstrated by articles regulating use of biometrics in the GDPR.

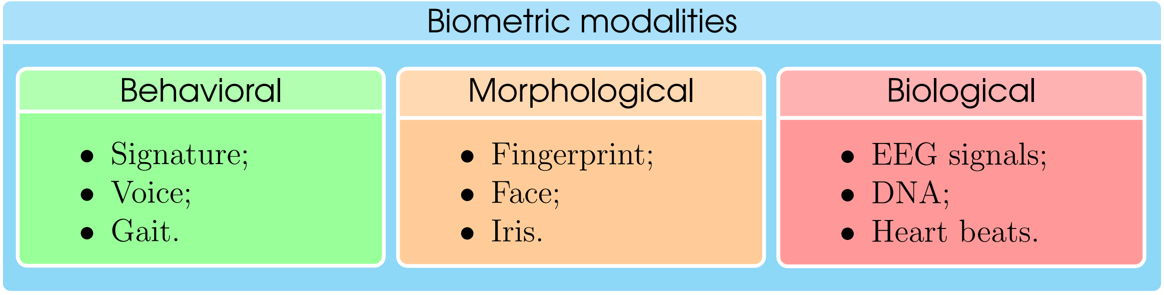


Figure 1.2: Biometric modalities

We present in Figure 1.2 the three kinds of biometric modalities, with examples for each. Biological biometrics, also called hidden biometrics, are users physical particularities invisible without proper devices whereas morphological biometrics can be easily seen by one another. Behavioral biometrics that are based on how users behave, s.a. their way of walking [Bours and Denzer, 2018]. In this thesis, we focus on behavioral biometrics.

We focus on behavioral biometrics as they often do not require additional users action, and thus enable continuous and/or transparent authentication. Unfortunately, the lack of specific action from users also enable their use without users knowledge and consent. Behavioural biometric are also subject of important intra-class variations, as users behaviors change over time, but also depending on their current state (e.g. tired, irritated, sad).

From browsers, Mouse and Keystroke Dynamics, i.e. way of using the mouse and the keyboard, can easily be collected thanks to a simple JavaScript code embedded in the pages visited by the user without requiring any additional sensors other than users keyboard and mouse. In this PhD thesis, we focus on Keystroke Dynamics, as it is one of the speciality of the GREYC laboratory with several thesis conducted in this domain [Mhenni et al., 2019, Idrus et al., 2013, Giot, 2012].

1.4 Thesis objectives

This PhD thesis focuses at answering Internet actors needs by using Keystroke Dynamics. We enable viewers/SP to use Keystroke Dynamics for security purposes, mainly for users authentication, while enforcing subjects/authors privacy and explicit consent, without needing moderators supervision.

By considering GDPR, we thus provide many contributions to security and privacy on the Internet:

- We propose real-time anonymisation of Keystroke Dynamics in order to prevent their unwanted collection. This enables to ensure users explicit consent for the processing of such data. We present several techniques to protect KD in Chapter 3, as well as a proof of concept in Chapter 6.

- We propose a GDPR-compliant multi-modal biometric authentication scheme without any privacy leakage as the biometric data is not disclosed to the SP. This therefore guarantees that such personal data will not be used by the SP for other purposes than authentication. We present in Chapter 4 such authentication protocol using Keystroke Dynamics, as well as user location, and browser configuration. A proof of concept is then presented in Chapter 6.

- A proof of authorship built on previous contributions is then proposed with a proof of concept in Chapter 6.

- A Social Identity Proof is proposed in Chapter 6 to verify users identity through peers recognition, as well as to enable misbehaving users accountability towards moderators through proper protocol. Still, users privacy remain guarantee.

- We then propose the use of Trusted devices in Chapter 6 in order to protect users Keystroke Dynamics, and therefore privacy, against a corrupted or malicious client.

- We propose a Keystroke Dynamics model, as a way to facilitate research, and to improve Keystroke Dynamics System performances. Modelling of Keystroke Dynamics enables their synthetic generation thus making it possible to augment existing datasets, and in the end, to share Keystroke Dynamics datasets for research purposes without disclosing any real user personal data.

- We present in the next chapter, information that can be deduced from Keystroke Dynamics collected through users browser, and how it threatens users online privacy. We quantify this privacy threat and study the impact of context and configuration on Keystroke Dynamics Systems performances. In this scope, we present a framework to fairly compare Keystroke Dynamics Systems.

Chapter 2

What do websites know about you?

In this chapter, we present information websites can obtain through your browser, posing threat to your online privacy. In the next chapter, we show how to protect these information from malicious websites, and in the next how to use such information in a privacy-compliant authentication scheme.

Keywords: *Keystroke Dynamics Anonymization System; Keystroke Dynamics; Anonymization; WebExtension; JavaScript; Browser; Browser Fingerprinting.*

Contents

Contributions presented in this chapter

- Fairly compare Keystroke Dynamics Systems.
- Distinction between context and configuration.
- Naming convention for context and configuration.
- Impact of context and configuration on Keystroke Dynamics Systems.
- Improvement of distance based Keystroke Dynamics Systems.

Publications

- Migdal, D. and Rosenberger, C. (2019b). Keystroke Dynamics Anonymization System. In *SeCrypt (B - Core)*, Prague, Czech Republic.

2.1 Information leaked by browsers

While browsing the Internet, browser leaks information about users. This section presents such leaked information in order to qualify the privacy threats they represents. In the next chapter, protection strategies are discussed.

Browser Fingerprinting aims at tracking users through their browser thanks to discriminant data a service can collect. This is usually proposed to "personalize services" corresponding to users profile-type, s.a. suggesting contents depending on the user's assumed Internet history. Browser Fingerprinting goal is not to identify users with assurance, but to classify the user into a category, e.g. by identifying a set of browsing sessions belonging to the same user, or type of users. Two PhD thesis study specifically this subject [Laperdrix, 2017, Vastel, 2019].

Panopticlick [Eckersley, 2010], IAmUnique [Laperdrix et al., 2016], and UniqueMachine [Cao and Wijmans, 2017] websites enable the computation of browser fingerprints from data collected by the website, generally through the network and a JavaScript API, to determine the fingerprint uniqueness among the previously computed. The more the browser fingerprint is unique, the more the service is able to discriminate the user. Information used for browser fingerprinting might be linked, e.g. to the hardware (e.g. GPU [Cao and Wijmans, 2017], screen), to the operating system, to the browser, its configuration, installed fonts [Eckersley, 2010, Laperdrix et al., 2016], browser history [Weinberg et al., 2011], or blacklisted domains [Boda et al., 2012]. Such identification and tracking are often not consented by the user, and poses a threat to users' online privacy, thus, leading researcher and developers to study this issue and propose solutions in order to protect users' privacy [Laperdrix et al., 2016, Nikiforakis et al., 2015, Moore and Thorsheim, 2016, Eckersley, 2010, Acar et al., 2013].

Biometric capture can also be added to browser fingerprinting, e.g. using the mouse [Jorgensen and Yu, 2011, Shen et al., 2013] or/and Keystroke [Revett et al., 2007a, Giot et al., 2011, Kim et al., 2018].

Other biometric information can also be collected s.a. the user location, and its journey routines; the time he visits webpages, and thus its daily habits. With authorization form the users, the webcam and microphone can also be used to collect users biometrics, or to deduce its environment.

2.2 Keystroke Dynamics

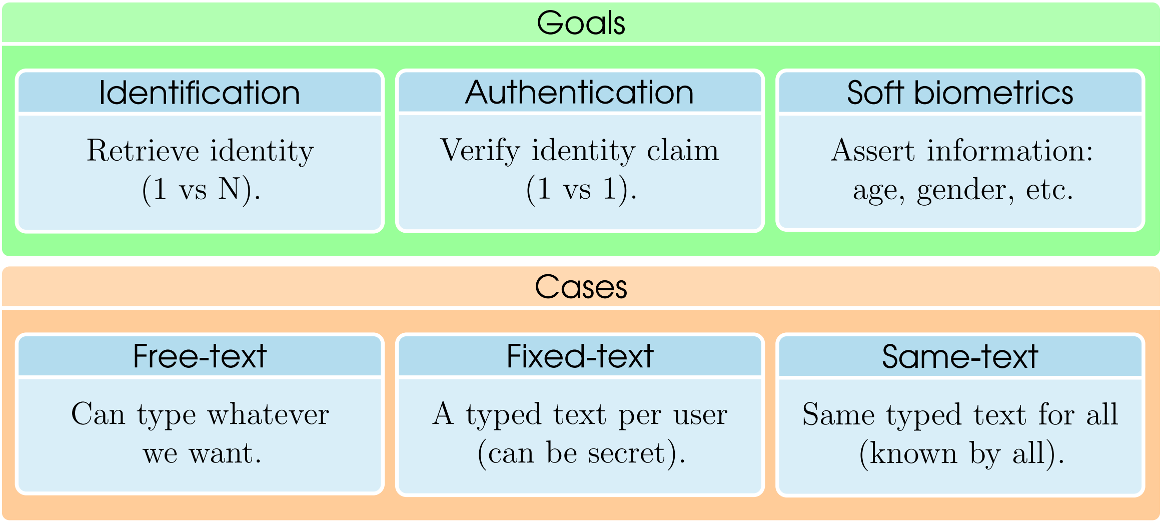
Keystroke Dynamics (KD) enables the profiling of users (s.a. identification, authentication, assertion of users gender/age/handedness/emotions) by analyzing their way of typing, e.g. when browsing the Internet.

First works on KD have been done in the eighties [Gaines et al., 1980], although, the idea of using a keyboard to automatically identify individuals has first been presented in 1975 [Spillane, 1975]. In the preliminary report of Gaines *et al.* [Gaines et al., 1980], seven secretaries typed several paragraphs of text and researchers showed that it is possible to differentiate users with their typing patterns. Since then, several studies have been done, allowing to decrease the quantity of information needed to build the biometric reference, while improving the performances [Umphress and Williams, 1985, Monrose and Rubin, 2000, Revett et al., 2007b, Lee and Cho, 2007, Giot et al., 2011]. However, studies cannot be compared, or are unfairly compared, as they each use different datasets and protocols [Giot et al., 2011].

Although, Keystroke Dynamics may be used to pursue legitimate purposes, it also constitutes a threat against users privacy. Indeed, KD enables identification and deduction of private information (s.a. gender, age, handedness, or emotions) without users consent or awareness. In this section, we are seeking to quantify this threat against user privacy that we later limit in the next chapter. The evaluation scheme and metrics introduced in this section is used in the next chapter to quantify privacy gains after KD anonymization, and in chapter 4 to compare KDS performances before and after KD protection. KD anonymization protects users against unwanted use of their KD, and KD protection enables KD-based authentication while protecting users privacy, which is a GDPR requirement.

This section is organized as follows. After presenting some backgrounds on Keystroke Dynamics, we propose an evaluation scheme enabling fair comparison of Keystroke Dynamics Systems. In a third part, we then demonstrate the necessity of our fair evaluation scheme by highlighting context and configuration influences on KDS performances.

2.2.1 Backgrounds



Goals

Identification: Retrieve identity (1 vs N).

Authentication: Verify identity claim (1 vs 1).

Soft biometrics: Assert information: age, gender, etc.

Cases

Free-text: Can type whatever we want.

Fixed-text: A typed text per user (can be secret).

Same-text: Same typed text for all (known by all).

Figure 2.1: Keystroke Dynamics usages.

In the scope of our work, we focus on the key pressure and released times received by the Operating System or/and the browser, on a laptop or desktop computer. Keystroke Dynamics on Smartphone usually use other modalities, s.a. screen pressures and phone orientations while typing, however, they are not studied here. Other studies use Neural Networks for KD on Smartphone [Clarke and Furnell, 2007]. More precisely, and as shown in Figure 2.2, for each typed digraph (two consecutive characters), we use the 6 durations (d_0 to d_5) from the 4 received time corresponding to the pressure (P0) and release (R1) of the first character, and the pressure (P0) and release (R1) of the second character. After the user has typed some text, the resulting vector of durations is then used to compute a reference (or a sample) modeling its way of typing. In the scope of this thesis, the duration i of the digraph d, in the entry e of the user u, will be written as $d_i(u, e, d)$. $*$ denotes all durations, digraphs, entries or users, e.g. $d_*(*, *, *)$ is the set of all durations, for all digraphs, entries, and users.

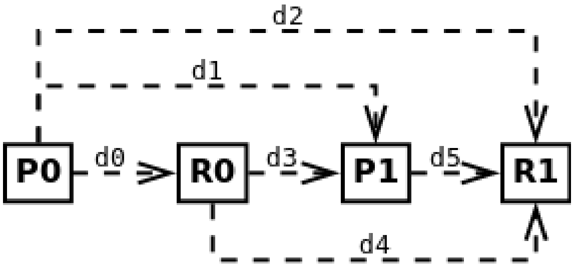
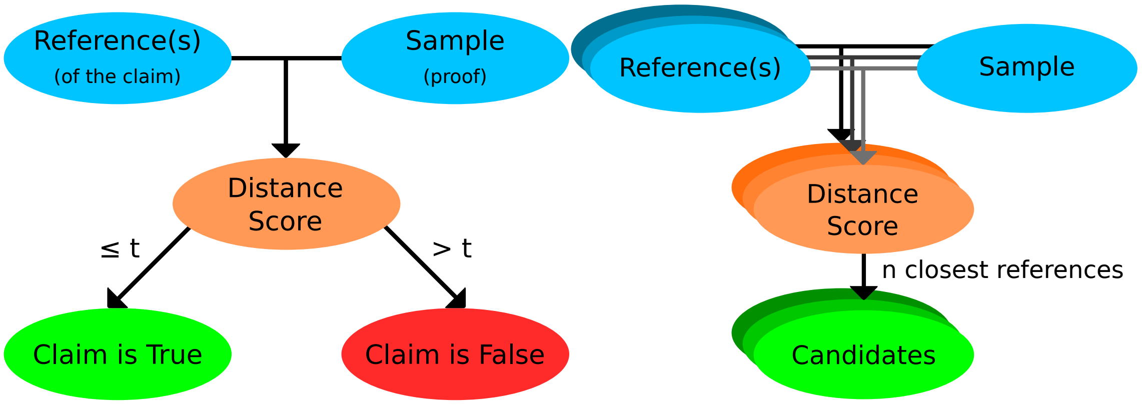


Figure 2.2: Digraph durations

As shown in Figure 2.1, Keystroke Dynamics can be used for different goals (Identification, Authentication, Soft Biometrics) in different cases (Free-text, Fixed-text, Same-text). As any biometric solution, Keystroke Dynamic Systems (KDS) require sets of prior knowledge (references) that are used to verify a new acquired data (sample). For identification and authentication, a reference describes the typing way of a specific user, whereas for soft biometrics, a reference describes the typing way of a set of users (e.g. man, woman, left-handed/right handed person). References then enable, from a sample, to retrieve, or verify, the typing user's identify. Soft biometrics are handled as an identification, replacing ”user” by ”set of users”.

References and samples can either be used in a distance-based or a learning-based (s.a. SVM, Random-Forest, Deep learning) KDS. However, as in identification and authentication scenarios the number of references is usually low, many Keystroke Dynamics Systems are distance-based. Indeed, learning-based techniques often require large amount of references in order to be efficient whereas distance-based techniques can even be used with only one reference per users.

As shown in Figure 2.3, in a distance-based authentication KDS, users claim an identity and will provide a sample as proof of their claim. A distance score is then be computed from the given sample, and the reference(s) associated to the claimed identity. If the distance score is below a given threshold (t), the claim is assumed to be true, else, the claim is assumed to be false. In a distance-based identification KDS, a distance score is computed for all the users reference(s), and the n-closest references to the sample are selected as candidates.



(a) Authentication (b) Identification

Figure 2.3: Distance-based KDS principle.

As previously stated, and as shown in Figure 2.1, Keystroke Dynamics can be used in different cases:

- *Same-Text:* all users type the same text, e.g. the website name, The typed text is thus known at the KDS conception. This *a priori* can thus be used to optimize the KDS performances.
- *Fixed-text:* users have a text associated to their account they need to type in order to authenticate. This text can be secret (e.g. password), unique (e.g. login), both, or neither. However, such text cannot be known at the KDS conception.
- *Free-text:* users can type whatever they like (e.g. a review, a blog/forum post, a chat message). Text is usually longer than in Same/Fixed-text, but is of variable length, thus requiring its transformation in order to use distance functions.

There exist many keystroke dynamics datasets [Monaco, 2018]. Datasets have been cleaned to remove incoherent data, e.g. entries in which the user did not type the asked text. This corresponds to 13% of entries in GREYC W, and less than 3 entries for other datasets. In Free-text datasets, repeated and non-keyboard events have been removed. All timing information has been converted into seconds. In order to get comparable sets in Same/Fixed-Text datasets, only the first 45 entries per users is kept, users with less than 45 entries, and datasets with less than 45 users, are discarded.

Datasets used in the scope of this thesis are described in the Tables below. From the existing Same/Fixed-Text datasets, only 3 matched our criteria. One of them containing two typed text, 4 datasets are thus presented in Table 2.1. For Same/Fixed-Text Soft Biometrics, 5 dataset are used, as shown in Table 2.2. The KPPDW dataset is used for Free-Text, where users wrote text on several subjects, with one entry per users, and at least 250 digraphs per entries. 3 Free-Text dataset are extracted from the KPPDW dataset, one per subject, cf Table 2.3. Free-Text Soft Biometrics is not studied as no available dataset has been found.

2.2.2 Fair Evaluation Scheme

As previously stated, studies are difficult to compare, or unfairly compared, as they each use different datasets and protocols [Giot et al., 2011], leading us to propose a fair evaluation scheme for Keystroke Dynamics Systems. We distinguish 4 components in our scheme, that is detailed in the following:

- Attacker model: describes the capabilities of the attacker;
- Metrics: quantifies the success of the attack;
- Context: describes the data obtained by the attacker;
- Configuration: describes the KDS.

Attacker model

We focus on a browser environment. The attacker is able to execute arbitrary JavaScript codes on the Web pages visited by the users, in order to identify, authenticate, or profile them, using only the keyboard events' timestamps.

As shown in Figure 2.4, the attacker model is based on a real-life two phases scenario. Due to a vulnerable or complicit website, the attacker is first able to collect Keystroke Dynamics and to assert the user true identity. Then, due to some changes, the attacker is subsequently only able to collect and compute a sample, without being confident on the user identity. The attacker will thus seek to retrieve the user identity with (authentication) or without (identification) an *a priori*.

Name	Text	# of users	Source
GREYC K	greyc laboratory	104	[Giot et al., 2009]
GREYC W1	laboratoire greyc	62	[Giot et al., 2012]
GREYC W2	sésame	46	[Giot et al., 2012]
CMU	.tie5Roanl	51	[Killourhy and Maxion, 2009]

Table 2.1: Description of used Same/Fixed-text datasets.

Name	Text	# of users	Source
GREYC N1	leonardo dicaprio	110	[Syed Idrus et al., 2013]
GREYC N2	michael schumacher	110	[Syed Idrus et al., 2013]
GREYC N3	red hot chilli peppers	110	[Syed Idrus et al., 2013]
GREYC N4	the rolling stones	110	[Syed Idrus et al., 2013]
GREYC N5	united states of america	110	[Syed Idrus et al., 2013]

Table 2.2: Description of used Same/Fixed-text Soft-Biometrics datasets.

Name	About	# of users	Source
KPPDW1	Gay Marriage	400	[Banerjee et al., 2014]
KPPDW2	Gun Control	400	[Banerjee et al., 2014]
KPPDW3	Restaurant Reviews	500	[Banerjee et al., 2014]

Table 2.3: Description of used Free-Text datasets.

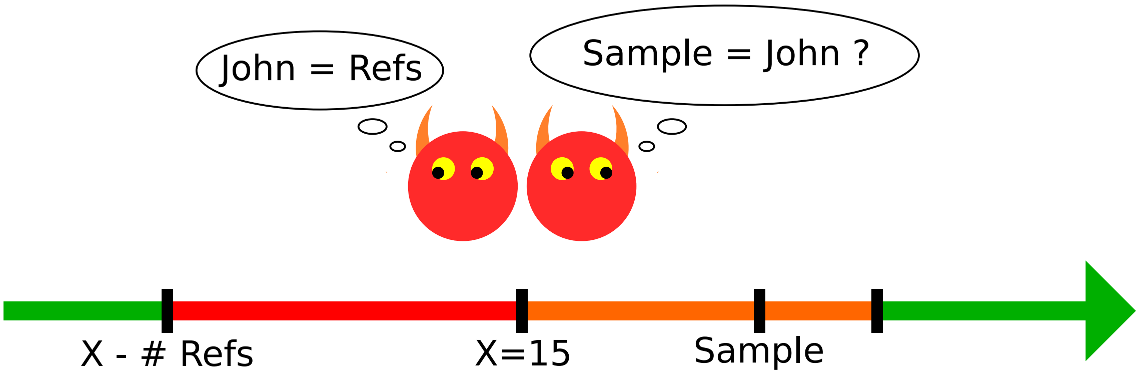


Figure 2.4: Attacker model

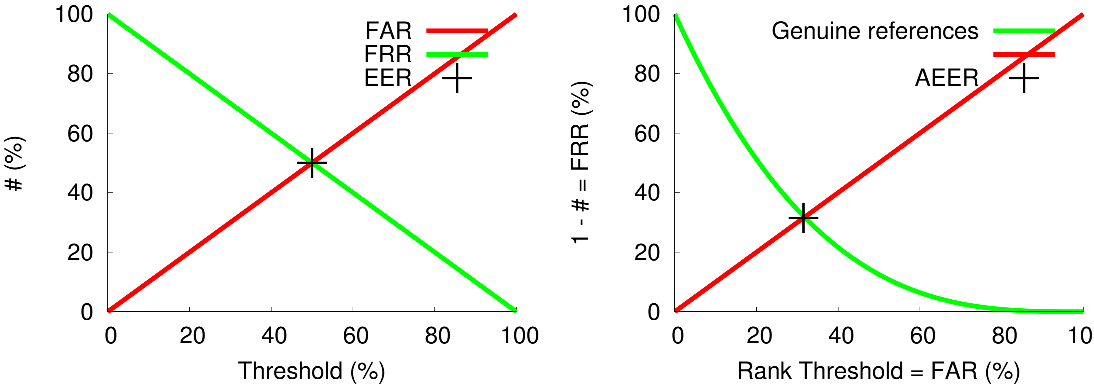
It is thus assumed that the attacker has, during the first phase, a way to identify users that becomes unavailable in the second phase. The reason could be a change in the user IP address as the user rebooted its Internet box, started to use a VPN, or is moving. The browser fingerprint could have been changed due to a browser update or a change of browser. An information could have been leaked by a third party integrated to the Webpage. Other biometric modalities, e.g. the mouse dynamics, face recognition through the webcam, etc. could have been used.

Our proposed attacker model has however some limitations. We assume that the samples are posterior to the references, yet they could as well be anterior or simultaneous to the references. As we focus on finding the best distance (described by the configuration) in function of the context, we are not interested in references update mechanisms, or in realistic systems with user-dependant thresholds. Indeed, both would impact the performances in the way that does not depend on the distances capabilities, but more on their own performances and compatibility with the distances. Moreover, we are not interested in a realistic attacker whose performances would depend, and be limited by, an arbitrary implementation. Instead, we aim at an unrealistic attacker whose performances are a theoretical maximal bound of the performances that could be expected in real-life; for example by giving the attacker the unrealistic power to assert, for each user, the optimal threshold, instead of depending on a sub-optimal implementation.

Metrics

In order to evaluate the performances of KDS with objective metrics, we use the Error Rate (ER) which corresponds to the number of false predictions over the total number of predictions, i.e. $ER = \#(Errors)/\#(Samples)$. It is worth noticing that $ER = 1 - Acc$, where Acc is the accuracy.

For authenticating KDS, the ER depends on the threshold, thus having one ER per possible threshold. As shown in Figure 2.5a, we use the ER obtained when the False Rejection Rate (FRR) equals the False Acceptation Rate (FAR), i.e. when the rate of rejected legitimate users equals the rate of accepted illegitimate users. The obtained ER is then called the Equal Error Rate (EER).



(a) EER for authentication

(b) AEER for identification

Figure 2.5: Identification and authentication KDS metrics.

For identifying KDS, the accuracy obtained by considering only the closest reference is generally used. However, we argue that such metrics is not satisfactory. Indeed, the rank of the legitimate(s) reference(s) indicates a level of anonymity, i.e. if the legitimate reference is the n-closest one, the legitimate user can be assumed to be anonymous among at least n users. Then, even if the legitimate reference is not the closest one, it could be e.g. among the 5 closest, or 1% closest, that is still an interesting result, enabling to discriminate users. As shown in Figure 2.5b, we use the EER considering the identifying KDS as an authenticating KDS with the authentication distance score being the rank of the legitimate reference. We call this metric Authentication EER (AEER), which correspond to ER where the rate of legitimate reference(s) that are not in the n-closest references equals the rate of illegitimate reference(s) that are in the n-closest references.

In order to keep reasonable computations times, considering that a large amount of contexts and configurations will be tested, ER are computed as following. For identification, only a forth of the references are used per samples. For authentication, all samples (30) are used to compute legitimate distances scores, while only 30 illegitimate distances scores will be computed per users. Illegitimate samples are randomly selected from the samples ensuring that all samples is used only once. For Soft Biometrics, all entries are used to compute the ER with cross-validation using a kFold of 5.

Context

The context describes the conditions in which the attack occurs. The context influences the attack performances. It includes the datasets used to compute the metrics, i.e. the number of users, the number of entries per users, the type of text (Same-Text, Fixed-Text, Free-Text), and the typed text with its length. In the scope of this thesis, the datasets presented in Section 2.2.1 will be used. Metrics are computed as the mean of the metric across all dataset.

Studies are often not comparable as they do not use the same datasets in order to produce their metrics. But even if they did, the different ways the references and samples are selected also influences the KDAS performances and make them incomparable. Indeed, some randomly selects references and samples, while other use the first entries as references, and the nexts as samples. In our attack model, 3 parameters describes the way the references and samples are selected:

- *The position of the attack*, separating the references from the samples. In our case, the first 15 entries are used as references as the others as samples.
- *The number of references*, from 1 to 15.
- *The sample range*, relatively to the position of the attack, a subset of [1;30].

In order to prevent bias due to the evolution of the keystroke through time, the position of the attack is fixed, and the n-first entries preceding the position are used as the n references.

Configuration

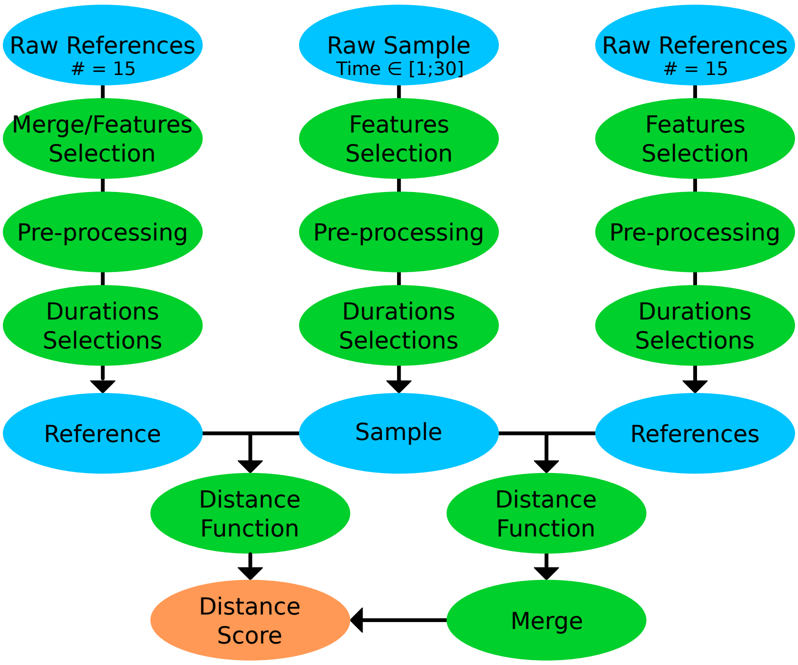


Figure 2.6: Context (in blue) and Configuration (in green)

There is an infinity of possible ways to compute a distance score from a reference(s) and a sample. Obviously, we are only able to test a subset of the possibilities. We propose in Figure 2.6 a pipeline enabling to compute such a distance score. The configuration describes each part of this pipeline:

- *Distance function*: computes a distance score from a reference and a sample.
- *Pre-processing*: modifies the references and samples in order to improve the distance function performances.
- *Feature selection*: for free-text, build a vector of durations from the free-text digraphs durations.

- *Durations selection*: selects the entry's durations to use.
- *Merging*: enables to compute a single distance score from several references and one sample.

Synergies has been observed between configurations parameters. We thus argue that the impact of a given parameter value, in a given context, cannot be estimated without computing the performances of all configurations involving this value. The impact of a given parameter value will thus be estimated as the maximal performance it enables to achieve.

Naming convention

We propose a naming convention of context and configuration in order to quickly describe them. We propose the following format: *configuration@context*, with *configuration* and *context* concatenations of parameters separated by '.'.

We propose the following format for *configuration*:

- *Distance function*, e.g. Hocquet, Manhattan;
- *Pre-processing*, e.g. density, reduce;
- *Feature selection* (for Free-Text), prefixed by the number of features, e.g. 38.quantiles;
- *Durations selection*, e.g. 034, 013, 045;
- *Merging* (for multiples references), e.g. min.

We propose the following format for *context*:

- The type of Same/Fixed-text (for Same/Fixed-text), st for Same-Text, ft for Fixed-Text;
- *Number of digraphs* in the entry (for Free-Text), e.g. 125;
- *Position of the attack*, e.g. 15 (Same/Fixed-Text), 1 (Free-Text);
- *Number of references*, e.g. 15, 9, 1.
- *Sample range* (optional), assumed to be all entries after the attack.

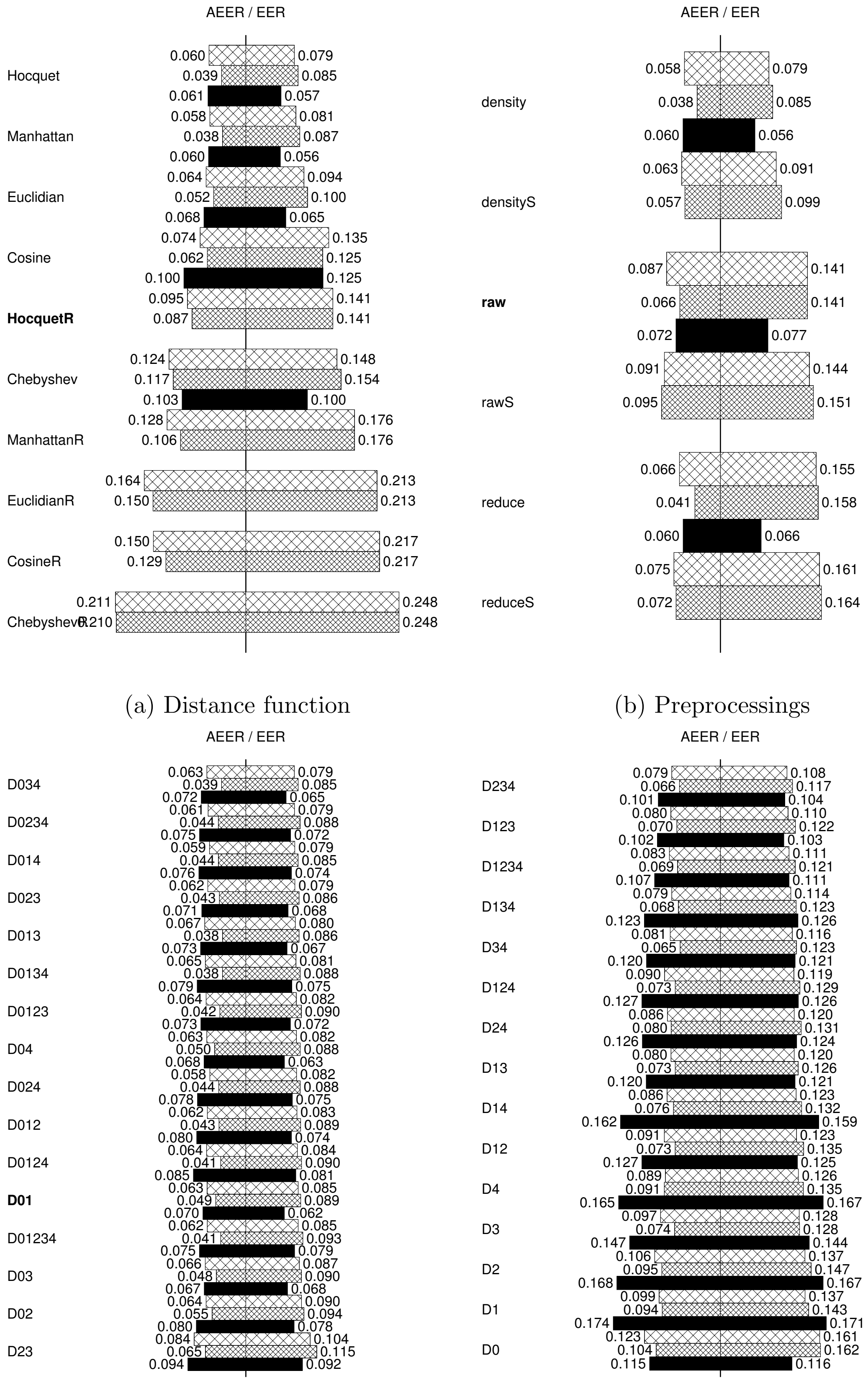
Some examples are proposed below:

- Hocquet.density.034.min@st.15.15
- Manhattan.density.013.min@ft.15.9;
- Manhattan.reduce.38.quantiles.045@125.1.1.

The next sections describe the influence of the parameters of configuration and context on the performances.

2.2.3 Influence of configuration and context on performances

Configuration and context influence the identification and authentication performances, the impact of each parameters is described in Figures 2.7 and 2.8. Each are described in the following. For a given parameter value, performances are the mean performance across all datasets, for the best configuration and context involving the



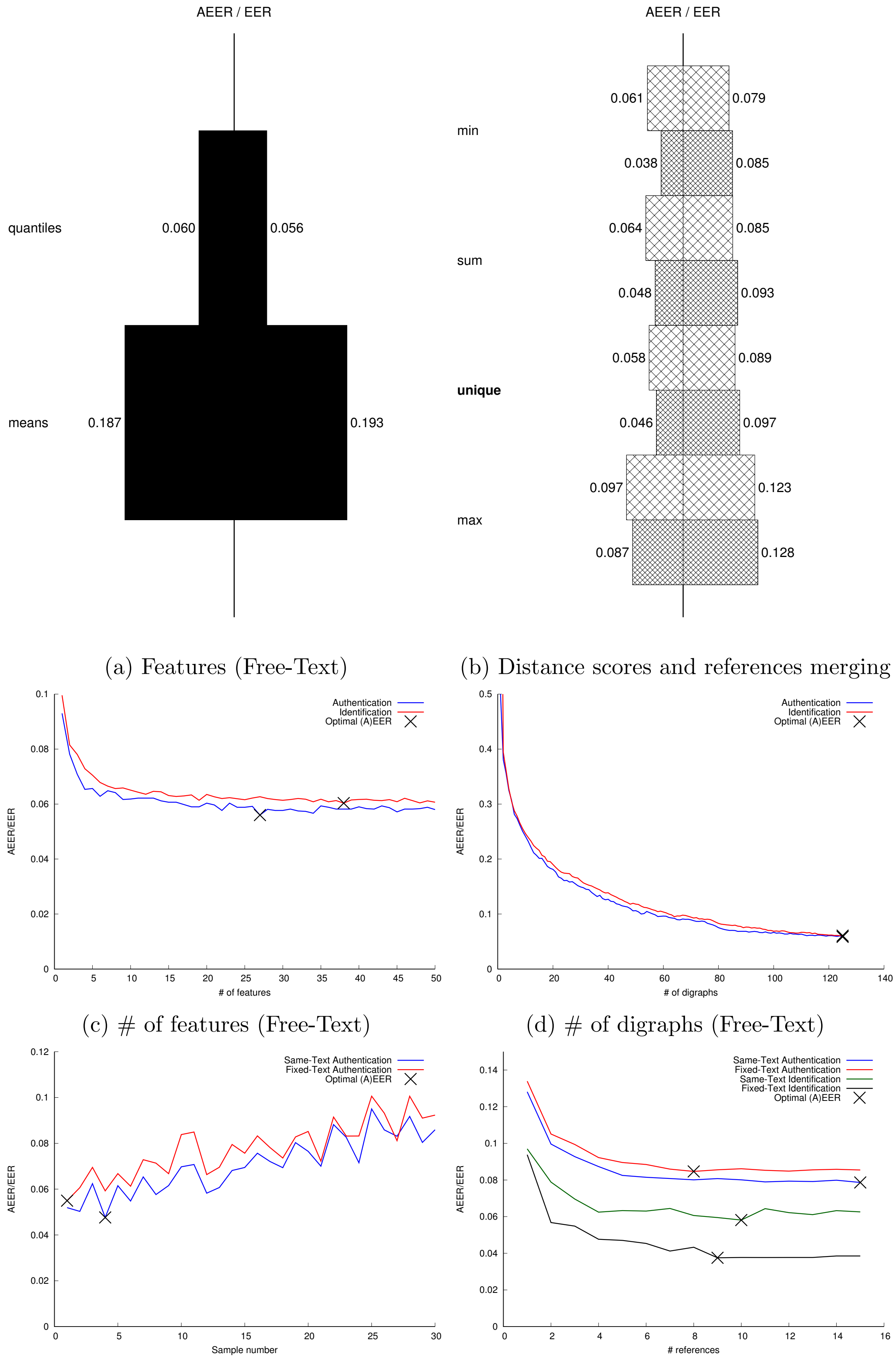
(a) Distance function

(b) Preprocessings

(c) Durations 1/2

(d) Durations 2/2

Figure 2.7: Influence on context and configuration on the performances (1/2)



(a) Features (Free-Text)

(b) Distance scores and references merging

(c) # of features (Free-Text)

(d) # of digraphs (Free-Text)

(e) Sample range

(f) # of references

Figure 2.8: Influence on context and configuration on the performances (2/2)

value. The best configuration and context is deepen in the end of this section, with user-dependent thresholds, and compared with the state of the art.

Distance function (configuration)

The distance function computes a distance from a single reference and a single sample. We focus on 5 different well-known distances functions and compares them to the SVM performances. The distances were generalized, and tested both with and without reduction.

Assuming a reference $R = d_*(u, e, *)$, and a sample $S = d_*(u', e', *)$, with $R_{i,d} = d_i(u, e, d)$ and $S_{i,d} = d_i(u', e', d)$, the distances are defined as follow:

- *Cosine*: $-\Sigma_{\{i,d\}} R_{i,d} * S_{i,d} / \sqrt{R_{i,d}^2} * \sqrt{S_{i,d}^2}$;
- *Minkowski*: $(\Sigma_{\{i,d\}} |R_{i,d} - S_{i,d}|^n / |S|)^{1/n}$;
- *Hocquet*: $1 - \Sigma_{\{i,d\}} e^{-|R_{i,d} - S_{i,d}|} / |S|$.

The cosine distance is computed as the opposite of the cosine similarity. The Minkowski distance is equivalent to a generalized mean of the absolute differences between the reference and the sample. The Hocquet distance is the generalization of the distance used in [Hocquet et al., 2007]. In this study, 3 particular cases of the Minkowski distance is used:

- *(n=1) Manhattan distance*: the mean of the absolute differences;
- *(n=2) Euclidian distance*: the mean of the square differences;
- *(n=+∞) Chebyshev distance*: the maximal difference.

It is worth noticing that the Bleha [Bleha et al., 1990] and Monrose [Monrose and Rubin, 1997] distances are equivalent to specific Euclidian distance-based configurations.

The reduction is performed by computing standard deviations from the references, $\sigma_{i,d} = \sigma(R_i(u, *, d))$, and using them to divides each values of the reference and sample $R'_{i,d} = R_{i,d}/\sigma_{i,d}$, $S'_{i,d} = S_{i,d}/\sigma_{i,d}$. Of course, reduction cannot be performed is only one reference is available, e.g. in case of a reference merge (see next sections).

Influence of the distance function on the performances are shown in Figure 2.7a. When values are reduced, the distance name is suffixed by 'R'. For authentication, the Hocquet distance outperforms all other, whereas for identification, Hocquet distance is the second best, behind the Manhattan distance, though the difference is not significant ($\leq 0.2\%$).

Surprisingly, the bests performances are not obtained with reduction, which is due to the density pre-processing that performs better without reduction, whereas e.g. the raw pre-processing performs better with reduction. This is an example of synergy between parameters.

The performances of distance-based KDS are compared to SVM-based KDS in Figure 2.9. Authentication uses a one-class linear SVM, while identification uses a multi-class linear SVM. For SVM, only a subset of parameters were tested, the number of references, the digraphs selection, with and without values sorting.

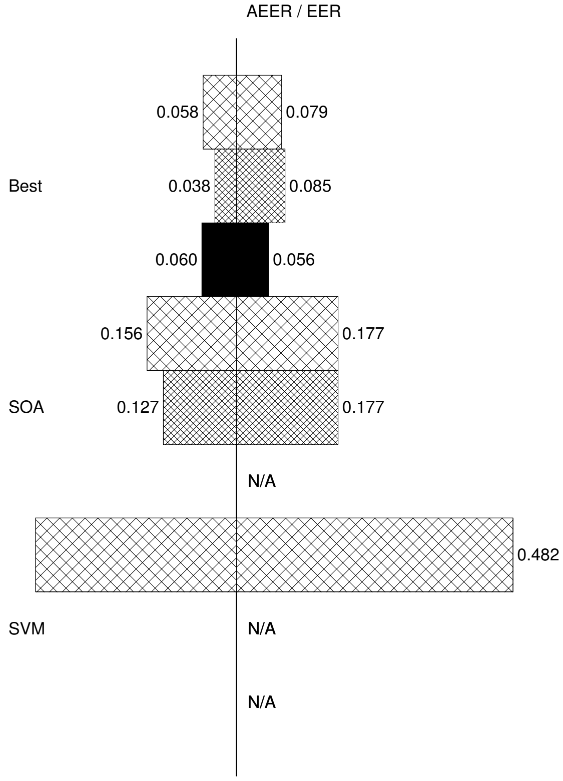


Figure 2.9: Comparison between distance-based (SOA and Best) and SVM-based KDS.

Performances are deceiving and do not correspond to the ones obtained in [Giot et al., 2011], which can be explained by the fact we do not randomly select references. Indeed, as the user keystroke dynamics evolves through time, randomly selecting references enables a more diverse learning. Moreover, the context and configuration slightly differs from ours.

Pre-processing (configuration)

Pre-processing modifies the references and samples in order to improve the distance function performances. We focus on 4 different pre-processing mechanisms. Values are then sorted by durations. Assuming en entry $E = d_*(u, e, *)$, with $E_{i,d} = d_i(u, e, d)$, the pre-processing functions are defined as follow:

- *Raw*: no preprocessing, $E'_{i,d} = E_{i,d}$;
- *Reduce*: $E'_{i,d} = E_{i,d}/\sigma_{i,d}$, with:
  - $\sigma_{i,d} = \sigma(d_i(*, *, d))$ for same-text;
  - $\sigma_{i,d} = \sigma(d_i(*, *, *))$ for fixed-text.
- *Uniformize*: $E'_{i,d} = cdf_{i,d}(E_{i,d})$, with *cdf* a cumulative distribution function:
  - $cdf_{i,d} = cdf(d_i(*, *, d))$ for same-text;
  - $cdf_{i,d} = cdf(d_i(*, *, *))$ for fixed-text.

We estimate the cumulative distribution function through a rank transformation (*density*), i.e. the position of the value in the list of all known values, divided by the length of the list, $cdf(x) = pos(x, X)/len(X)$, with X the sorted list of the known values. It is worth noticing that the attacker has knowledge of the global distribution of the duration across all users. Other estimations of the cdf function are experimented at the end of this section, but only for the best configuration and context with same-text authentication and identification, due to computation time (and memory space) limitations.

Although value reduction is tested, value centering and standardization is not, as most of the distances are based on a difference between entries values, i.e. a centering of the values would make no differences on the results, $x - \mu - (y - \mu)$ being equal to $x - y$.

Discretization as preprocessing is also not tested in this study. Existing studies have found that the clock resolution influence KDS performances [Killourhy and Maxion, 2008], and discretization might improve KDS performances [Giot et al., 2011]. However, the gain is small (near 1 point on the EER), and it is unclear if the gain is not simply due to chance. In a previous study [Migdal and Rosenberger, 2019c], we found out that such gains are seemingly unpredictable and dataset-dependant, as shown in Figure 2.10. In this study, we used the Hocquet distance on an authentication KDS with the 10 first entries used as references. This study also shown that the clock jitter does not have a significant impact on the KDS performances.

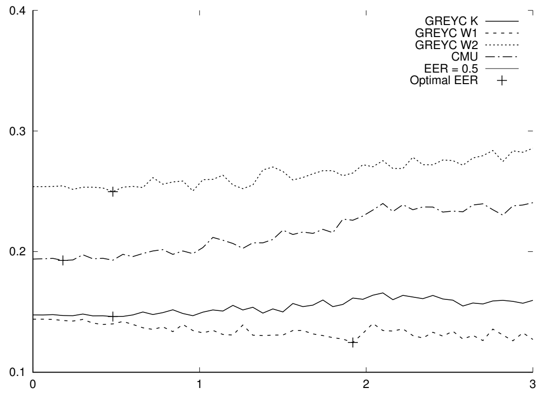


Figure 2.10: Impact of the discretization on the performance of KDS (zoomed).

Influence of the pre-processing on the performances are shown in Figure 2.7b. When values are sorted the preprocessing name is suffixed by 'S'. The density pre-processing offers the best performances while sorting the values decrease the performances. Reducing provides better performances than raw on identification KDS, and on free-text authentication KDS.

As previously stated, the cdf function can be estimated in other ways, by assuming a random distribution followed by the values, and estimating its parameters. We tested several estimators while assuming the values to follow a normal, gumbel, or logistic law. The parameters of these laws can either be computed from the mean and standard deviation of the values (raw estimator), or by using a fitting algorithm (see Chapter 5).

Contrary to the rank transformation that requires all known values, these estimations only requires to transmit and store the law name and the parameters. Still, the estimation of the parameters requires all known values, although, a better understanding of the model followed by Keystroke Dynamics could enable to compute any parameters without requiring any values. In Chapter 5, we propose a first modelization of Keystroke Dynamics, but further studies still needs to be conducted.

The law providing the best performances has found to be the gumbel law. On the two configurations and contexts tested, the raw estimator decreases the performances by 1.2 points for authentication, and 0.2 points for identification, whereas the fitting estimators increases the performances by 0.05 points for authentication, and decreases the performances by 0.9 points for identification.

These results might differs depending on the configuration and context. It is also worth noting that the bests configurations and contexts performances are very close by a non-significant distance (e.g. 0.03 points for two best configurations and context for same-text authentication). A better model could also lead to improve such performances.

Feature selection (configuration)

In Free-text, the typed text changes at each entry, even for the same user, i.e. the typed digraphs differ. State of the art either assumes some digraph/world would be typed [Idrus et al., 2013], or requires a long typed text. We propose in this section ways to build a fixed-length vector from any typed free-text.

Assuming an entry $E = d_*(u, e, *)$, with $E_{i,d} = d_i(u, e, d)$, the pre-processing functions are defined as follows:

- *mean*: mean of the durations for each of the most frequent digraphs in the typed text, $E'_{i,f} = \mu(E_{i,\{d=f\}})$;
- *quantiles*: $E'_{i,*} = quantiles(E_{i,*})$.

The number of the features f is fixed empirically.

Results are shown in Figures 2.8a and 2.8c. Quantiles features are near 4 times better than means features. As expected, the more features we use, the better the performance is. However, as the performances initially quickly increase as the number of features increases, it quickly stagnates near 20 features for authentication, and 15 for identification.

Durations selection (configuration)

As previously stated in Section 2.2.1 with Figure 2.2, digraphs are composed of 6 durations. As shown in Figures 2.7c and 2.7d, using all available durations does not provide the best performances, which is achieved when using durations d_0, d_3, and d_4.

For Same/Fixed-text, as consecutive digraphs share a same duration, $d_5(u, e, d) = d_0(u, e, d+1)$, d_5 combinaisons involving d_5 durations are not tested, and the last d_5 duration is added to the list of d_0 durations. Free-text uses d_5 durations, but the results involving d_5 durations are not displayed in the Figures.

Other selection of durations could also be experimented, s.a. computing a distance score for each durations, then using a weighted mean to obtain a final distance score. A digraph durations could also be transformed, e.g. using a PCA.

Merging (configuration)

In our model, the attacker has many references per users, however the distance function only takes one reference to compute a distance score. Then, either the references should be merged to produce an unique reference, and thus an unique distance score, or many distances scores should be computed then merged to produce an unique distance score.

The tested merging strategies are defined as below:

- *min*: the minimal score;
- *max*: the maximal score;
- *sum*: the sum of the scores (equivalent to the mean);
- *unique*: the mean of the references, $R_{i,d} = \mu(d_i(u, *, d))$.

The performances of such strategies are presented in the Figure 2.8b, the min merging strategy is the best in all cases, although, for identification, it is very close to the sum and unique strategies. The max strategy however, performs worst than any other.

The experimented merging strategies could also be generalized. Be S_i the distance score computed with the reference R_i, and S the final distance score. The mean, max, and sum strategies can be generalized as a weighted sum $S = \Sigma_i w_i * S_i$, assuming the scores sorted by values. The scores could also be sorted by quality (using a quality metric), or by time. Unique is computed as a mean of the references, but could also be computed as a weighted mean, and as for distance scores merging, the weight could depend on the quality, or the time of the reference.

Number of digraphs (context)

Free-text is likely to be used for continuous authentication or identification. However, the size of the chunk used to compute the references and samples impacts the performances, as shown in Figure 2.8d.

Obviously, using a very small number of digraphs will not enable neither identification nor authentication, and the more the number of digraphs, the better the performances are.

Number of references (context)

The number of references also impact the performances. The number of references is also a specific case of the generalized merging of distance score, when the scores are sorted by time.

As shown in Figure 2.8f, even one reference is enough to enable authentication or identification. and obviously, the more the number of references we consider, the better the performance is. However, the performances quickly stagnate, which seems to indicates that too old references are not useful.

Sample range (context)

The freshness of the references compared to the received sample also impact the performances.

As shown by Figure 2.8e, the best performances are obtained if the sample is consecutive to the references, and the performances decrease when the references are too old.

This suggests that, in order to maintain good performances over time, new references should be added regularly.

User dependant thresholds (Attacker model)

In our attacker model, we assume that the attacker use a unique threshold for all users. However, in real-life, attackers can improve performance of KDS by using user-dependent thresholds [Mhenni et al., 2019].

We thus quantify, in the following, the maximal theoretical performance gain offered by users-dependant thresholds on Same/Fixed-Text authentication. The attacker is then granted the unrealistic capacity to choose the optimal threshold for each users.

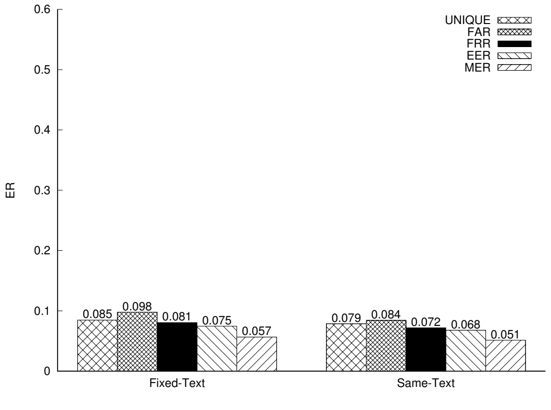


Figure 2.11: Impact of user-dependant thresholds on authentication.

The thresholds are chosen is such a way that each user:

- *UNIQUE*: have the same thresholds (no-user dependant);
- *FAR*: have the same FAR;
- *FRR*: have the same FRR;
- *EER*: have the same EER;
- *MER*: have a minimal Error Rate.

Only two of the bests configurations and contexts are tested, Hocquet.density.014.min@st.15.8 for Same-Text, and Hocquet.density.034.min@ft.15.15 for Fixed-Text. The (E)ER of each strategies is shown in Figure 2.11. MER strategy offers a significant gain on the ER (2 to 3 points). Although the FRR and EER offers performance gains, the FAR strategy increases the ER.

The MER strategy is the best suited strategy for an attacker scenario. However, in a legitimate authentication scenario, the ER strategy is unsuited as some users might have very high FAR or FRR. In the same way, even though they guarantee the same FRR or FAR for all users, FRR and FAR strategies are not suited to legitimate authentications. In such a case, the EER strategy would be more suited as it ensures that all users has a balanced security between FAR and FRR.

References-dependent thresholds could also be tested for identification, with threshold(s) on the distance score or/and the rank.

Limitations

Only a small subset of all possible configuration and context could have been tested. Some differences between configurations and context might not be significant, and due to e.g. the random selection of the impostor samples that introduces some variation on the performances computations. The mean of several iteration should be computed along with a margin of error before assessing a strong ranking of the configurations and contexts.

Soft Biometrics were not tested using distances function. We propose some results based on the work of [Idrus et al., 2013] using a SVM with RBF kernel, γ=0.125, cost=128. The performances of the best configuration and context is shown in Figure 2.12. The same configuration and contexts will are in the next chapter as an attack on user privacy.

2.3 Conclusion

We demonstrate that Keystroke Dynamics constitutes a major threat to users privacy that needs to be addressed. The threat posed by Keystroke Dynamics has been quantitatively estimated through experiments. However, this is a minimal estimation of the maximal attacker power. Indeed, the non-existence of a KDS configuration enabling to achieve greater performances cannot be demonstrated.

Still performances obtained in this chapter are enough to demonstrate threat to users privacy. In Fixed-Text authentication, EER values of 8.5% can be achieved by

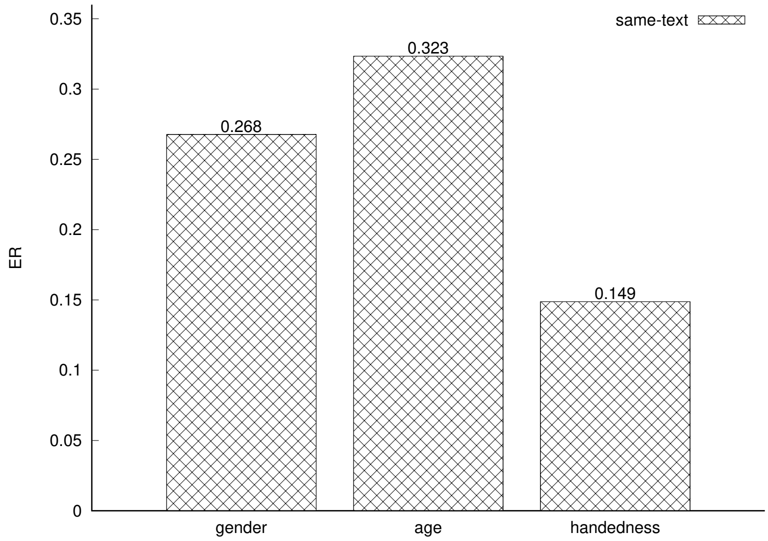


Figure 2.12: EER for authentication

attackers and could even reach 5.7% with user-dependent thresholds. Fixed-Text identification offers even better performances with an AEER value of 3.8%.

In this chapter, we tried to increase KDS performances. In the next chapter, we seek to decrease such performances in order to protect users privacy.

In short: *Although Keystroke Dynamics can be used for security purposes, it also represents a threat to user privacy (identification, profiling). To quantify this threat, we proposed a fair comparison of KDS while distinguishing between context (what the attack has) and configuration (what the attacker do) and their impact on the performances. We focused on the timing event received by the OS/Browser. Our findings strongly suggest the use of user-dependant threshold for better performances, and the use of sliding windows on references in order to maintain performances through time. Better KD modeling could also increase performances of the attacks.*

Chapter 3

How to protect my information from malicious websites?

As seen in the previous chapter, browsers expose personal information that threaten users privacy. This chapter presents several techniques enabling to protect such information against unwanted collect.

Keywords: *Privacy; Anonymization; Keystroke Dynamics Anonymization System; Keystroke Dynamics; WebExtension; JavaScript; Browser.*

Contents

Contributions presented in this chapter

- Real-time Keystroke Dynamics anonymisation.

Publications

- Migdal, D. and Rosenberger, C. (2019b). Keystroke Dynamics Anonymization System. In *SeCrypt (B - Core)*, Prague, Czech Republic.
- Migdal, D. and Rosenberger, C. (2019a). Don't listen to my Keystroke Dynamics! (Summer School). 16th Int.l Summer School on Biometrics and Forensics 2019.
- Migdal, D. and Rosenberger, C. (2019e). Schéma d'Anonymisation de Dynamique de Frappe au Clavier. In *APVP*, Cap Hornu, France.

Note: *[Migdal and Rosenberger, 2019b] has received the best poster awards at Secrypt 2019.*

3.1 Protection of Keystroke Dynamics

As seen in Chapter 2, Keystroke Dynamics enables the profiling of users by analyzing their way of typing [Giot and Rosenberger, 2012, Epp, 2010]. However such modality can be easily be collected without the users consent or knowledge, e.g. when browsing the Internet. Indeed, a simple JavaScript code embedded in the visited web pages enables internet services to collect Keystroke Dynamics and profile users. This poses a major threat to users privacy that needs to be addressed.

While most studies in the state of the art focus on increasing Keystroke Dynamics Systems performances, we address the opposite issue of avoiding the biometric capture of Keystroke Dynamics in order to protect users' privacy. In this section, we aim at protecting users' privacy by anonymizing keystroke data, thus limiting browser fingerprinting and preventing deduction of private information about users, while still allowing the use of this modality for consenting users authentication. Authentication, identification, and profiling can be considered as attacks we limit in this contribution. Experimental results obtained on significant datasets show the benefits of the proposed approaches.

We propose multiple simple solutions for internet users to decide whether its Keystroke Dynamics features could be used or not on a specific website. Using Keystroke Dynamics could be useful to enhance the security of authentication avoiding complex passwords (logical access control to a bank account as for example). For other services, such as social networks, an user might choose to disable the Keystroke Dynamics capture. The proposed methods have been implemented as a WebExtension as an operational proof of concept. With this WebExtension, any user can easily decide for which service, its Keystroke Dynamics features could be used or not (GDPR requirement).

3.1.1 Background

The attacker model is defined in chapter 2. The attacker collects keyboard events timestamps through a JavaScript code embedded in the visited web pages. In a first phase, it is able to build references for each users and will try, in a second phase, to authenticate, identify, or profile users.

Protection

Information can be protected thanks to different techniques. We present some of them below. The next section will introduce some Keystroke Dynamics protections strategies inspired from these techniques.

The most common is simply to suppress such information, or, at least, to ask for users consent before granting access to it. This is used e.g. by WebExtensions and Android applications. Such applications need user consent before accessing some API (e.g. location API). Consent can be given either as the application needs it, or during the application installation. However, this is hardly applicable to Keystroke Dynamics as it cannot be suppressed. This technique inspired the costless (A) strategy. Indeed, if keystroke events are suppressed, the user will no longer be able to type text. If only the keystroke event timestamp is suppressed, attacker could simply use, instead, the time it notices the changes induced by the keystroke events. JavaScript could also be disabled, however it has lots of usage consequences.

Another technique is to lower the information precision, which is already done by browsers. Indeed, browsers round timestamps in order to prevent fingerprinting attacks [1]. However, the precision of 2 milliseconds is still too high to prevent profiling through Keystroke Dynamics, and a too low precision would be prejudicial to some applications (e.g. online games). A combination of rounded timestamp and a required authorisation for higher timestamp resolutions would however be interesting. This technique inspired the costless (D) strategy and the non-blocking delay strategy.

Randomizing the information can also protect them. Obviously, the content of the typed text cannot be randomized, but the timestamp of the events, in some extends, can. Such technique is used, e.g. to counter browser fingerprinting [Nikiforakis et al., 2015]. This technique inspired the non-blocking rdelay strategy.

Adding false information can also prevent profiling, although it is not compatible with Keystroke Dynamics as added Keystroke events would modify the typed text.

Other techniques can also seek to standardize the information. This strategy is used by the Tor Browser to prevent browser fingerprinting attacks [2]. This can be hardly enforced in KD as users have very different typing speeds. This technique inspired the blocking strategies.

The main idea of protecting users from identification/profiling given the Keystroke Dynamics data is thus to disturb the collected information. The attacker being able to embed arbitrary JavaScript code into web pages, it is able to measure the timestamps of keyboard events she/he receives with the JavaScript function `Date.now()`. Thus, modifying the events' timestamps will have no effect, as the attacker can measure them himself. However, events can be delayed, i.e. waiting some time before sending the keyboard event. As JavaScript events loop is mono-threaded, any active wait is troublesome and will be easily detected by the attacker using `setInterval()`, thus requiring the delayed event to be destroyed, and recreated after a passive wait with `setTimeout()`.

The way the Keystroke Dynamics is protected, and the eventual parameters of such anonymization scheme is also assumed to be known by the attacker. Thus, such parameters should be set for all users in order to prevent the attacker from using them to discriminate users through browser fingerprinting techniques [Eckersley, 2010].

[1] https://developer.mozilla.org/en-US/docs/Web/API/DOMHighResTimeStamp

[2] https://panopticlick.eff.org/self-defense

State of the arts

Very few works have been done in the state of the art to avoid the correct capture of Keystroke Dynamics on Internet. To our knowledge, there exists a single work (but no research papers) implemented as a browser extension. KeyboardPrivacy [Moore and Thorsheim, 2016] is a Google Chrome extension that implements such a protection. Timestamp of each event is computed as follows:

$$t'_i = max(t'_{i-1}, t_i) + \begin{cases} b & \text{1 time out of 2} \\ 0 & \text{1 time out of 2} \end{cases}$$

Where b is a random value following an Uniform distribution between 0 and a (this value is user-defined).

Constraints

As the keyboard events will be delayed, this implies a latency i.e. a time the user will have to wait for its keyboard input to be processed/"drawn". The latency must be minimal and as unperceivable as possible for the user.

Latency will be measured in the number of screen frame skipped assuming a screen frame every 1/60 seconds (60Hz). Costless protections will thus be assumed to have a latency of 0. The latency of a typed text is computed as the maximal latency observed in each pressure events. The latency of a dataset is computed as the mean of the latency of all typed text, while the maximal latency is computed as the maximal latency observed in each typed text. The final metrics are computed as the mean accross all tested datasets.

Pressure keyboard events can only be delayed, and in no case anticipated, as it is impossible (at the exception of auto-complete features), to predict what the user will type. The order of the pressure keyboard events must also be kept in order to keep the meaning of what is typed.

Contrary to other anonymisation techniques, the anonymization is not offline, but in real-time, as the user type. Also, we are not required to conserve any statistical meaning, only the meaning of what is typed.

Metrics will only be computed on the best authentication, identification, and soft biometrics context and configuration found in chapter 2. Used dataset are reminded below in Tables 2.1 to 2.3. These tables are repeated below. This is only indicative as there is no guarantee that these configurations and context are still the best after protection of Keystroke Dynamics. In order to produce better metrics, all configurations and contexts should be tested and compared for a given protection and expected latency. KDAS were tested without user-dependant thresholds. The used configurations and contexts are the following:

- *authentication:* Hocquet.density.034.min@st.15.15;
- *identification:*
  - *Same/Fixed-Text:* Manhattan.density.013.min@ft.15.9;
  - *Free-text:* Manhattan.reduce.38.quantiles.045@125.1.1.
- *soft biometics:* 0123

Name	Text	# of users	Source
GREYC K	greyc laboratory	104	[Giot et al., 2009]
GREYC W1	laboratoire greyc	62	[Giot et al., 2012]
GREYC W2	sésame	46	[Giot et al., 2012]
CMU	.tie5Roanl	51	[Killourhy and Maxion, 2009]

Table 2.1: Description of used Same/Fixed-text datasets.

Name	Text	# of users	Source
GREYC N1	leonardo dicaprio	110	[Syed Idrus et al., 2013]
GREYC N2	michael schumacher	110	[Syed Idrus et al., 2013]
GREYC N3	red hot chilli peppers	110	[Syed Idrus et al., 2013]
GREYC N4	the rolling stones	110	[Syed Idrus et al., 2013]
GREYC N5	united states of america	110	[Syed Idrus et al., 2013]

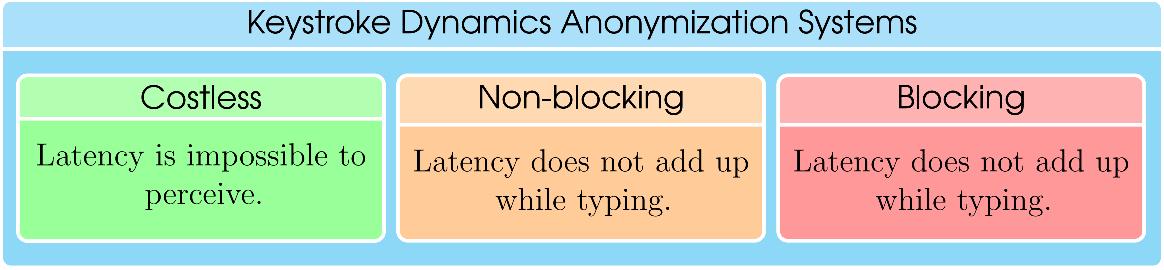
Table 2.2: Description of used Same/Fixed-text Soft-Biometrics datasets.

Name	About	# of users	Source
KPPDW1	Gay Marriage	400	[Banerjee et al., 2014]
KPPDW2	Gun Control	400	[Banerjee et al., 2014]
KPPDW3	Restaurant Reviews	500	[Banerjee et al., 2014]

Table 2.3: Description of used Free-Text datasets.

3.2 Keystroke Dynamics Anonymization System

We propose different solutions to anonymize keytroke dynamics of users we call Keystroke Dynamics Anonymization System (KDAS). Their objective is to be able to use Keystroke Dynamics features for internet services when the user consents (for security applications), and to provide altered data otherwise (for privacy protection). We present in the following several families of KDAS resumed in Figure 3.1.



Keystroke Dynamics Anonymization Systems

Costless

Latency is impossible to perceive.

Non-blocking

Latency does not add up while typing.

Blocking

Latency does not add up while typing.

Figure 3.1: Keystroke Dynamics usages.

3.2.1 Costless protection

Costless KDAS delay keyboard events in such a way that latency cannot be perceived by users. We propose 3 costless KDAS based on 2 strategies:

- Generation of release keyboard events (A);
- Delaying events until the next screen frame (D);
- Doing both (DA).

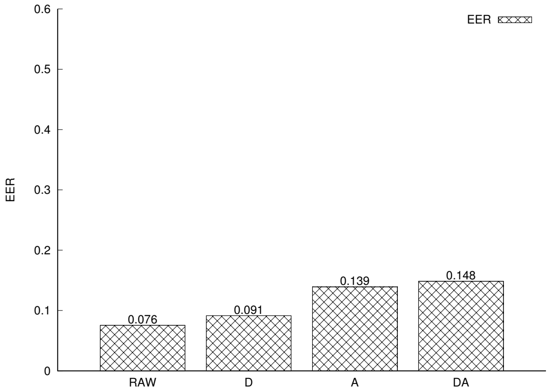


Figure 3.2: Absolute *loss* on the EER with automatic release (A), delay (D) or both (DA) costless KDAS.

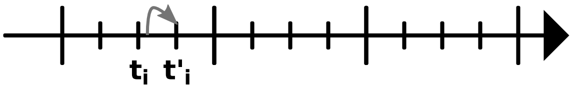


Figure 3.3: Visual representation of the delaying strategy (D). Each vertical line represents a screen frame.

The first strategy (A) is based under the assumption that each pressed key will be released. Release keyboard events can thus be automatically generated after the pressure event, i.e. $d_0(u, e, d) = d_5(u, e, d) = t$, with t an arbitrary time after which the release event will be generated. In case of repeated keyboard events, i.e. the key is hold pressed in order to produce several characters, a new pressure event can be generated for each additional characters. As shown in figure 3.2, such strategy significantly increases the EER value (+7.2 points).

The second strategy (D) exploits the fact that computer screens are refreshed at a regular rate. This means that any changes on the displayed elements will not be drawn immediately, but on the next frame. Thus, under ordinary use, the exact time an event occurred makes no difference to the users, only the frame on which the event will be "drawn" matters. i.e. any delay of an event to match the time of the next frame is *de facto* impossible to perceive for an user, and thus assumed costless. Such operation is described in Figure 3.3 and can be trivially done in JavaScript thanks to `Window.requestAnimationFrame()`. We will assume that users typically possess 60Hz screens, i.e. that the screen draw a frame every 1/60 seconds.

As shown in Figure 3.2, delaying events to the next frame (D) increases slightly the EER value (+1.5 points). Doing both strategies (DA) produces a very small increase of privacy (+0.9 points) compared to (A). However, such strategy is still interesting as it would suppress information that could be exploited by other KDS.

In the following, the presented KDAS will be assumed to implements the DA costless strategy: pressure events will be delayed beforehand (D), and release events will be automatically generated afterwards (A).

3.2.2 Non-blocking protection

In order to further increase the EER values, some events have to be delayed beyond the next frame, thus inducing latency. Such latency might be perceivable by the users and thus constitutes a cost in terms of usability of the KDAS.

Non-blocking KDAS delays pressure events independently from the previous, with the only constraint to preserve the events' order. Their parameter p is the number of frames that can be skipped, and *de facto* their latency. In order to enable a fair comparison of KDAS, performances are evaluated at equals latencies.

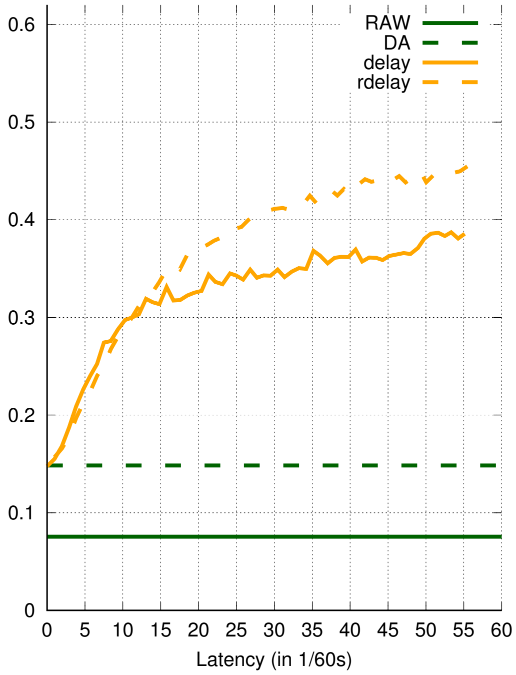
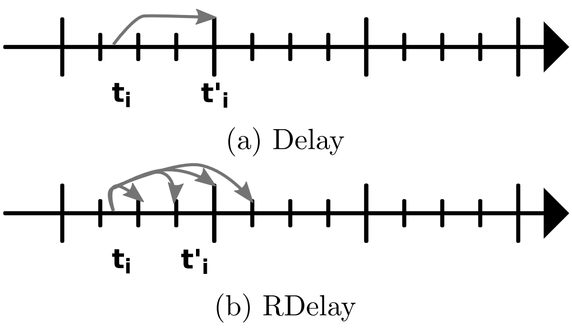


Figure 3.4: Absolute *loss* on the EER with non-blocking KDAS (in orange) in function of the latency.



(a) Delay

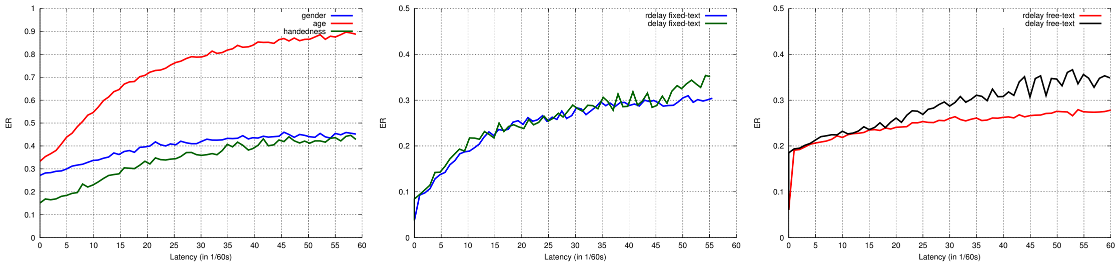
(b) RDelay

Figure 3.5: Visual representation of non-blocking strategies (p=4).

Two non-blocking KDAS are studied. First, events are discretized with a resolution of $(p+1)/60$ (delay), and in the second, events are delayed by n frames with n an uniform discrete noise $n{\sim}U(0,p)$ (rdelay). A visual representation of these two strategies are shown in Figure 3.5.

As shown in Figure 3.4, both provide significant protection compared to the costless KDAS (DA). For the same latency, rdelay seems always better than delay except for low latency ($<$ 10 frames) where delay is slightly better than rdelay.

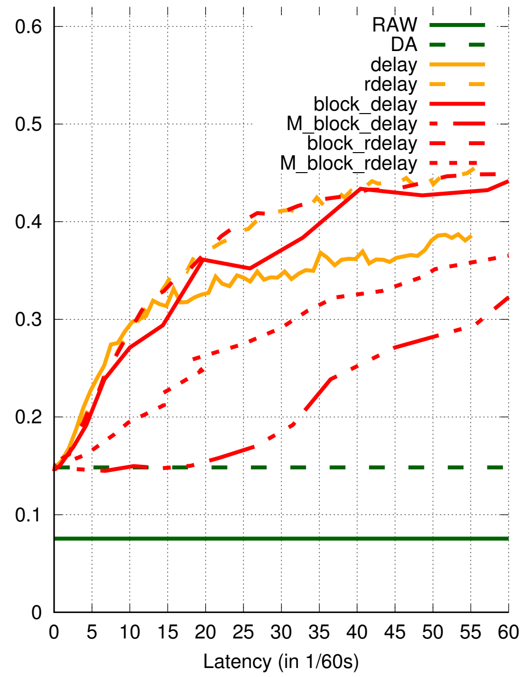
As shown in Figure 3.6a, KDAS also offers protection against soft biometrics profiling. When considering identification, as shown by Figure 3.6b, gains are not as high as for authentication. For Free-Text identification, as shown by Figure 3.6c, costless KDAS offers great gain on privacy, while gains offered by non-blocking KDAS grows slowly. delay and rdelay offers similar privacy for low latency ($<$ 15 frames). On Free-Text and for high latencies ($>$ 15 frames), delay outperforms rdelay.

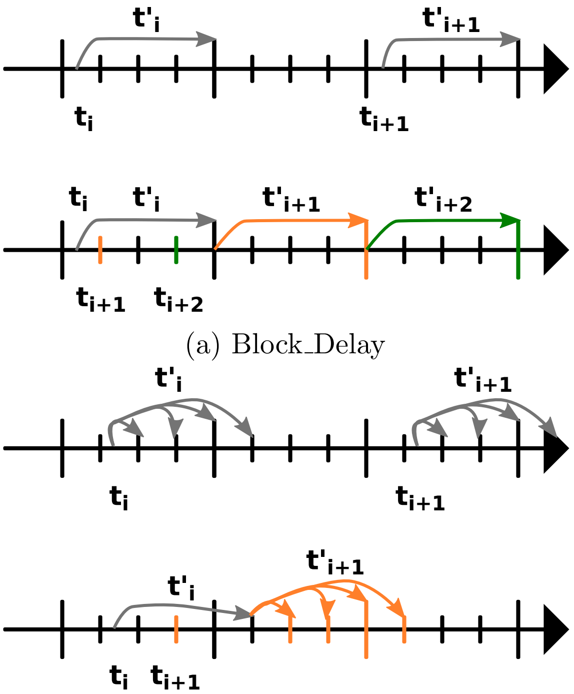


(a) Soft Biometrics (rdelay) (b) Fixed-text identification (c) Free-text identification

Figure 3.6: Non-blocking KDAS performances.

3.2.3 Blocking protection





(a) Block_Delay

(b) Block_RDelay

Figure 3.7: Absolute *loss* on the EER with blocking KDAS (in red) in function of the latency and maximal latency (prefixed by M_).

Figure 3.8: Visual representation of blocking strategies (p=4).

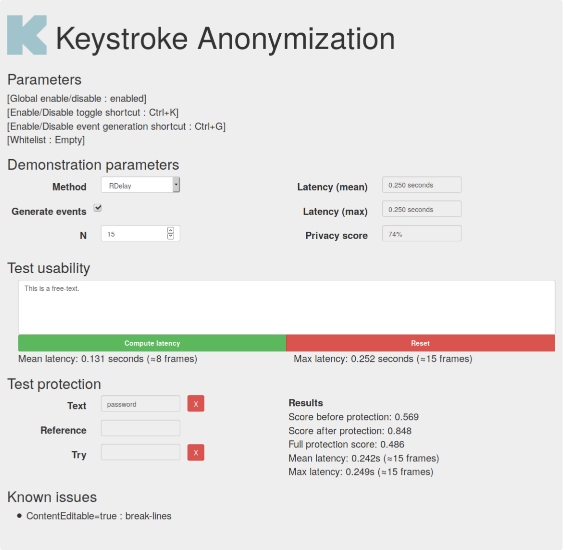
In order to continue to increase the EER value, events can be delayed depending on the previous event. The first blocking KDAS ensures that there is at least n frames between each pressure events (block_delay), the second (block_rdelay) delays them such as the i^{th} pressure event's delayed timestamp (t'_i) is computed from the original timestamp t_i as follows: $t'_i = max(t'_{i-1}, t_i) + U(0, p)$. A visual representation is given in Figure 3.8.

As shown in Figure 3.7, block_rdelay is in average slightly better than its non-blocking equivalent rdelay, and worst than delay for low latencies ($<$ 10 frames). As for block_delay, it is worst than both delay and rdelay for low latencies ($<$ 15 frames). However, when considering the maximal latency, non-blocking KDAS out-perform by far blocking KDAS.

Moreover, when users type too fast (or p too high), blocking KDAS latency adds up at each pressed key. When this happens, t'_i will only be computed from t'_{i-1}, i.e. every users will have the same way of typing, but at the cost of a non-ergonomic and unacceptable latency. Adapting p to match the user typing speed would enable browser fingerprinting attacks, as it would enable the attacker to discriminate users in function of their configuration, i.e. the p parameter. This suggests that blocking KDAS should be avoided in favor of non-blocking approaches.

3.3 Proof of concept implementation

We developed *Keystroke Anonymization*, a Firefox WebExtension, that implements the previously cited KDAS. The WebExtension was used during the writing of [Migdal and Rosenberger, 2019c] on Overleaf (method: rdelay, p=15). Users can enable/disable the protection using the Ctrl+K shortcut, and can enable/disable generation of events using the Ctrl+G shortcut.



Keystroke Anonymization

Parameters

[Global enable/disable : enabled]
[Enable/Disable toggle shortcut : Ctrl+K]
[Enable/Disable event generation shortcut : Ctrl+G]
[Whitelist : Empty]

Demonstration parameters

Method: RDelay
Generate events: ☑
N: 15
Latency (mean): 0.250 seconds
Latency (max): 0.250 seconds
Privacy score: 74%

Test usability

This is a free-text.

Mean latency: 0.131 seconds (≈8 frames)
Max latency: 0.252 seconds (≈15 frames)

Test protection

Text: password
Reference:
Try:

Results
Score before protection: 0.569
Score after protection: 0.848
Full protection score: 0.486
Mean latency: 0.242s (≈15 frames)
Max latency: 0.249s (≈15 frames)

Known issues

- ContentEditable=true : break-lines

Figure 3.9: Screenshot of the WebExtension (debug mode).

A demonstration is also integrated to the WebExtension enabling users to test usability and the protection of different configurations (see Figure 3.9).

3.3.1 Implementation issues

The manifest is a JSON configuration file used by WebExtensions. In order to make active the WebExtension on all pages, `content_script`'s `matches` field is set to

<all_url>.

The WebExtension listens on each Keyboard events in order to delay them. One important point is that the WebExtension listeners must be called *before* any other, or else the attacker will be able to block call to the WebExtension listeners, i.e. to prevent events from being delayed by the WebExtension.

For that, `content_script`'s `run_at` field must be set to `document_start`, in order to the WebExtension script to be executed before the page scripts, thus allowing it to register listeners before any else. Indeed, listeners are called in the order of their registration.

Moreover, listeners must be added on `document`, with the third parameter of `addEventListener()`, `capture`, set to true. Indeed, event propagation has two phases in JavaScript, capture and bubble. In the capture phase, events are propagated from the root element, `document`, to the target element, e.g. an input. Then, during the bubble phase, events are propagated from the target element to the root element. Thus, in order to be the first to capture the event, the WebExtension must capture it during the capture phase, on the root element. The page must be reloaded upon WebExtension installation or activation, in order to ensure to be the first to register listener on already opened pages.

Only the keydown and keyup events are listened to. If the event has be been delayed, its immediate propagation is stopped. If the event is a keydown, the event is captured, i.e. added to an array. As previously stated, delaying event must be done without active wait. Thus, requiring to stop the event propagation with `event.stopImmediatePropagation()`, and to latter re-inject it with `event.target.dispatchEvent(event)`.

As the order of pressure keyboard events has to be kept, attacker could estimate the real event timestamp by regularly generating pressure keyboard events. Untrusted events should not be delayed, and too much untrusted keyboard events in a short time should raise an alert.

The function `window.requestAnimationFrame()` is used to call an handler in order to process captured events before each frame. The frame in which each event will have to be re-injected in then computed depending on the KDAS method, and the parameter p.

However, re-injected events will loose their *trusted* status as it no longer originate from user action. This means that the event will trigger listeners but will not trigger the target default behavior, e.g. add a character on an input. This default behavior has thus to be simulated. Keyboard events that are not a character (`event.key.length != 1`), or when the ctrl key is pressed (`event.ctrlKey`) will not be delayed.

For inputs and text area, this requires to delete the current selection (between `elem.selectionStart`, `elem.selectionEnd`), insert the character between, set the cursor position (`elem.setSelectionRange(start+1, start+1)`), generate an input event, and add a listener to trigger a change event when the element loses focus. As `elem.selectionStart` and `elem.selectionEnd` are not defined for all types of inputs (e.g. email), the type of the input (`elem.type`), has to be changed to `text` while accessing and modifying these properties.

div elements can also be used to type text thanks to the `contentEditable=true` attribute. This is used, e.g. by the webmail GMail to write e-mail. For contentEditable elements, current selection must be deleted `window.getSelection().deleteFromDocument()`. The element and position in which insert the character is givent by `selection.focusNode` and `selection.focusOffset`. If the element is a div, its content must be cleared (`div.removeChild(div.fistChild)`), and a new div containing a TextNode must be appended to the first div. If the element is a TextNode, or once the TextNode created, its content is modified through `textNode.textContent`. Before creating an input event, the cursor has to be updated in the following way:

```
let range = document.createRange();
range.setStart(textNode, start+1);
range.setEnd(textNode, start+1);
range.collapse(false);
selection.addRange(range).
```

Unfortunately, the creation of new lines ignore the position of the cursor if the mouse or the arrows key has not been used since the last delayed event. Events s.a. keypress, input, change, could also not be generated when simulating the default behavior on events, to increase the privacy protection by making it more difficult to an attacker to deduce the event timestamp, however, this might impact the functionality of some websites.

3.3.2 Comparison with KeyboardPrivacy

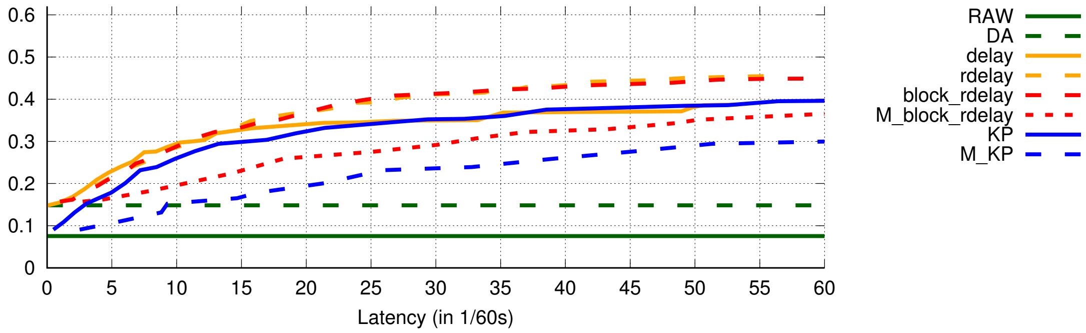


Figure 3.10: Absolute *loss* on the EER with Keyboard Privacy (in blue) in function of the latency and maximal latency (prefixed by M_).

As shown in Figure 3.10, KeyboardPrivacy is less efficient than delay and its maximal latency is worst than block_rdelay. It is even less efficient than the costless KDAS when the maximal latency is near under 10 frames. The construction of this KDAS extension seems to be ad hoc, and could be improved using the conclusion of this chapter:

- use passive waits instead of active waits;
- automatically generates release events;
- delays pressure events to the next frame;

- use non-blocking KDAS (rdelay or delay) to limit the latency;
- use fixed parameters for all users to prevent fingerprinting attacks.

It also suffers from several security vulnerabilities. Indeed, the events are captured during the bubble phase, instead of the capture phase. Moreover, the script is, by default, executed after the page has been loaded. This WebExtension also does not support ContentEditable fields. As an active wait is used to delay events, keyboard events cannot be, at the same time, protected, and used for e.g. authentication.

3.4 Conclusion and perspectives

This work constitutes a preliminary study on the Keystroke Dynamics Anonymization Scheme. Performances of presented KDAS has been demonstrated using state of the art fixed-text Keystroke Dynamics datasets presented in Chapter 2. However, performances and latency may vary depending on the written text, and the user. KDAS introduce a trade-off between performances (security) and latency (usability). The latency has been evaluated in term of duration, and should be evaluated in terms of usability / user acceptability.

Other KDS could also be tested. An hardware implementation of such KDAS, is presented in Chapter 6, in the form of a programmable USB to USB device between the keyboard and the computer. Presented KDAS techniques could be applied to other biometrics modalities, s.a. mouse events.

In short: *Protection of Keystroke Dymanics is necessary to guarantee online users privacy. However, it introduces a trade-off between privacy and usability (latency). KDAS were not fairly compared as only the bests configuration and context were tested without guarantee that such configuration and context remains the bests after protection. Implementation of KDAS, specially in the form of a WebExtension must be done with care as any configuration error might leak users Keystroke Dynamics. The fact that generated events are untrusted requires to reproduce by end the default behaviors of such events.*

Chapter 4

Using personal data in a privacy compliant scheme

The previous chapter presented methods to protect personal information from being used against users consent. This chapter presents a way to use them for authentication, without any privacy leakage.

Keywords: *Authentication; Privacy; BioHashing; Behavioural biometrics ; Personal information; Location; Keystroke Dynamics.*

Contents

Contributions presented in this chapter

- Use of BioHashing with Keystroke Dynamics;
- Use of BioHashing with Browser Fingerprints;
- Use of BioHashing with location (GPS and IP adress);
- Computing a single BioHash from multiple modalities;
- Improvement of BioHashing performances through pre-processessings;
- Secure authentication scheme based on BioHashing.

Publications

- Migdal, D. and Rosenberger, C. (2019d). My Behavior is my Privacy & Secure Password ! In *Cyberworlds (B - Core)*, Kyoto, Japan.
- Migdal, D. and Rosenberger, C. (2018c). Towards a Personal Identity Code Respecting Privacy. In *International Conference on Information Systems Security and Privacy (ICISSP)*, Madeira, Portugal.
- Migdal, D. and Rosenberger, C. (2018b). Protection de données personnelles pour la sécurité sur Internet. In *Atelier sur la Protection de la Vie Privée*, Porquerolles, France.
- Migdal, D. and Rosenberger, C. (2017). Vers un Code Personnel d'Identité Respectueux de la Vie Privée. In *CORESA*, Caen, France.

4.1 Introduction

In Chapter 2, we stated that, when browsing the Internet, users disclose information that enable their profiling (authentication, identification, soft biometrics), even without the users consent or knowledge. In Chapter 3, we proposed strategies to prevent usage of such information without the users consent or knowledge.

The new GDPR regulation establishes rules in order to protect user privacy and ensure its consent. These modalities have thus to be used by the service to authenticate users, but, as possible, without knowing or enabling an attacker to know, these modalities. In this chapter, we thus propose an authentication scheme based on personal data without any privacy leakage.

Many studies propose strong user authentication based on biometric modalities. However, they often either, assume a trusted component, are modality-dependant, use only one biometric modality, are reversible, or does not enable the service to adapt the security on-the-fly. We propose in this chapter the concept of Personal Identity Code Respecting Privacy (PICRP), a non-cryptographic and non-reversible signature computed from any arbitrary information. We then propose an implementation of PICRP with the use of Keystroke Dynamics, IP and GPS geo-location and optimize the pre-processing and merging of collected information. We demonstrate the performance of the proposed approach through experimental results before presenting an example of its usage.

Note: *This work being prior to the previous chapters, the Fair Evaluation Scheme presented in Chapter 2 is not used in this chapter. The configuration and context used in the chapter is equivalent to BioHashing.*.03.*@st.1.1. Still, the presented metrics on Keystroke Dynamics should be recomputed, using the fair evaluation scheme.*

4.1.1 State of the art

Biometric authentication is a well-studied subject in the literature, however, proposed solutions often either, assume a trusted component, are modality-dependant, use only one biometric modality, are reversible, or does not enable the service to adapt the security on-the-fly. Trusted computing using secure element or sensors often gives the best security, but requires the possession of a specific hardware that a user might not possess. Such solutions thus only protect owner of such specifics hardware, that might be lacking in desktop or laptop computers. It also assumes that such devices are trusted and cannot be attacked.

Homomorphic or Functional encryption [Tian et al., 2018] enables to compare biometric modalities in the encrypted domain, i.e. without having the knowledge of the content. However, the encrypted data often require a lot of memory space, that can be repellent for a web service. Other solutions can be mono-modal or built for a specific modality. Some of which, like Zero Knowledge Protocol [Saini and Singh, 2018], does not enable the change of the security level on the fly.

Other solutions are based on the generation or extraction, of a fixed key from the modalities, s.a. in Fuzzy vaults, fuzzy commitment, or fuzzy extractor, often based on error-correcting code. This assumes that variations in the modality are errors to be corrected in order to produce or extract the secret key. However, the security level is often set at the creation of the secret key and cannot be changed afterward.

The usage of a trusted device can also enable a secret key unlocking, when the user is authenticated. The security level can easily be changed on the fly by configuring such trusted device. However, the security remains only on the assumption that the trusted device is assumed secure and that no attacker will ever crack it. This is a strong security assumption, even if the trusted device is tamper resistant, or has a tamper response. The use of a trusted device should not dispense from protecting the biometric information inside the trusted device when it is possible.

Finally, other solutions computes non-invertible soft hashes that conserve distances. The authentication decision can then be based on the distance between the references and the given sample, enabling to adapt the security on-the-fly, and to build more complex security policies.

4.2 PICRP

4.2.1 Principle

The issue we want to address in this work is the possibility to answer to Internet services applications (s.a. authentication, attacks detection) while preserving the user privacy. From collected personal data, we aim at generating a binary signature as dynamical user characteristics having lost its semantic description. Finally, the service is able to exploit this signature without knowing the information used to generate it.

The goal of the proposed method is to compute a binary code linked to an user from personal information (technical and biometrics). This code must answer several requirements:

- *Non reversibility*: the binary code associated to an user must not give any information about the collected personal data.
- *Confidentiality*: the attribute value cannot be known, nor deducted, even by the service (considered here as honnest but curious).
- *Unlikability*: two binary codes computed from different secrets should not reveals their similarity.
- *Similarity conservation*: if users' personal information are similar, then their binary code must be too (Hamming distance).
- *Non-usurpation*: a tiers cannot forge a code enabling him/her to usurp legitimate users' identity.
- *Revocation*: a legitimate user must be able to revoke an existing binary code.
- *Discriminant:* different users should have different codes.

- *Stable:* codes computed for a same user (and with the same secret) are similar.
- *Not-costly:* in terms of memory, computation time, ergonomics, financial cost.

In the scope of chapter, a trust score can be computed from the Hamming distance between the proof and the commitment, both fixed-size binary vectors. Therefore, we consider and detail the following personal information modality:

- *What the user is/knows to do*: its behavioural biometric;
- *What the user has*: its browser;
- *Where the user is*: its physical and organizational localization;
- *What the user "prefers"*: personal machine configuration.

A password is used as a secret key [Lacharme and Plateaux, 2011]. In this case, the user, by inputting its password, consents to give the binary code to the service. The different computation steps are introduced later.

We thus introduce the concept of Personal Identity Code Respecting Privacy (PICRP) as a cancelable non-reversible binary code enabling the similarity comparison of arbitrary private data through the Hamming distance of two PICRP (e.g. a reference and a sample). PICRP follows the previous defined requirements and can be computed from several different modalities. Any type of personal information can be added to the PICRP as long as it can be represented as a fixed-length real vector (e.g. browser history, free-text, mouse, ...). Soft-biometrics information (s.a. age, gender) could also be computed from existing modalities and integrated to the PICRP, in order to improve performances.

In this chapter, we implement the PICRP concept as a BioCode computed from private data, with the BioHashing algorithm [Teoh et al., 2004], presented in the next section.

4.2.2 Biometric protection

Biometric data are sensitive information that cannot be easily renewed, and thus needs to be protected. Indeed, if a biometric modality is stolen, e.g. a fingerprint, an attacker will be able to build fake fingerprints in order to fasly authenticate. Contrary to passwords, biometric modalities cannot be easily changed, requiring either a change in behavior, or a transplant.

Moreover, knowing the biometric modality, the attacker can also attack user privacy, either by deducing personal data (s.a. gender, age), or by using it to track the user.

Locality-Preserving Hashing

Usually, hashes algorithms, s.a. SHA256 or SHA3, are used to transform arbitrary vector of data into a binary vector of fixed length called hash. As the hashing algorithms are deterministic, i.e. the same input will produce the same hash, and as the original data cannot be retrieved from the hash, it is widely used to enable comparison of data without revealing them.

For example, it is common for password authentication to store the hashed password into the server upon user registration. Then, for verification, the user sends its password, and the server computes its cryptographic hash. If the computed hash is equal to the stored one, the user is authenticated. Hashing passwords to store them is a common practice that *theoretically* prevents any attackers to retrieve the original passwords, even if the server dataset is leaked.

The original data could be retrieved from the hash by a brute force attack, i.e. hashing all possible data and see if their hashes is equal to the attacked hash. Such attack is in practice impossible without any heuristics, mainly due to the fact that humans are not much inventive when choosing their passwords. Indeed the number of possibility is such that it would requires an non-realistic computation power and computation time.

However, hashes does not conserve the similarity of the data used to compute them, i.e. two non-equal similar data would produce very different hashes. As the modality we use are mainly biometric data, they suffers from little variations from one acquisition to another. We thus need soft hashing, a kind of hash function that conserves distances. As the conservation of similarity enables hill-climbing attacks, any attacker, being able to obtain a distance score between a given hash and hashes computed from arbitrary data, might be able to retrieve the original data from the hash.

More particularly, we are interested in Locality-Preserving Hashing (LPH) which is defined as the function $\mathcal{H}$ s.a.:
$dist_1(A,B) < dist_1(B,C) \implies dist_2(\mathcal{H}(A),\mathcal{H}(B)) \lesssim dist_2(\mathcal{H}(B),\mathcal{H}(C))$. In our cases, $dist_2$ is the Hamming distance. Some LPH algorithms assumes $dist_1$ to be an Hamming distance, i.e. the distance is defined as the number of differences between two vectors. This does not match our needs as in biometric data variations are generally distributed on all values.

In biometrics, two LPH are commly used, BioHashing [Teoh et al., 2004], and BioPhasor [Teoh and Ngo, 2006]. A new LPH derived from BioHashing, GREY-CHashing, [Atighehchi et al., 2019], has also been proposed very recently. In this thesis we choose to use the BioHashing algorithm, as presented below.

BioHashing

Biohashing is a well-known algorithm in biometrics. It enables a biometric data transformation when represented by a fixed-size real vector. It allows the generation of a binary model called BioCode having a size inferior or equal to the original size. This transformation is non-reversible and allows to keep input data similarity. This algorithm originally has been proposed for face and fingerprints by Teoh *et al.* in [Teoh et al., 2004]. Biohashing algorithm can be used on every biometric modality, or personal information, that can be represented by a fixed-size real vector. This transformation requires a secret linked to the user. In our case, this could be a password input by the user [Lacharme and Plateaux, 2011]. The BioCode comparison is realized by the computation of the Hamming distance. The BioHashing algorithm transforms a parameter vector $T = (T_1, \dots T_n)$ into a binary model called BioCode $B = (B_1, \dots B_m)$, with $m \leq n$, as follows:

1. m random orthonormal vectors $V_1, \dots, V_m$ of length n are generated from a secret used as a seed for random draw (typically with the Gram Schmidt algorithm).

2. For $i = 1, \dots, m$, compute the dot product $x_i =< T, V_i >$.

3. BioCode computation $B = (B_1, \dots, B_m)$ with the quantization process:

$$B_i = \begin{cases} 0 & \text{if} \quad x_i < \tau \\ 1 & \text{if} \quad x_i \geq \tau, \end{cases}$$

   Where τ is a given threshold, generally equals to 0.

The algorithm performance is granted by the dot product with orthonormal vectors, as detailed in [Teoh et al., 2008]. The quantization process guarantees the data non-reversibility (even if $n = m$), as each input coordinate T is a real value, when the BioCode B is binary. We propose the use of this transformation to protect personal information.

The BioCode being a simple hash, it is vulnerable to replay attacks, and thus needs to be integrated into a secure protocol. When sending it e.g. to a server for authentication, it should be transmitted through a secure communication channel, s.a. a TLS connection. A secure protocol is described in Section 4.6, and the use of trusted device is proposed in Chapter 6. To evaluate the performance of the proposed PICRP, the BioHashing secret is assumed to be known by the attacker (worst case). All PICRP were thus be computed using the same BioHashing secret: `0x1534FA2C4D37`. In the next section, we present the PICRP pipeline we used in this study.

4.2.3 PICRP pipeline

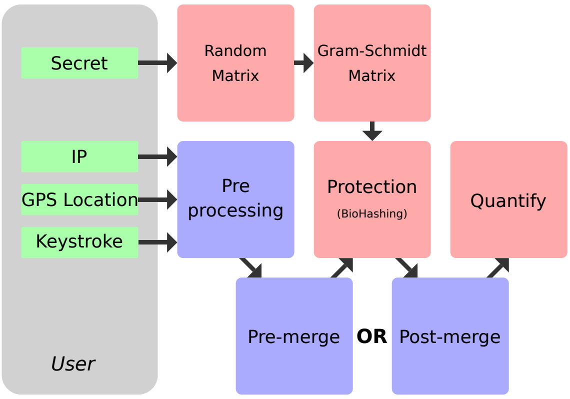


Figure 4.1: PICRP computation pipeline.

The computation of PICRP is shown by Figure 4.1. In red, the BioHashing steps that had been previously presented. In green, the data used to compute the PICRP, and in blue, the steps added to the BioHashing. Data and additional steps are presented in the following sections.

Modalities are pre-processed in order to be converted into fixed-length vectors of real and then merged to produce a unique PICRP. Merging can be performed before (pre-merge) or after (post-merge) protection (using BioHashing). The secret is a modality, or set of modality, that is not subject to variations, s.a. a password, a MAC address, etc.

PICRP can be viewed as a binary interface as its computation does not influence its usage, and vice versa. This gives the freedom for the user to easily use any arbitrary modalities. We also argue that the computation of the PICRP, in the context of a browser, should be performed by a WebExtension, to give to the user control over the PICRP computation.

Computations of the PICRP can thus be improved without any impact on its usage, facilitating adoption of PICRP improvements. Usages can also be added, without any modification on its computation process. For example, one could imagine, instead of authentication, to use PICRP to generate or extract private keys.

4.3 Datasets

4.3.1 Keystroke Dynamics Datasets

The datasets used in this chapter are presented in Chapter 2, in Tables 2.1 and 2.3. These tables are repeated below. Contrary to Chapter 2, only the first entry are used as reference, and the 44 nexts, as samples. As usual, metrics given in this chapter are computed as the average value of the metric across all datasets.

Name	**Text**	**# of users**	**Source**
GREYC K	greyc laboratory	104	[Giot et al., 2009]
GREYC W1	laboratoire greyc	62	[Giot et al., 2012]
GREYC W2	sésame	46	[Giot et al., 2012]
CMU	.tie5Roanl	51	[Killourhy and Maxion, 2009]

Table 2.1: Description of used Same/Fixed-text datasets.

Name	**About**	**# of users**	**Source**
KPPDW1	Gay Marriage	400	[Banerjee et al., 2014]
KPPDW2	Gun Control	400	[Banerjee et al., 2014]
KPPDW3	Restaurant Reviews	500	[Banerjee et al., 2014]

Table 2.3: Description of used Free-Text datasets.

4.3.2 Location datasets

Location datasets are generated from the DBIP dataset [DB-IP, 2019] where each entry describes an IP network and a GPS location. Only IPv4 entries are considered. Each user is associated to an origin place, randomly chosen among the DBIP entries, each entry having a probability to be chosen given by the number of IP addresses the network enables. Each user entry is then generated by randomly choosing another entry from DBIP which distance with the origin place is below an arbitrary value we name *user mobility.* The generated datasets have 100 users with 45 entries each. IP addresses are randomly chosen among the one belonging to the IP network. Two datasets are generated, *IP addresses generated from place* where each entry IP addresses are generated from its network, and *IP addresses from network*, where the IP address is generated from the origin place network.

GPS coordinates are converted in XYZ location described by 3 reals. In *XYZ location, generated from places*, XYZ locations are computed from the DBIP entries GPS coordinates. In *XYZ location, generated from positions*, XYZ locations are randomly picked in all possible coordinates at a distance from the origin place inferior to the user mobility. XYZ location is a coordinate in the Euclidean space and enables non-biased distances, as longitude can go to +180 to -180, and that in function of the latitude, differences in longitudes do not correspond to the same distance.

4.3.3 GPS formula

This section presents formula applied to GPS/XYZ locations used in the scope of this study.

XYZ locations

XYZ locations (x,y,z) are computed from latitude (lat) and longitude (long) GPS coordinates with the following algorithm:

```
gpsToXYZ([lat, long]) : [x,y,z]
    $lat \mathrel{*}= \pi / 180,$                    $long \mathrel{*}= \pi / 180;$
    $y = 0.5 + sin(lat) * 0.5,$          $r = cos(lat) * 0.5;$
    $x = 0.5 + sin(long) * r,$           $z = 0.5 + cos(long) * r;$
```

Distances

Distances between two places are computed as an angle a using the cosinus law. The distance between two XYZ locations A, B is computed as follow:

```
angle(A,B): a
    $cos^{-1}(1 - 2 * (\Sigma_{i \in \{x,y,z\}}(A[i] - B[i])^2))$
```

Distances in meters m are converted into angle a distance with the following formula (assuming the circumference of the earth c to be 40,075,000 meters):

$$a = m/c * 2 * \pi$$

Random locations

Random locations are generated from an origin location o and a user mobility r, i.e. the distance between the random locations and o is at most r. Random locations are generated by randomly picking a polar coordinate $[a, d]$ in a circle of radius r using *pickInCircle()*. The polar coordinate are then converted to a GPS location $gps = [long, lat]$ using *pointInCircleToGPS()*. Random locations are generated, first assuming the origin location to be the North Pole (lat. 90, long. 0), then by rotating the space in order to move the North Pole to the origin location using *moveNorthPoleToOrigin()*.

```
pickInCircle(r): [a,d]
    $a = rand() * 2 * \pi,$        $d = r * \sqrt{rand()};$
pointInCircleToGPS([a,d]): [long,lat]
    $lat = 90 - 180 * d/\pi,$        $long = -180 * a/\pi + 180;$
moveNorthPoleToOrigin(gps, o = [olong, olat]): gps
    $gps = latRotation(gps, (olat-90) / 180 * \pi);$
    $gps = longRotation(gps, olong / 180 * \pi);$
latRotation(gps, dx): gps
    $[x, y, z] = gpsToXYZ(gps);$
    $a = angle([0.5,y,z], [0.5, 0.5, 1]) + dx;$
    $r = dist3D([x,y,z], [x,0.5,0.5]);$
    $y = 0.5 + r * sin(a),$        $z = 0.5 + r * cos(a);$
longRotation(lat, long], dy): gps
    $long += dy / \pi * 180;$
    $long = (long + 360 ) \% 360 - 360;$
Dist3D(A,B): d
    $d = \sqrt{\Sigma_{i \in \{x,y,z\}} (A[i] - B[i])^2};$
```

4.4 Pre-processing

We intend to protect collected data with a biometric template protection scheme called BioHashing. Collected information thus have to be pre-processed in order to be represented as a vector of real values:

- *Fixed-text Keystroke Dynamics* data are pre-processed in Section 4.4.1. In this section, we show that the values distributions in the real vector influence the BioHashing performances.
- *Locations* (GPS and IPv4 addresses) are pre-processed in Section 4.4.2. In this section, we reduce collisions in the final BioCode by extending small vectors of reals.

Free-text Keystroke Dynamics are also considered and pre-processed, but are not integrated to the final PICRP.

4.4.1 Fixed-text Keystroke Dynamics

Keystroke dynamics can be trivially represented as a concatenation of dwell (d_0/d_5) and flight times (d_3). However, such representation (raw) gives disappointing performances (EER=40%). As stated in Chapter 2, 6 duration times can be extracted from each digraph, with a shared duration between two consecutives digraphs $d_5(u, e, d) = d_0(u, e, d + 1)$. However, as these duration times can be rewritten as additions of dwell and flight times, they are, by construction, not bringing any additional security or performance to the BioHashing algorithm. We thus present, in the following, several pre-processing techniques to the raw representation of Keystroke Dynamics that improve performances (some of them have already been presented in Chapter 2).

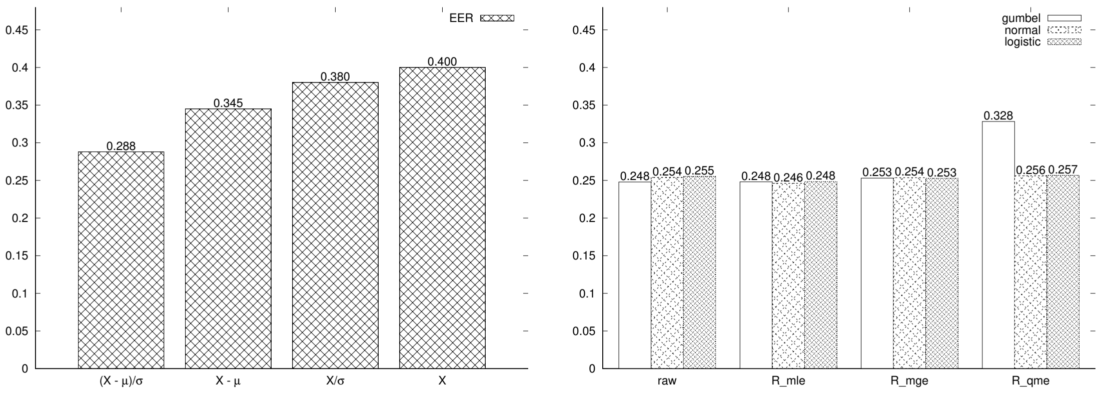
Standardization

A common practice in Data Sciences is to normalize variables, i.e. to center and reduce them. Assuming X_i the variable associated to the i^{th} real of the vector, with μ_i and σ_i its mean and standard deviation, those processes are described by the following formulas:

- *center*: $X'_i = X_i - \mu_i$;
- *reduce*: $X'_i = X_i/\sigma_i$;
- *standardize*: $X'_i = (X_i - \mu_i)/\sigma_i$;

However, in this study, we used the median value instead of the mean one. Indeed, the median is more resilient to aberrant values (e.g. hesitation times), and ensures equal numbers of positives and negatives values after centering.

Indeed, due to the construction of the BioHashing algorithm, two opposed real vectors produce opposed binary vectors. Thus, centering variables is expected to help BioHashing to cover the binary vectors space, thus reducing collisions, and improving performances.



(a) With standardization.

(b) With uniformization

Figure 4.2: Influence of preprocessing on BioHashing performances.

Results: As shown in Figure 4.2a, we found that Standardization significantly improves the EER value (28.8%) compared to the raw (40%), reduced (38%), and centered (34.5%) cases.

Uniform distribution modelling

Another practice is to change the variable distribution. As the previous section shown that centered variables seem to significantly improve the EER value, we choose a target distribution that is centered. We also seek to draw closer extrema values, and to distance closed values. For these reasons, we choose the target distribution to be, in this study, a uniform distribution with support $[-1; 1]$. Change the distribution of a variable can be easily performed with the following formula: $X'_i = cdf'^{-1}_i(cdf_i(X_i))$, with $cdf_i(X_i)$ the Cumulative Density Function describing the distribution of the variable X_i, and $cdf'_i(X'i)$, the target distribution. However, while the target distribution is known ($cdf'^{-1}_i(x) = 2x - 1$), the variable distribution $cdf_i(X_i)$ has to be estimated.

As already stated in Chapter 2, a naive estimation of X_i distribution is given by $cdf_i(x) = pos(x, A_i)/len(A_i)$, with x a value of X_i, A_i a sorted array of all known values of X_i, $len(A_i)$ its length, and $pos(x, A_i)$ the position of x in A_i. A more practical estimation of $cdf_i(X_i)$ is to compute the parameters of the law X_i is assumed to follow. In this chapter, we tested 4 laws (gumbel, normal, logistic, and laplace) and 4 fitting functions (raw, R_mle, R_mge, and R_qme) that are latter described in Chapter 5. All dwell times were assumed to follow the same law, but with different parameters, as well for the flight times. Dwell and flight times could however follow different laws. Configurations are labeled as follow: *fitness function*.*dwell law*.*flight law*.

Results: As shown in Figure 4.2b, over the 64 tested configurations, the optimal EER value (24.2%) was found with R_mle.normal.gumbel (fitting). The best raw estimation configuration, raw.gumbel.gumbel (estim), was found slightly better (24.8%) than the naive estimation (25.4%).

Discretization

As already stated in Chapter 2, previous studies have shown the influence of discretization on performance of Keystroke Dynamics Systems based on an Hocquet distance (up to $\simeq -0.5$ points) [Migdal and Rosenberger, 2019c], and on SVM (up to -1.04 points) [Giot et al., 2011]. To the knowledge of the authors, none has yet study the impact of Keystroke Dynamics discretization on BioHashing-based KDS. Keystroke Dynamics data were discretized and uniformized into identical probabilities values using the following formula: $X'_i = cdf'^{-1}_i(disc_n(cdf_i(X_i)))$, with $disc_n(x) = \lfloor x * n \rfloor$ / (N-1), and n the desired number of discrete values. $\lfloor n \rfloor$ is assumed equal to $n - 1$.

Results: Uniform Keystroke Dynamics data have been discretized using 999 different values of $n \in [\![2, 1000]\!]$. The EER value is computed as the lowest EER obtained from the 999 discretization configurations. As shown in Figure 4.4, discretization produces negligible EER gains (-0.1 to -0.4 points). Figure 4.3 shows the impact of discretization on the 4 datasets. As the gains is negligible, and as the best descretization threshold seems to be dataset-dependant, it is unclear whether the gain is due to chance or not.

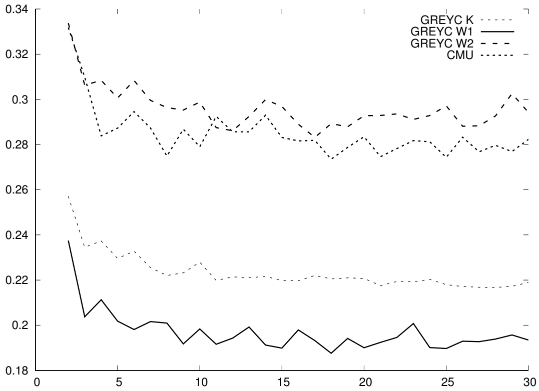


Figure 4.3: Impact of discretization on the BioHashing performances

Discussion

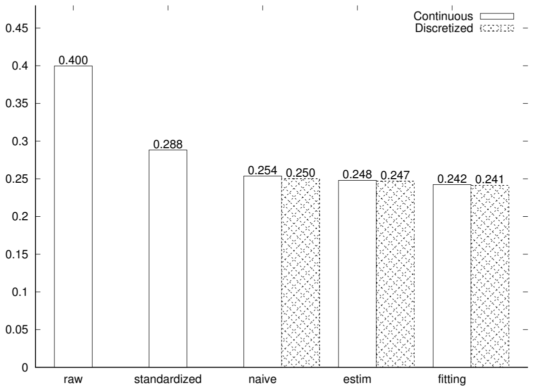


Figure 4.4: Keystroke Dynamics pre-processing performances (EER)

As shown in Figure 4.4, uniformization of Keystroke Dynamics greatly improves EER (24.2% to 25.4%) compared to raw (40%) and normalized (28.8%) values. Discretization however produces negligible EER gains (-0.1 to -0.4 points). In addition to being less efficient (+1.2 points), naive estimation of $cdf_i(X_i)$ requires headcounts of each possible values for each variable. As the BioCode is computed on the client side, this means a large sending and storing large amount of data. Assuming a fixed-text of 16 characters with 1000 possible values for each variable, this represents an increase of at least 248ko in the webpage that can be troublesome for small Internet connections. In the contrary, non-naive estimations of $cdf_i(X_i)$ only requires mean/median and standard deviations that represent less than 0.5ko, assuming a fixed-text of 16 characters. While raw estimation of $cdf_i(X_i)$ is less efficient than fitting estimation (+0.6 points), it may be more practical as mean/standard deviations can easily be computed and updated, storing, for each variable, only the number, sum, and squared sum of its known values.

Although the comparison is not fair (different configuration and context), protection introduces a high cost in performances (24.2% against 7.9% without).

With Free-Text

Free-Text were also considered with BioHashing, although we do not use it in our final PICRP. The feature vector is composed of 25 quantiles extracted from durations d_0 and d_1. Only standardizations and naive uniformization pre-processings were tested.

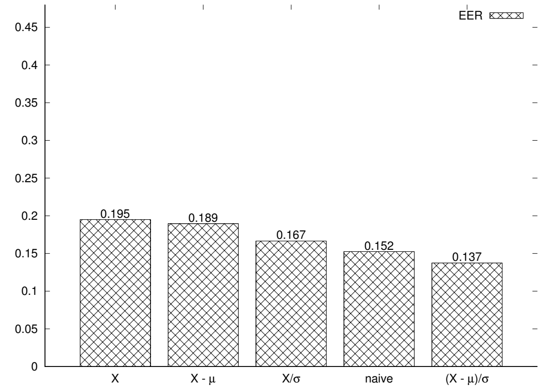


Figure 4.5: Free-Text Keystroke Dynamics with BioHashing.

Results are shown in Figure 4.5. Surprisingly, the best pre-processing is the standardization (EER = 13.7%) ahead of naive uniformization (EER = 15.2%). Raw performances are quite good (EER = 19.5%) compared to Fixed-Text (EER = 40%). After pre-processing performances are still better than Fixed-Text (EER = 24.2%) by 10.5 points, but still higher than without protection (EER = 5.6%) by 8.1 points.

Limits

In the previous sections, we assumed that all users are asked to type the same fixed text in order to authenticate themselves. However, in real life, they would be more likely to be asked to type an identifier, s.a. their login or e-mail address, which is a personal fixed-text known by others. However, as the pre-processing parameters depends on the content being typed, this would require the service to make hundred of users type each possible/used fixed-text in order to compute them.

Pre-processing parameters can be estimated by assuming that same parameters apply to all dwell (or flight) times, enabling to compute them from known dwell

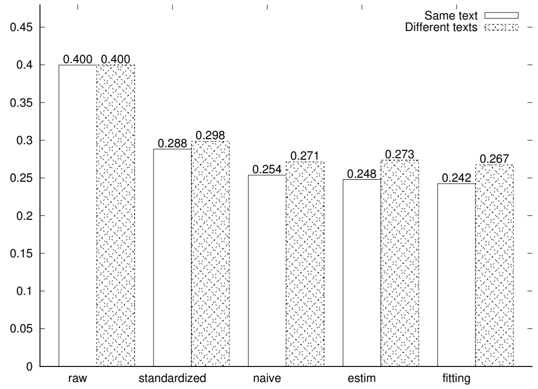


Figure 4.6: Keystroke Dynamics pre-processing performances (EER)

(or flight) times. Figure 4.6 shows that this assumption induces a significant loss of EER value (+1 to +2.5 points). The best estimation and fitting configurations under this assumption were found to be raw.normal.laplace and R_mle.normal.laplace. It is worth noticing that the naive estimation of $cdf_i(X_i)$ outperforms estim and fitting under this assumption. Pre-processing parameters might be estimated with better accuracy using additional knowledge on Keystroke Dynamics. E.g. computing parameters in function of the typed character, digraph, or tri-graph either by knowing their statistics, their position on the keyboard, or/and their frequency in the user language.

BioHashing should not enable further profiling, e.g. assessment of the age or gender, although this should be tested.

4.4.2 Location

IP address are represented as a set of bits some of them being more significant than others, while XYZ locations are represented as a set of 3 real values. Bitfields b on length n can easily be converted into reals r (and vice versa) thanks to the following formula: $r = \Sigma_{i=0}^{n-1} b_i * 2^{-i-1}$.

Using the real representation, IP address and XYZ locations are represented by 1 and 3 real values, thus producing BioCode of 1 and 3 bits. 4 bits, i.e. 16 possible BioCodes, is obviously not enough for both performances and security reasons. We thus present, in the following, several pre-processing techniques extending small vectors of reals in order to improve BioHashing performances. Real values will be assumed to equate to a bitfield of 32 bits, and will be transformed as vectors of 32 real values R, while ensuring that the most significant bits have the most weight.

LogDist

LogDist does not ensure the bits significance. It associates each real value R_i to a bit b_i that determines its sign. In order to ensure that, e.g. bitfields 0111 (=0.4375) and 1000 (=0.5) have similar representations, the amplitude is computed from the following bits (viewed as a real). The amplitude is computed so that, the more such real is close to 0 or 1, the more the amplitude is close to 0, and the more it is close to 0.5, the more the amplitude is close to 1: $R_i = (-1)^{b_i} * (3^{b_{i+1}} - 1 + (-1)^{b_{i+1}} \Sigma_{j=1}^{n-i-2} b_{j+i} * 2^{-j}$).

PrefixDist

PrefixDist associates to each real R_i a bit b_i whose sign is determined by b_i as well as the previous computed real: $R_i = (-1)^{b_i} * R_{i-1}$. In this way, two bitfields sharing their n most significant bits produce vectors sharing at least n real values.

PrefixHash

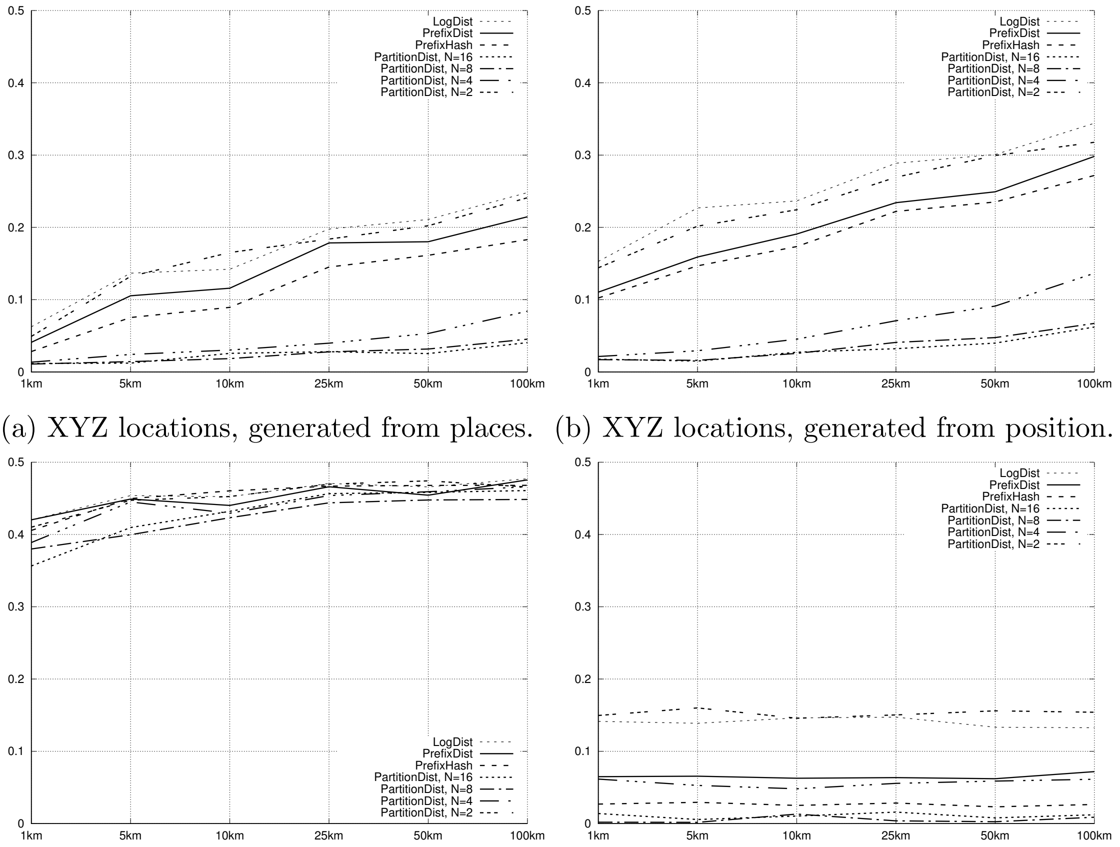
PrefixHash is a variant of PrefixDist. It associates each real R_i to an hash computed from all bits $b_{\{j \leq i\}}$: $R_i = H_i * 2^{-31} - 1$, with H_i a 32-bit hash. The hash is computed as $H_i = \mathcal{H}(H_{i-1}, b_i)$, with $\mathcal{H}$ the hashing function. As the hashing function is not used for its security properties, but to diversify the output, Java `hashCode()` algorithm is used in this study: $\mathcal{H}(IV, c) = IV << 5 - IV + c$. The first IV (i.e. H_{-1}) is computed from the BioHashing secret.

PartitionDist

PartitionDist generates 32 reals R_i by combining N bits from 32 bits b_i while preserving their significance: $R_i = (-1)^{f_i(b,0)} * \Sigma_{j=1}^{N-1} f_i(b,j) * 2^{-j}$. $f_i(b,j)$ equals to b_r with r chosen between $j * 32/N$ and $(j+1) * 32/N - 1$. All combinations of the $n = \lceil log(32)/log(32/N) \rceil$ first bits are listed. 32 of them are randomly selected, each determining the $f_i(b,j)$ bits for $j < n$. The other $f_i(b,j)$ bits are randomly selected. Random engines are initialized from the BioHashing secret.

Results

Figure 4.7 shows the performance of the previously described location pre-processing methods. In all 4 tested datasets, PartitionDist, N=16, and PartitionDist, N=8 outperform other pre-processing methods. IP addresses generated from places provides an unacceptable EER value ($\geq$ 40% for a user mobility $\geq$ 5 km) whereas IP addresses generated from networks provides a great EER value ($0.17\% < EER < 1.6\%$). PartitionDist, N=8 is the best setting for IP addresses pre-processing methods. In real-life, users do not use all networks from a place, like in IP addresses generated from places, but might still use several networks. This suggests the need of several templates, e.g. one template per network the user connects to. Further studies should be conducted with real-life data. XYZ locations provide great EER values both generated from places ($1.08\% < EER < 4.53\%$) and from positions ($1.49\% < EER < 6.71\%$). PartitionDist, N=16 is the best of XYZ locations pre-processing methods. Contrary to IP addresses, XYZ locations does not need several templates.



(a) XYZ locations, generated from places. (b) XYZ locations, generated from position.

(c) IP addresses, generated from places. (d) IP addresses, generated from networks.

Figure 4.7: Location pre-processing performances (EER) in function on users mobility.

4.5 Merging of pre-processed data

.

A naive way to merge personal data is to compute a BioCode for each and concatenating them. However, by doing so, some BioCodes might be too small to be protected against brute force attacks. For example, an IP address has less than 2^{32} possibilities and its resulting BioCode cannot have more than 32 bits, i.e. 2^{32} possibilities. We present in this section new merging methods, applied on the 3 previously pre-processed modalities:

- XYZ locations, generated from places: PartitionDist, N=16;
- IP addresses, generated from networks: PartitionDist, N=8;
- Keystroke: R_mle.normal.gumbel.

In both merging methods, each modality vector m_i is associated to, and multiplied by, a positive weight w_i. In the first method, vectors are weighted and concatenated before applying the BioHashing algorithm. In the second method (post), a BioCode is computed on each modality. Before the BioHashing quantification step, a new vector v is computed as a weighted mean of the 3 non-quantification BioCode $m_{\{0,1,2\}}$ with the following formula: $v_i = 1/(\Sigma_{j=0}^{2} w_j)\Sigma_{j=0}^{2} w_j m_j[i\%len(m_j)]$. v length is computed

	Pre	EER (%)	Post	EER (%)
min(0)	pre.len.1.98	0.069	post.raw.14.85	0.077
min(K)	pre.len.19.80	0.102	post.raw.15.84	0.203
min(X.IP)	pre.raw.10.6	20.00	post.raw.3.36	21.78
min(K,X.IP)	pre.raw.13.25	20.39	post.raw.3.38	21.92
min(IP.K,X.K,X.IP)	pre.len.47.25	21.42	post.raw.34.26	24.14

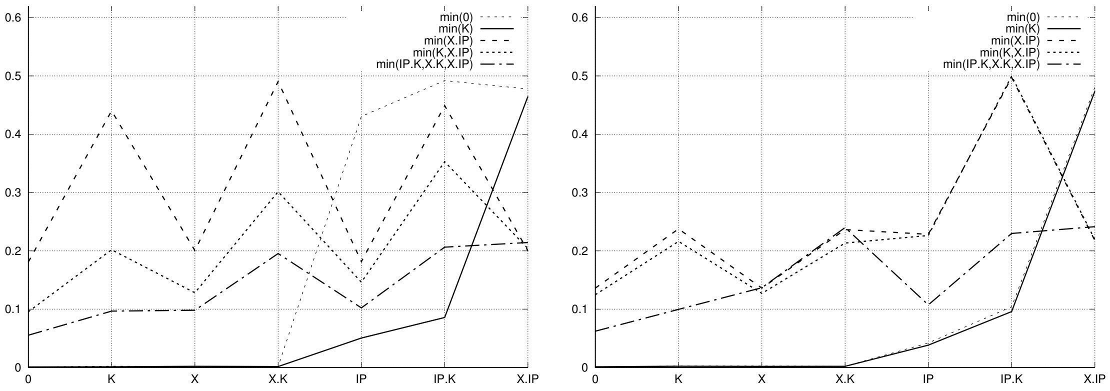
Table 4.3: Best merging configuration performances.

as the length of the longer BioCode. BioHashing quantification step is then applied on the resulting vector v.

Merging methods are evaluated through 7 scenarios, labelled as the concatenation of modalities known/stolen by the attacker among the Keystroke (K), XYZ location (X), and IP addresses (IP). The scenario in which the attacker has no knowledge is 0. All possible combinations of weights have been tested. Weights were chosen from 1% to 98% per step of 1 point so that their sum is 100%, then multiplied by the number of modalities (here 3). As modalities vectors have different lengths and amplitudes, weights were multiplied to 3 modifiers: raw (no modification), len (correct the influence of modalities length), alen (correct the influence of both modalities length and mean amplitude). Modifiers are computed as follows:

- raw: $raw_i = 1$;
- len: $l_i = 1/(len(m) * len(m_i)) * \Sigma_{j=0}^{len(m)-1} len(m_j)$.
- alen: $la_i = l_i/ampl(m_i)$,
  with $ampl(m_i) = 1/len(m_i) * \Sigma_{j=0}^{len(m_i)-1} |m_i(j)|$.

Merging configuration are labeled as:
$\{$pre$|$post$\}.\{$raw$|$len$|$alen$\}.w_0.w_1$.



(a) Pre merging.

(b) Post merging.

Figure 4.8: Best merging configurations performances under 7 scenarios.

Results: Table 4.3 shows the best pre and post merging methods configurations that minimize the EER value in the scenario indicated in the first column, if several scenarios are indicated, the configuration minimizes the maximal EER value of each scenario. Figures 4.8a and 4.8b show the performances of each configuration under each scenario. As shown in Figures 4.8a and 4.8b, configurations that minimize the EER value under (0) scenario produce really great EER ($< 1\%$), however such scenarios poorly perform (EER $> 40\%$) if IP addresses and XYZ location are known by attackers. Other configurations ensure an EER value $\lesssim 20\%$ under all scenarios at the cost of a worst EER value under (0) scenario. A solution would be to use several configurations to benefit from the best EER under (0) scenario while minimizing EER under (X.IP) scenario. It is worth noticing that if attackers have the knowledge of the BioCode (and the BioHashing secret), and some modalities, attackers could use such knowledge to invert the BioCode, mainly for pre merging configurations. A possible countermeasure would to to reduce the BioCode size.

4.6 PICRP usage case

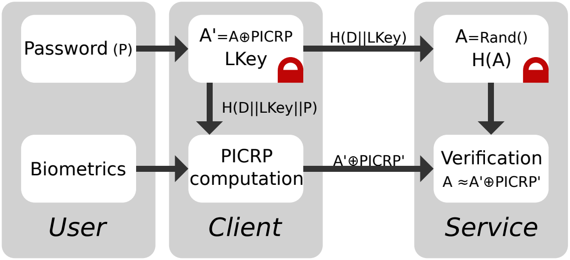


Figure 4.9: Proposed PICRP Authentication Scheme.

In this section, we propose a PICRP authentication scheme illustrated by Figure 4.9. We assume that the client and the service communicate using a secure channel, s.a. TLS, with authentication of the service, e.g. using TLS certificates. For the enrollment, a localkey (LKey) and a random binary vector of length equals to the PICRP (A) is generated by the client. The hash (H, e.g. using SHA2/SHA3) of the service domain name (D), LKey, and the user password (P) is used to compute the PICRP with the user biometrics. A' is computed from the PICRP and A as $A' = PICRP \oplus A$. A' and LKey are stored in the client, encrypted (e.g. with AES256) by the password. A is transmitted to the service, and is encrypted, with its hash (H(A)) using the hash of the Domain and LKey. LKey is detailed in Section 4.7.1.

To authenticate, the client retrieves LKey and A' with the user password, thus enabling the client to compute PICRP' and thus $A \oplus PICRP'$. By sending H(D||LKey) and $A \oplus PICRP'$ to the server, the server is able to retrieve A, verify its integrity, and compare it to $A' \oplus PICRP'$. If the Hamming distance between A and $A' \oplus PICRP'$ is below a given threshold, the user is authenticated.

In this system, an attacker getting into the client cannot gain knowledge as the only stored information are encrypted using a password, without integrity checks. If the attacker knows the password, it would only be able to know LKey and A' that are useless without the knowledge of the biometric data, or A. The system is obviously vulnerable if the attacker gets into the client system while the user discloses its biometric data on the client. In the same way, attacker cannot gain knowledge by getting into the service as information are encrypted using a long random key. If the attacker gets into the service as the user authenticates, it would be able to obtain A that transports no information as randomly generated. It would however be able to authenticate (that we can mitigate by adding a 0-Knowledge proof of LKey to the authentication process).

Thus, to retrieve the biometric data, the attacker has to get into the service, the client, and guess the user password, or to collect directly the biometric data on the client during its usage by the user. As LKey is client-dependent, if the user has many devices, it will either need to get a reference per device, or to compute a new A' for each devices from A and the device-dependant PICRP. The latter solution has the advantage to not disclose the devices used by the user to the server. As the password is only used to encrypt the data on the client-side, it can also be easily changed without impacting the existing biometric references.

4.7 Additional modalities

Other modalities than the one presented above can be used. In a very first initial preliminary study, aimed at demonstrating the feasibility of PICRP, we collected and used other modalities without evaluating performances. Personal information have been simply concatenated without any appropriate pre-processing. The data collection is described in Section 4.7.3.

4.7.1 Collection of personal information

We detail below the modalities collected and used in our initial study.

Browser

To authenticate the browser, a simple key, stored on it is enough. The key, we named *localkey*, is an n-bits value randomly generated upon first usage of the browser. This key is then used to authenticate the browser. For n big enough, the probability of collisions is insignificant, and the exhaustive research, hard. In the frame of the experience, n=64, for higher security needs, the key size may be increase, e.g. with n=512. The key might be stored in the browser localStorage[1], or, ideally, in a WebExtension simple-storage. Nonetheless, it is possible for an attacker to steal the key if it has access to the device, or to the user session. The keys being randomly

[1] HTML5 feature

generated, the theft of one do not compromise the others possessed by the user. The key might be protected, e.g. with encryption, or the fraudulent usage be detected, e.g. with others personal information. However, this will not be introduced in the frame of this study.

Localization

IP addresses are distributed by ranges, from IANA[2] to RIR[3], from RIR to LIR[4], and finally from LIR to users. It is then possible to deduce from it the user network, the administrative (e.g. county) or physical (e.g. GPS position) localization. However, the TOR network, a VPN, or a proxy might be used to masquerade the user IP address. Then, the network and locations deduced from the IP address would be the proxy, VPN, or exit TOR nodes. In the scope of this study, the administrative (country, region, county, town) and physical (latitude and longitude) are extracted through the Google MAP API from an address extracted from the database DB-IP[5]. In a future work, it would be possible to deduce either the user's ISP (Internet Service Provider), or the structural localization among an entity (e.g. company, university, research center, state agencies), thanks to DNS, reverse DNS, WHOIS IP, and WHOIS domain queries. It is also possible to get more information about the IP address thanks to DNSBL[6].

Network data

Data sent to the service by the communication protocols are discriminant and enable, by browser fingerprinting techniques, user identification [Eckersley, 2010, Laperdrix et al., 2016]. In the same way, such data can be used for user authentication by comparing them to enrolment data. As a consequence, this modality cannot be used if the data are randomized for each transaction. However, usurpation is trivial for whom knows this data, e.g. for whom provides a service to the user. Moreover, the usage of normalized data, e.g. by user the TOR browser, increases the collision probability. This modality gives little trust in the user authentication, but enables to detect reception of unusual data. In the scope of this study, the following fields are extracted from the HTTP header:

- *User-Agent*: arbitrary string defined by the browser;
- *Accept, Accept-Language, Accept-Encoding*: formats, languages, and encoding preferences (values $\in [0, 1]$);
- *Referer*: previous pages URL, sometimes randomized, truncated, or removed;
- *Cookie*: cookies sent by the browser;
- *DNT, Connection, Upgrade-Insecure-Requests*: other parameters.

Figure 4.10 shows the distribution of some network and browser data for different users. We can see clearly some differences for each user even if most of them are

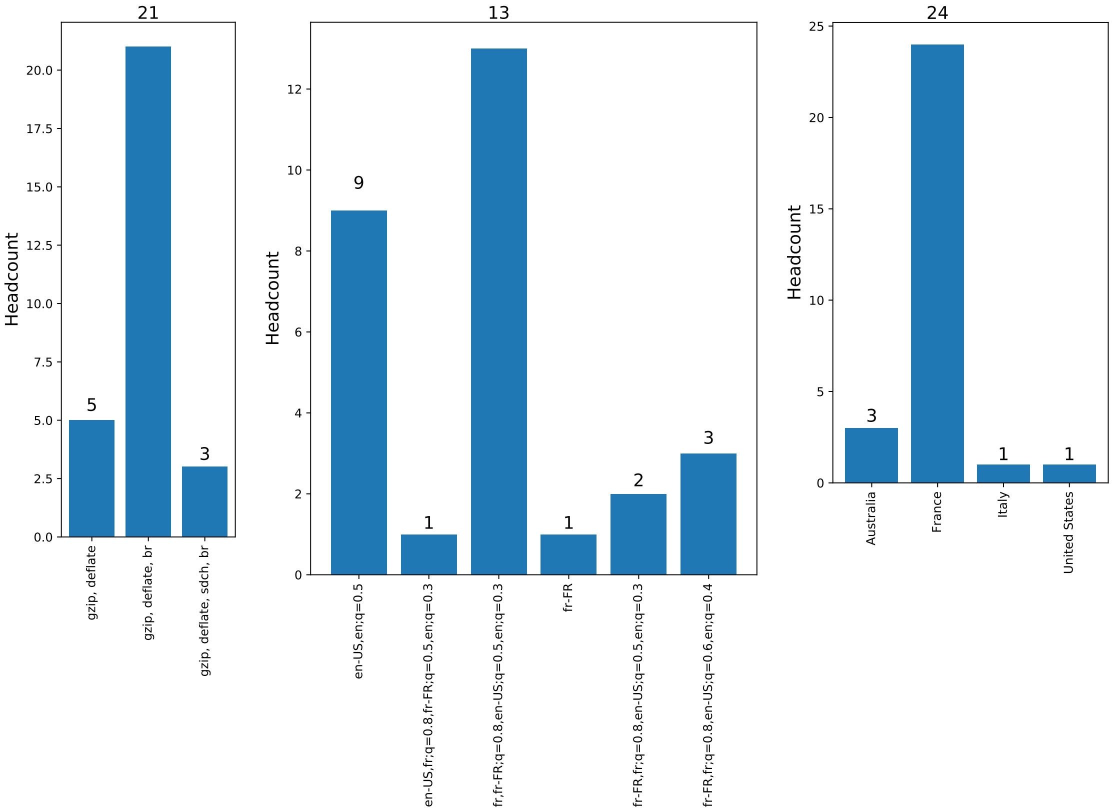
[2] Internet Assigned Numbers Authority

[3] Regional Internet Registry

[4] Local Internet Registry

[5] download.db-ip.com/free/dbip-city-2017-05.csv.gz

[6] DNS Blacklist



(a) Accept-Encoding

(b) Accept-Language

(c) Country

Figure 4.10: Personal information collection: distribution of some collected data (raw data).

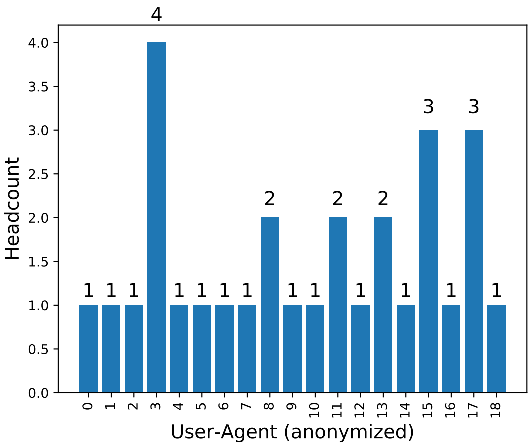


Figure 4.11: Personal information collection: distribution of the user agent (raw data).

french. Figure 4.11 shows the distribution of the User-agent value that are very discriminant among users.

4.7.2 Data pre-processing

To obtain, for each modality, a fixed-size real vector (required for the protection scheme), collected data are converted to real vectors then appended. The distance between two vectors might be influenced by extremes values, they are consequently normalized.

Browser

Localkey (n-bits key) is converted into a n-bits vector. Thus, the 16-bits localkey *"0x0123"*, is converted into *[0,0,0,0, 1,0,0,0, 0,1,0,0, 1,1,0,0]*.

Localisation

An IP address is converted in a vector composed by:

- a vector composed by the IP address bits divided by 2^{32-p-1} with p (bit weight);
- a vector composed by the $128/2^k$ first bits of the locality name's md5 hash with k=1 for "country", k=2 for "region", k=3 for "county", and k=4 for "town";
- a vector composed of 3 angles $\in [-90; +90]$ representing the GPS localization's latitude (lat), and the longitude l (lng1, lng2); lng1 and lng2 are equal to:

  $$sign(\alpha) * ||\alpha| - (|\alpha| > 90) * 180|$$

  with $\alpha = l$ for lng1 and $\alpha = rot90(l) = (l - 90)\%360 - 180$ for lng2. These angles in degree are normalized by the following formula:

  $$angle^* = (angle + 90)/180$$

As for example, the IP adress *"127.0.0.1"* is converted in *[0, 0.5, 0.25, 0.125, 0.0625, 0.03125, 0.015625, 0, 0.0078125, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,* $4.6566 * 10^{-10}$*]*. The following GPS localization *(135, 0)* is converted in *[0.5, 0.75, 0.25]*.

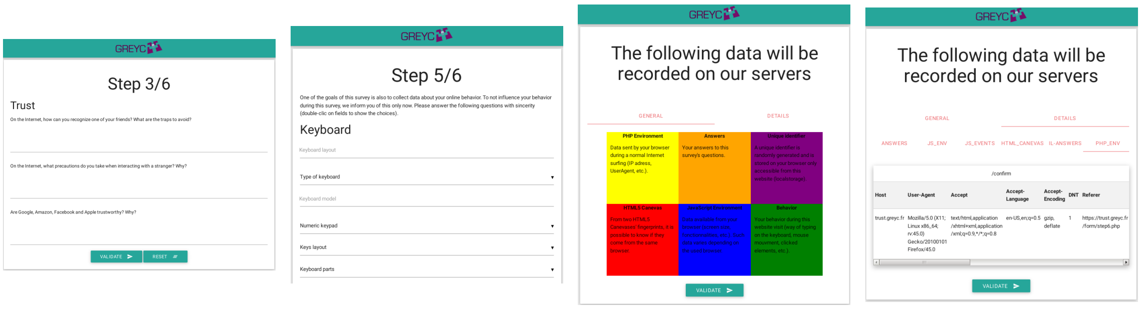
Network data

Referer, User-Agent, Connection and *Cookie* are converted into histograms, vectors giving for each character its headcount. Only the ASCII characters $\in [0x20, 0x7F[$, so 95 characters, are considered. *Accept, Accept-Encoding,* and *Accept-Language* are converted into vectors giving the preference for each format, encoding, and language from a predefined list. An additional value indicates the presence of spaces after comma in the field. *DNT* and *Upgrade-Unsecure-Requests* are converted into a 1-integer vector, equals to 1 if setted, 0 otherwise. The predefined lists are:

- Accept: "text/html", "application/xhtml+xml", "application/xml", "image/webp", "image/jxr";
- Accept-Encoding: "gzip", "deflate", "br", "sdch";
- Accept-Language: "fr", "fr-FR", "en-US", "en".

As for example, the following User-Agent value
"Browser/1.0 (Operating System; rv:1.0) Engine/20170701 Browser/1.0"
is converted by considering only characters in [a-z] by
[1, 0, 0, 0, 5, 0, 2, 0, 2, 0, 0, 0, 1, 3, 2, 1, 0, 6, 3, 2, 0, 1, 2, 0, 1, 0].
The Accept-Language *"fr;q=0.8, fr-FR;q=0.5, en-US"* is described by *[0.8, 0.5, 1, 0, 1]*. The DNT value *"1"* is converted in *[1]*.

4.7.3 Experimental protocol and results



Step 3/6

Trust

On the Internet, how can you recognize one of your friends? What are the traps to avoid?

On the Internet, what precautions do you take when interacting with a stranger? Why?

Are Google, Amazon, Facebook and Apple trustworthy? Why?

Step 5/6

One of the goals of this survey is also to collect data about your online behavior. To not influence your behavior during this survey, we inform you of this only now. Please answer the following questions with sincerity (double-clic on fields to show the choices).

Keyboard

The following data will be recorded on our servers

PHP Environment	Answers	Unique identifier
Data sent by your browser during a normal Internet surfing (IP adress, UserAgent, etc.).	Your answers to this survey's questions.	A unique identifier is randomly generated and is stored on your browser only accessible from this website (localstorage).
HTML5 Canevas	**JavaScript Environment**	**Behavior**
From two HTML5 Canevases' fingerprints, it is possible to know if they come from the same browser.	Data available from your browser (screen size, fonctionnalities, etc.). Such data varies depending on the used browser.	Your behavior during this website visit (way of typing on the keyboard, mouse mouvment, clicked elements, etc.).

The following data will be recorded on our servers

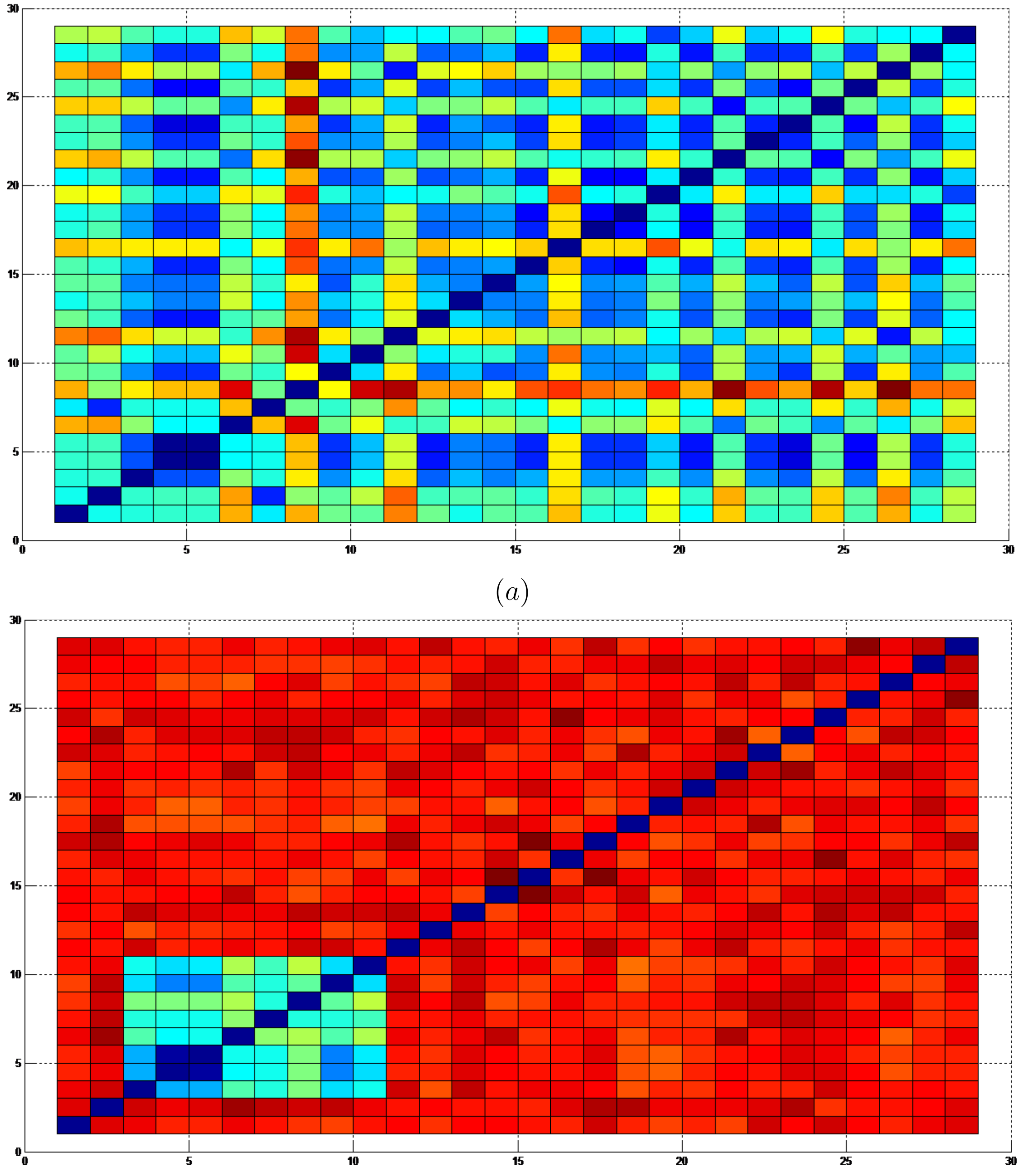
/confirm

Host	User-Agent	Accept	Accept-Language	Accept-Encoding	DNT	Referer
trust.greyc.fr	Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0	text/html,application /xhtml+xml,application /xml;q=0.9,*/*;q=0.8	en-US,en;q=0.5	gzip, deflate	1	https://trust.greyc.fr /form/step6.php

Figure 4.12: Personal information collection questionnaire's screens

Experimental protocol

An acquisition campaign as be organized in march 2017 in the trust.greyc.fr website. The participants have been recruited from the GREYC laboratory and the engineering school ENSICAN, broadcast lists. Thus, collected data come from a unique place, indeed the majority of the participants are localized in Caen, use the same networks (ENSICAN and UNICAEN), and thus have the same IP address. Moreover, the use of GREYC and ENSICAEN devices make the participants configuration, and network data quite similar. With only 22 participants, mostly located in Caen, the sample is not representative, but enables a first experimentation of the personal identity code. During the acquisition, participants are invited to answer to 8 questions on privacy, then to copy an extract of the Universal Declaration of Human Rights (see 4.12). To prevent any influence for the keystroke dynamics, participants are informed of the data collection only from the step 5, where they are invited to give the authorization to use personal information for research purposes. All the collected data are stored in the browser sessionStorage and are submitted only after user validation through the confirmation page, resuming the collected information types, and detailing collected information. Once the data are submitted, a localkey is generated and stored inside the browser localStorage, to recognize the browser upon multiples submissions. The localkey is also printed to users so that they can exercise their right of data access and correction.



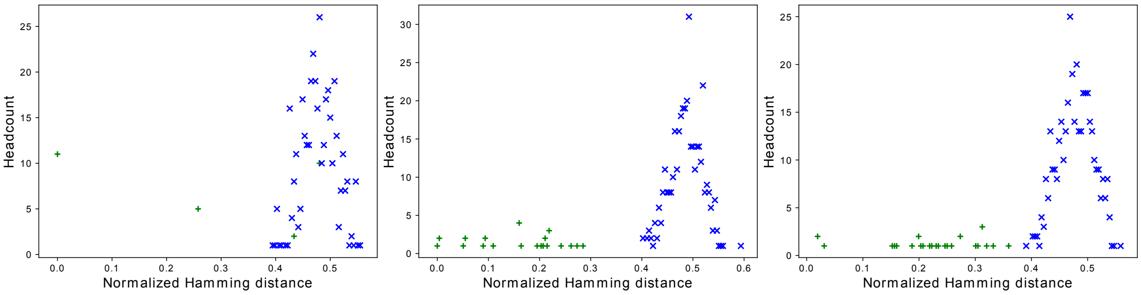
(*a*)

(*b*)

Figure 4.13: Information comparison between the pre-processed data (a) and after protection (b). In coordinates, the compared entries number: blue for an high similarity, red for low.

Experimental results

From the 29 collections, from 22 users (8 have been made by the same user in different contexts), we estimate in which proportion these information enable to compute users similarity. Figure 4.14 presents the distribution of BioCodes comparisons for all users using different collected information and the total. In green, are represented intra-users comparisons between BioCodes and in blue inter-users comparisons. The distribution of BioCodes generated by taking into account only localization (Figure 4.14 (a)) show some errors to discriminate users. Indeed, the same user provided some information at different localizations (sometimes more similar to other users). The BioCode generated using the PHP environment and the total, permits to clearly discriminate users from each others.



(a) Localisation BioCode (b) PHP env. BioCode (c) Total BioCode

Figure 4.14: Distribution of BioCode comparisons for all users. In green, are represented intra-user comparisons and in blue inter-users comparisons.

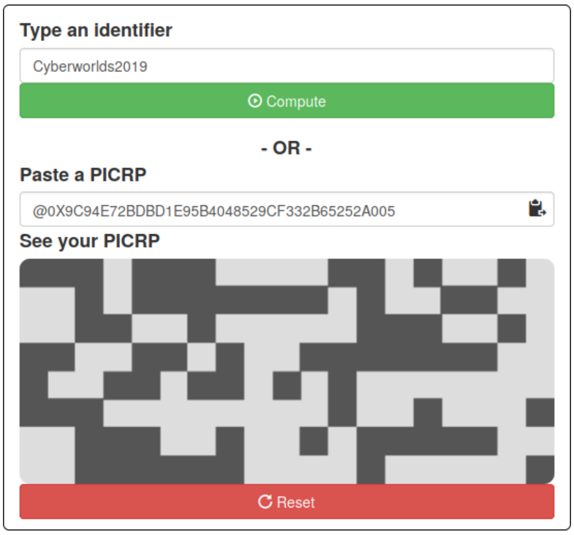
Figure 4.13 shows two distance matrices. The first (a) compare pre-processed data (without any protection) with the cosine distance $(1 - cos(A, B)$, if A and B are two real vectors. In this figure, we can notice two things. The first is that the signatures 4 and 5 are judged very similar. This is in fact the same user in the same context. The only difference is in the keystroke dynamics. Signatures 3 to 10 have been generated by the same user, but in different contexts (s.a. Wifi, browser), the similarity is more contrasted. The second important observation is the relative similarity of the signatures 4 and 5 with others signatures. This can be explained as these signatures have been acquired inside the laboratory with devices with similar configurations and IP address. Figure 4.13 (b) represents the distances between BioCodes (protected signatures) with, for each user, a unique secret key. With the protection and these keys, we highlight the similarity between users. For binary codes linked to signatures 3 to 10, we identify a similarity between then with variations depending on the similarity of personal information. This demonstrates the capacity of the proposed method to produce an exploitable code for personal information similarity computation.

Discussion

This very first experiment only demonstrated the feasibility of a PICRP with the previously mentioned modality. Modalities should be pre-processed and merged as presented in sections 4.4 and 4.5. The very low number of participants in the dataset (22 users), and another attempt (1 user), compared to the time to prepare such online acquisition, lead us to opt for dataset synthetic generation (see Chapter 5) and for chimeric datasets (as in Section 4.3).

4.8 Conclusion

In this chapter, we defined and implemented the concept of PICRP with the use of Keystroke Dynamics, IP, and GPS geo-location. The pre-processing of Keystroke Dynamics permits to significantly increase performances (EER from 40% before pre-processing to 24.2% after). Geo-location has been found to produce great performances (EER $\lesssim 5\%$). Performances obtained after merging of these modalities produce satisfactory performances (EER $< 1\%$). A PICPR authentication scheme has been introduced as a possible use case. Other usage could be found s.a. generation of keys from PICRP in order to sign, encrypt, or hash.



Type an identifier

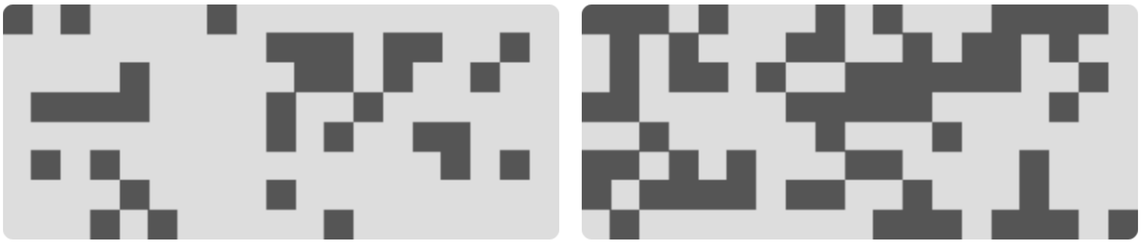
Cyberworlds2019

Paste a PICRP

@0X9C94E72BDBD1E95B4048529CF332B65252A005

See your PICRP

Figure 4.15: PICRP demonstration interface.



(a) Same user, same location.

(b) Different users, different location.

Figure 4.16: Examples of differences (in black) between two PICRP (using the same secret) computed with pre.len.47.25 under scenario (0).

The interface of a PICRP demonstration is shown in Figure 4.15. Figure 4.16 shows differences in PICRP from the same user and from different users.

Future works could focus on improving the pre-processing steps and merging methods. Other modalities could also be integrated to the PICRP, s.a. Browser Fingerprints, mouse, or even soft-biometrics computed from modalities. In this study, users geo-location have been synthetically generated. Further study should be conduct with real-life data. References were merged into one in this study, template-update techniques with user-dependant threshold could also be explored s.a. in [Mhenni et al., 2019].

In short: *Due to the failure of initial collection sessions, we opted for synthetic and chimerics datasets. We proposed PICRP, a soft hash based on BioHashing, in order to compare modalities without revealing them. While such protection introduces a cost in performances, merging modalities provides satisfactory results, even if some are assumed stolen.*

Chapter 5

How to use this data for research while respecting privacy?

This chapter addresses the issue of Keystroke Dynamics modelling. Keystroke Dynamics are time-consuming to collect and, as any biometric modality, are subject to the European GDPR legislation. Thus, it has for consequence the need of Keystroke Dynamics synthetic generation. Moreover, as seen in the previous chapter, a better understanding of Keystroke Dynamics could also improve KDS performances.

Keywords: *Keystroke dynamics; Statistical modelling; Synthetic dataset; Data Analysis.*

Contents

Contributions presented in this chapter

- Modelling of Keystroke Dynamics;
- Generation of Keystroke Dynamics for a given user;
- Usurpation of given user Keystroke Dynamics.

Publications

- Migdal, D. and Rosenberger, C. (2019f). Statistical Modeling of Keystroke Dynamics Samples For the Generation of Synthetic Datasets. *Elsevier Journal on Future Generation Computer Systems, Special Issue on CyberSecurity & Biometrics for a better Cyberworld (Q1 - JCR).*
- Migdal, D. and Rosenberger, C. (2018a). Analysis of Keystroke Dynamics For the Generation of Synthetic Datasets. In *CyberWorlds (B - Core)*, Singapour, Singapore.

5.1 Motivations

Biometrics is an emerging technology more and more present in our daily life. However, building biometric systems requires a large amount of data that may be difficult to collect. Collecting such sensitive data is also very time consuming and constrained, s.a. GDPR legislation in Europe. In the case of keystroke dynamics, most existing databases have less than 200 users. For these reasons, it is crucial for this biometric modality to be able to generate a significant and realistic synthetic dataset of keystroke dynamics samples. We propose in this chapter an original approach for the generation of synthetic keystroke data given samples from known users as a first step towards the generation of synthetic datasets. Experimental results show the capability of the proposed statistical model to generate realistic samples from existing datasets in the literature.

User authentication with keystroke dynamics is generally done in real time (*i.e.*, online) in a real world system. Scientists working on keystroke dynamics do not analyze the performance of their system in an online way (*i.e*, by asking users to authenticate themselves in real time and to impersonate other users). Indeed, they work in an offline context by using samples previously collected by other researchers, and stored in a benchmark dataset. A complete list of available keystroke dynamics datasets has been made in [Monaco, 2018, Giot et al., 2015]. As it can be seen, most of datasets have less than 200 individuals and few samples are available for each user. The collection of such datasets is very time consuming, this is the main reason why there is not more very large datasets like for the face modality [Learned-Miller et al., 2016]. This is a crucial problem for the research in this area.

In this chapter, our objective is to model real KD data in order to be able to generate very large synthetic KD datasets. This approach has been used for the digital fingerprint modality with the SFINGE software [Cappelli et al., 2004] as their collection and distribution are regulated in many countries. We believe the KD model could help the research community to create a new dataset of higher quality than the existing ones. We think this work is important, because it is known that KD studies are not fair as (i) acquisition protocols are different between studies [Giot et al., 2011]; (ii) there is not always a comparative study [Killourhy and Maxion, 2011] when authors propose new algorithms; and (iii) there are not always a valuable statistical evaluation [Killourhy and Maxion, 2011]. Our work contributes to solve these problems. We show in this chapter that is possible to statistically model the KD of users from any existing datasets.

5.2 Background

5.2.1 Keystroke Dynamics Systems

In this chapter, we considered 4 configurations and contexts:

- *Bleha*: Euclidian.raw.*.unique@st.10.10;

- *Hocquet*: Hocquet.raw.*.unique@st.10.10;
- *Monrose*: Euclidian.raw.*.min@st.10.10;
- *BioHashing*: BioHashing.raw.*.min@st.10.10.

Used dataset are presented in Table 2.1 which is repeated below. We used both 23 and 45 entries per users in this chapter. 23 enables to split sets into 5 classes while respecting the Cochran rule, i.e. 80% of the classes having at least 5 elements [Sugden et al., 2000]. 45 enables to split sets into 9 classes of 5 elements, knowing that 46 is the maximal value that do not discard the GREYC W2 dataset.

Table 5.2 and Figure 5.1 give, for each datasets and each Keystroke Dynamics System, the Equal Error Rate and the ROC curve.

Name	**Text**	**# of users (23)**	**# of users (45)**	**Source**
GREYC K	greyc laboratory	102	104	[Giot et al., 2009]
GREYC W1	laboratoire greyc	79	62	[Giot et al., 2012]
GREYC W2	sésame	66	46	[Giot et al., 2012]
CMU	.tie5Roanl	51	51	[Killourhy and Maxion, 2009]

Table 2.1: Description of used Same/Fixed-text datasets.

Distance	**CMU**	**GREYC K**	**GREYC W1**	**GREYC W2**
BioHashing	0.307	0.220	0.201	0.237
Bleha	0.360	0.315	0.303	0.284
Hocquet	0.183	0.146	0.107	0.212
Monrose	0.343	0.281	0.255	0.233

Table 5.2: Equal Error Rate of used datasets with 45 entries per users.

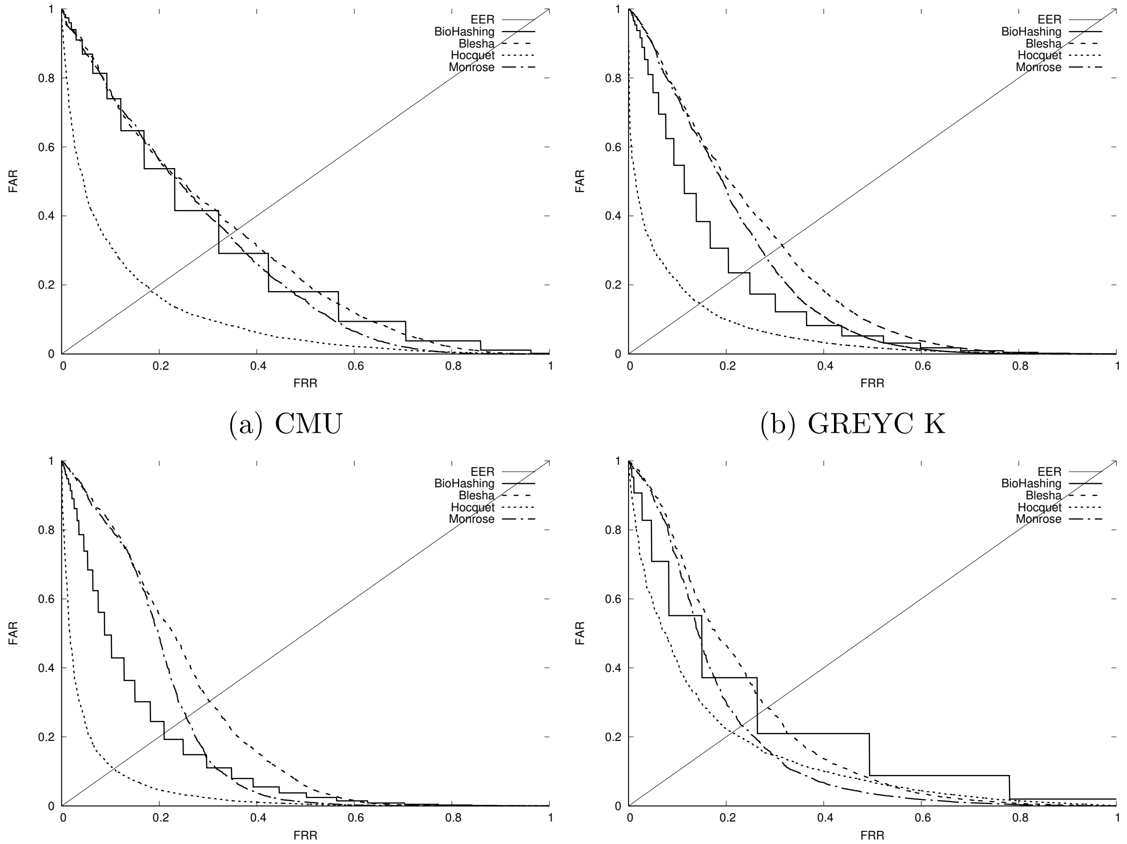
Note that the times in each dataset have been acquired in different ways. In particular, GREYC K used C# programming DateTime which has a resolution of 10.0144ms [1], which explains χ^2's poor results on this dataset. Indeed, some sets of durations have only 8 distinct values which is, when using 45 as the number of elements, less than the number of classes.

5.2.2 Related works

The generation of synthetic keystroke samples has already been discussed in [Stefan et al., 2012, Stefan and Yao, 2010] where authors generated synthetic keystrokes from known users in order to test the robustness of a SVM classifier (used as matching algorithm). Only the uniform and the normal laws have been considered, with the laws parameters directly computed from the mean and standard deviation of the real durations. Authors wanted to generate synthetic keystroke dynamics samples as a naive attack to test the robustness of their presented model.

Keystrokes durations have been analyzed in [Iorliam et al., 2015] where authors aim at assisting the detection of synthetic keystroke samples, by detecting aberrant

[1] `https://manski.net/2014/07/high-resolution-clock-in-csharp/#datetime`



(a) CMU

(b) GREYC K

(c) GREYC W1

(d) GREYC W2

Figure 5.1: ROC curves for the used datasets with 45 entries per users.

duration. Authors found out that some durations follow a Zipf's or/and a Benford's(/power) law on the CMU dataset using the Maximum Likelihood Estimator fitness algorithm to estimate the laws parameters. However, these findings do not enable the synthetic generation of keystroke samples as durations are not separated by users and digraphs, and thus cannot generate a duration for a given user and digraph.

In this chapter, we aim at generating synthetic keystrokes as a way to replace real keystrokes in KD studies. With this approach, we consider 19 laws to find out that the distribution durations follow a gumbel law more than a normal one. We also show that laws parameters computations from the mean and standard deviation give poor results, and the use of a fitness function is required. Moreover, we are interested in the consistency of the duration between them, to generate keystroke samples as real as it can be.

5.3 Analysis of real KD datasets

In this section, we analyze the features from KD samples in existing datasets. We first define the formalism we consider in this study.

5.3.1 Formalism

We define many terms to build the proposed analysis method:

- **Digraph:** $D = [C_0, C_1]$, array of two characters.

- **DigraphTime:** $DT_D = [d_0, d_1, d_2, d_3, d_4, d_5]$, as shown in Figure 5.3, is an array of 6 durations from 4 times corresponding to the pressure (P) and release (R) times of each character of a Digraph D. A DigraphTime DT_D is defined as partially consistent if the following equations are verified, consistent if the following equations and inequalities are verified, and inconsistent otherwise:
    - $d_0 = d_2 - d_4$;
    - $d_0 = d_1 - d_3$;
    - $d_1 = d_2 - d_5$;
    - $d_3 = d_4 - d_5$;
    - $d_0 \geq 0$
    - $d_1 \geq 0$
    - $d_5 \geq 0$

- **Text:** $T_n = \{D_i\}_{i \in [\![0,n[\![}$, an array of n Digraphs D_i. A text T_n is said consistent if $\forall i \in]\!]0, n[\![, D_{i-1}[1] = D_i[0]$.

- **Keystroke dynamics:** $K = [\{DT_i\}_{i \in [\![0,n[\![}, T_n]$, an array of n DigraphTime DT_i associated to the Digraph $T_n[i]$. Keystroke is said consistent (or partially consistent) if T_n, and all DT_i are consistent (or partially consistent), and if $\forall i \in]\!]0, n[\![, DT_{i-1}[5] = DT_i[0]$.

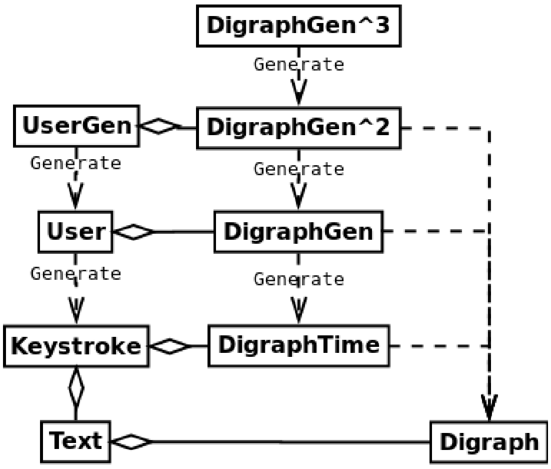


Figure 5.2: KD Generative model

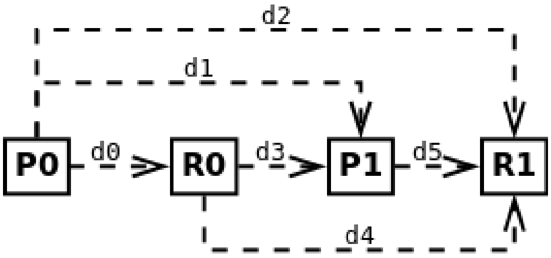


Figure 5.3: DigraphTime

We propose in this chapter a generative keystroke dynamics model. We explain its different components (see also Figures 5.2 and 5.3):

- **DigraphGen:** $DG_D() = DT$, generates a DigraphTime for a given Digraph.

- **User:** $U(T_n) = K$, generates a keystroke dynamics sample from a given Text. A User is composed of a set of DigraphGen.

- **DigraphGen^2:** $DG^2_D() = DG_D$, generates a DigraphGen for a given Digraph.

- **UserGen:** $UG() = U$, generates a User. A UserGen is composed of a set of DigraphGen^2.

- **DigraphGen^3:** $DG^3(D) = DG^2_D$, generates a DigraphGen^2 for a given digraph.

5.3.2 Statistical modelling

As previously seen, generating a keystroke dynamics template from a given text T_n consists in generating an array of DigraphTime, i.e. generating $6 * n$ durations. To be able to generate a keystroke dynamics sample similar to that one user could type, these $6 * n$ durations have to be transformed into a set of assumed independent variables which laws and parameters can then be estimated for a user. We need then to randomly generate durations associated to a given user. In the scope of this chapter, only the linear (in)dependency of variables is considered.

Variables (in)dependency

Linearly correlated variables can be transformed into a set of non-linearly correlated variables, through PCA (Principal component analysis), first introduced by Pearson in 1901 [KPFRS, 1901]. However, we show that durations are not strongly correlated between them, and thus, in the scope of this chapter, we assume them to be independent. Even if the usage of PCA is irrelevant in such a case, its first step enables the computation of the inter-correlations of two variables by the computation of a correlation matrix. In a correlation matrix $C = \{C_{i,j}\}_{\{i,j\} \in [\![0,n[\![^2}$, $C_{i,j}$ is the linear correlation between the variables i and j. A correlation matrix $C = \{C_{i,j}\}_{\{i,j\} \in [\![0,n[\![^2}$, with $C_{i,j}$ the linear correlation between the variables i and j, is computed as follows:

1. Given a matrix $M = \{M_k\}_{k \in [\![0,K[\![}$ of K entries $M_k = \{M_{k,i}\}_{i \in [\![0,n[\![}$, with $M_{k,i}$ the realization of the variable i for the entry k.

2. $\bar{M} = \frac{M_{k,i} - \mu_i}{\sigma_i}, i \in [\![0, n[\![, k \in [\![0, K[\![$ where μ_i is the mean of $\{M_{k,i}\}_{k \in [\![0,K[\![}$, and σ_i, its standard deviation.

3. $C = 1/K * \bar{M}^T * \bar{M}$

To qualify the presence of specific correlations between two variables i, j inside m subsets of entries, m correlations matrix $C^l, l \in [\![0, m[\![$ are computed from such subsets. Each element $C_{i,j}$ of the final correlation matrix C is then computed as the mean of each $C^l_{i,j}$: $C_{i,j} = \frac{1}{m} \Sigma_{l=0}^{m-1} C^l_{i,j}$. If each subset corresponds to, e.g. a User, M will be said, in this chapter, "splitted by User", and C will qualify the presence of

User-specifics correlations across all Users.

To identify the same correlations between two sets of variables $\{i_x\}_{x\in[\![0,m[\![}$, $\{j_x\}_{x\in[\![0,m[\![}$, of length m, entries are splitted in m sub-entries $M'_{m*k+x} = \{M_{k,o_x}\}_{o\in\{i,j\}}$. The correlation matrix C is then computed from M'. If each x corresponds to, e.g. a Digraph, M will be said, in this chapter, "merged by Digraph", and C will qualify the presence of non-Digraphs-specifics correlations across all Digraphs.

Laws followed by Variables

Once the variables are assumed independent, or transformed in such a way, laws followed by each variable are searched through the following process:

1. Given the realizations of a variable X, and a law law_p with unknown parameters p;
2. Estimate $\hat{p}$ from the median, mean, min, max, or/and standard deviation of X;
3. Estimate p through a fitness algorithm using $\hat{p}$ as a starting point.

In the scope of this chapter, we seek to maximize $1 - \chi^2(X, law, p)$. The χ^2 test qualifies the capacity of a set of observed values to match a set of expected values. The χ^2 test returns $\chi^2(X, law, p) = 1 - \alpha$, in which α is the p-value, i.e. the probability to obtain the same $1 - \alpha$ score if X follows law_p. If the p-value is below an arbitrary threshold (s.a. 0.05), the hypothesis "X follows law_p" can then be rejected.

However, in the scope of this chapter, our goal is not to reject hypothesis, but to select laws that best represent X. The $\chi^2(X, law, p)$ score can then be seen as a score of distance between observed values of X, and the expected values. For the same reason, the number of estimated parameters is not subtracted to the freedom, in order to have comparable values across all laws.

We compute $\chi^2(X, law, p)$ as follows:

1. Let $Card(X)$ be the cardinal of X;
2. Let $a\%b$ be the rest of the division of a by b;
3. $\mathbb{R}$ is divided in $n = \lceil Card(X)/5 \rceil$ subspaces $E_i, i \in [\![0, n[\![$, each expected to contain 5 elements of X. E_{n-1} is expected to contain $Card(X)\%5$ elements of X if $5 \nmid Card(X)$;
4. Let $X_i = X \cap E_i$;
5. Let $Card(E_i) = 5$, and $Card(E_{n-1}) = Card(X)\%5$ if $5 \nmid Card(X)$;
6. Let $Sum = \Sigma_{i=0}^{n-1}(Card(E_i) - Card(X_i))^2/Card(E_i)$.
7. Let cdf_f be the cumulative distribution function of the law χ^2 of freedom f;

8. $\chi^2(X, law, p) = cdf_{n-1}(Sum)$.

To qualify the capacity of n subsets of X, $X_i, i \in \llbracket 0, n \llbracket$, to follow a same law law, but each with different parameters p_i, $s = 1 - \chi^2(X, law)$ is computed as the mean of the χ^2 test applied on each X_i: $s = 1 - \frac{1}{n}\Sigma_{i=0}^{n-1}\chi^2(X_i, law, p_i)$. The higher s is, the more the law law is assumed to fit the observed values. In the scope of this chapter, 5 fitness algorithms are used:

- Maximum Likelihood Estimation (R_mle);
- Quantile Matching Estimation (R_qme);
- Maximum Goodness-of-fit Estimation (R_mge);
- The best estimation between R_mle, R_qme, and R_mge (R_max);
- $\hat{p}$ (raw);

The R_mle, R_qme, and R_mge fitness algorithms are executed through R's `fitdist` function[2]. {1/3,2/3} is used as probs parameter for R_qme. If the fitness algorithm fails to estimate p, p is set to $\hat{p}$, and $1 - \chi^2(X, law, p)$ is assumed to be 0.

In this chapter, a set of 19 laws have been tested with the raw estimator, with and without exclusion of aberrant values (here, values that differ from $\pm 3\sigma$ from the median value of X):

- arcsine
- beta
- betaprime
- chi
- chisquare
- raised cosine
- erlang
- exponential
- f
- gamma
- gumbel
- laplace
- logistic
- lognormal
- uniform
- normal
- rayleigh
- student's t
- triangular

From these tested laws, the best 3 are selected, i.e. the 3 laws that maximize $s = 1 - \frac{1}{n}\Sigma_{i=0}^{n-1}\chi^2(X_i, law, p_i)$, and are tested again with the other fitness algorithms. All laws are not directly tested with all fitness algorithms to gain time on the execution, but also due to the fact that all laws (s.a. raised cosine) are not defined in R.

5.3.3 Experimental observations

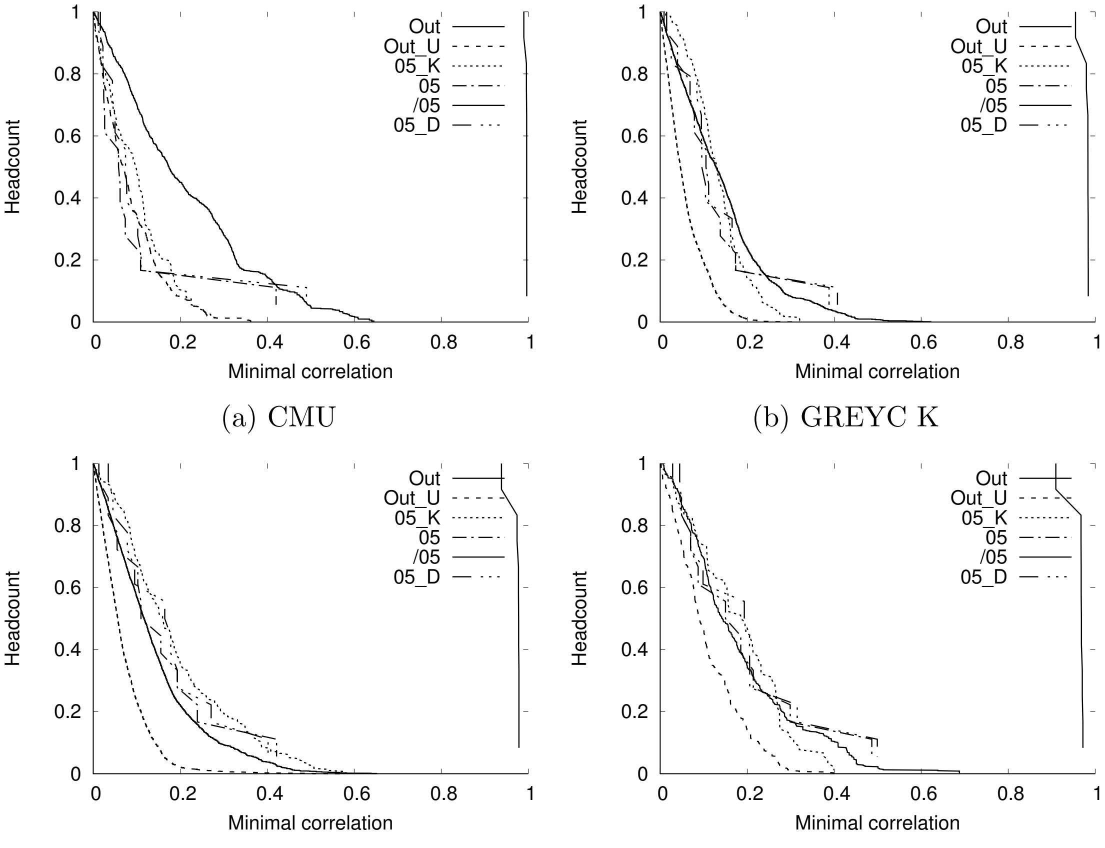
In this section, we first analyze the statistics of real keystroke dynamics from the datasets presented in section 2.3.

Durations correlations

We analyze as a starting point the correlation between durations in a keystroke dynamics sample.

First, diagonals of correlation matrix are discarded. Correlations between two durations $DT_{D_i}[5]$, and $DT_{D_j}[0]$ are discarded if $j = i+1$, as they are in fact the same

[2] `https://www.rdocumentation.org/packages/fitdistrplus/versions/1.0-11/topics/fitdist`



(a) CMU

(b) GREYC K

(c) GREYC W1

(d) GREYC W2

Figure 5.4: Number of correlations greater to a minimal value, between durations from different Digraphs (Out, Out_U), durations d_0 and d_5, with durations of the same Digraph (05_K, 05_D, 05), and between durations d_1 to d_4 inside a same Digraph (/05).

duration. Digraph are considered equal if their positions in the keystroke sample are equals.

As shown in Figure 5.4, no strong stable correlation has been found between durations from different Digraph, (Out: dataset, Out_U: dataset splitted by User). DigraphTime will be thus assumed as independent. Also, no strong stable correlation implying durations d_0 and d_5 of a same DigraphTime has been found (05_K: dataset splitted by User, 05_D: dataset merged and splitted by Digraph, 05: dataset merged by Digraph).

Stable correlations have been detected between durations d_1, d_2, d_3, d_4 of a same DigraphTime ($\overline{05}$: dataset merged by Digraph). It is easy to understand such a result as these durations can be written as $d_x = d_3 + k_x * d_0 + l_x * d_5$ with $l_x \in \{0, 1\}$, $k_x \in \{0, 1\}$, and $\sigma(d_3) \approx 3 * \sigma(d_0 + d_5)$ (see Table 5.3). In the scope of this chapter, DigraphTime is assumed to be computable from 3 independent durations.

Dataset	$\sigma(d_3)/(\sigma(d_0 + d_5))$	$\sigma(d_3)/(\sigma(d_0) + \sigma(d_5))$
GREYC K	3.86	2.98
GREYC W1	3.14	2.41
GREYC W2	2.51	2.03
CMU	6.24	4.96

Table 5.3: Standard deviation of d_0 durations, compared to the standard deviation of d_3 and d_5 durations.

Datasets	Rank	d_0	χ^2	d_1	χ^2	d_2	χ^2	d_3	χ^2	d_4	χ^2	d_5	χ^2
CMU	1	normal (3σ)	0.550	gumbel (3σ)	0.262	gumbel (3σ)	0.266	gumbel (3σ)	0.266	gumbel (3σ)	0.285	normal (3σ)	0.546
	2	logistic (3σ)	0.546	logistic (3σ)	0.194	logistic (3σ)	0.200	logistic (3σ)	0.193	logistic (3σ)	0.210	logistic (3σ)	0.546
	3	cosine (3σ)	0.524	normal (3σ)	0.172	normal (3σ)	0.189	normal (3σ)	0.166	normal (3σ)	0.197	cosine (3σ)	0.523
	4	logistic	0.505	laplace (3σ)	0.159	laplace (3σ)	0.167	laplace (3σ)	0.156	laplace (3σ)	0.178	logistic	0.507
	5	normal	0.491	cosine (3σ)	0.149	cosine (3σ)	0.163	cosine (3σ)	0.142	cosine (3σ)	0.162	normal	0.484
	6	cosine	0.438	gumbel	0.127	gumbel	0.133	gumbel	0.123	gumbel	0.142	cosine	0.433
	7	gumbel (3σ)	0.410	logistic	0.081	logistic	0.084	rayleigh (3σ)	0.078	logistic	0.085	gumbel (3σ)	0.403
	8	gumbel	0.388	normal	0.068	laplace	0.076	logistic	0.077	laplace	0.077	laplace	0.391
	9	laplace (3σ)	0.384	laplace	0.063	normal	0.072	laplace	0.067	normal	0.067	laplace (3σ)	0.390
	10	laplace	0.380	cosine	0.057	cosine	0.063	normal	0.055	cosine	0.051	gumbel	0.377
GREYC K	1	normal (3σ)	0.009	gumbel (3σ)	0.149	gumbel (3σ)	0.175	gumbel (3σ)	0.143	gumbel (3σ)	0.157	cosine (3σ)	0.008
	2	cosine (3σ)	0.008	normal (3σ)	0.143	normal (3σ)	0.173	normal (3σ)	0.140	normal (3σ)	0.154	normal (3σ)	0.008
	3	normal	0.008	cosine (3σ)	0.135	logistic (3σ)	0.162	logistic (3σ)	0.137	logistic (3σ)	0.147	normal	0.008
	4	cosine	0.007	logistic (3σ)	0.135	cosine (3σ)	0.153	cosine (3σ)	0.129	cosine (3σ)	0.142	cosine	0.007
	5	logistic (3σ)	0.006	gumbel	0.099	gumbel	0.109	gumbel	0.092	gumbel	0.098	logistic (3σ)	0.005
	6	logistic	0.005	laplace (3σ)	0.088	laplace (3σ)	0.103	laplace (3σ)	0.089	laplace (3σ)	0.096	logistic	0.005
	7	uniform (3σ)	0.004	normal	0.076	normal	0.090	logistic	0.079	logistic	0.084	uniform (3σ)	0.004
	8	gumbel	0.004	logistic	0.074	logistic	0.086	normal	0.072	normal	0.076	gumbel (3σ)	0.004
	9	gumbel (3σ)	0.004	cosine	0.068	cosine	0.075	cosine	0.067	cosine	0.069	gumbel	0.004
	10	uniform	0.004	laplace	0.055	laplace	0.065	rayleigh (3σ)	0.055	laplace	0.057	uniform	0.004
GREYC W1	1	cosine (3σ)	0.149	logistic (3σ)	0.194	logistic (3σ)	0.231	logistic (3σ)	0.164	normal (3σ)	0.188	cosine (3σ)	0.147
	2	normal (3σ)	0.145	normal (3σ)	0.192	normal (3σ)	0.227	gumbel (3σ)	0.159	logistic (3σ)	0.186	normal (3σ)	0.145
	3	logistic (3σ)	0.136	gumbel (3σ)	0.192	gumbel (3σ)	0.220	normal (3σ)	0.153	gumbel (3σ)	0.181	logistic (3σ)	0.135
	4	logistic	0.124	cosine (3σ)	0.171	cosine (3σ)	0.207	cosine (3σ)	0.132	cosine (3σ)	0.164	logistic	0.124
	5	normal	0.119	laplace (3σ)	0.140	laplace (3σ)	0.166	laplace (3σ)	0.123	laplace (3σ)	0.133	normal	0.119
	6	cosine	0.116	logistic	0.114	gumbel	0.140	gumbel	0.089	logistic	0.108	cosine	0.115
	7	laplace	0.095	gumbel	0.110	logistic	0.137	logistic	0.084	gumbel	0.107	laplace	0.096
	8	laplace (3σ)	0.095	normal	0.103	normal	0.131	normal	0.076	normal	0.093	laplace (3σ)	0.095
	9	gumbel (3σ)	0.092	laplace	0.091	cosine	0.113	laplace	0.072	laplace	0.085	gumbel (3σ)	0.092
	10	gumbel	0.091	cosine	0.086	laplace	0.104	cosine	0.063	cosine	0.074	gumbel	0.091
GREYC W2	1	normal (3σ)	0.208	gumbel (3σ)	0.235	gumbel (3σ)	0.264	gumbel (3σ)	0.198	logistic (3σ)	0.226	normal (3σ)	0.210
	2	cosine (3σ)	0.191	logistic (3σ)	0.217	logistic (3σ)	0.250	logistic (3σ)	0.188	gumbel (3σ)	0.219	logistic (3σ)	0.190
	3	logistic (3σ)	0.190	normal (3σ)	0.193	normal (3σ)	0.224	normal (3σ)	0.188	normal (3σ)	0.212	cosine (3σ)	0.187
	4	logistic	0.161	cosine (3σ)	0.179	cosine (3σ)	0.214	cosine (3σ)	0.155	cosine (3σ)	0.173	logistic	0.165
	5	gumbel (3σ)	0.158	laplace (3σ)	0.169	laplace (3σ)	0.179	laplace (3σ)	0.128	laplace (3σ)	0.156	gumbel (3σ)	0.150
	6	normal	0.148	gumbel	0.138	gumbel	0.146	gumbel	0.106	gumbel	0.136	normal	0.148
	7	cosine	0.135	logistic	0.114	logistic	0.124	normal	0.096	logistic	0.121	cosine	0.133
	8	gumbel	0.132	normal	0.093	normal	0.117	logistic	0.094	normal	0.107	gumbel	0.131
	9	laplace (3σ)	0.131	laplace	0.092	laplace	0.115	laplace	0.084	laplace	0.105	laplace (3σ)	0.123
	10	laplace	0.118	cosine	0.082	cosine	0.104	rayleigh (3σ)	0.083	cosine	0.092	laplace	0.115

Table 5.4: Top 10 results of χ^2 tests with 19 laws, using raw estimator, with (3σ) and without exclusion of aberrant values. $\chi^2 = \mathbf{1}$ -$\chi^2(X, law)$

Durations laws

For the 6 DigraphTime durations $DT_D[i]$, $i \in [\![0, 6[\![$, the 10 best laws that maximize $1 - \chi^2(DT_D[i], law)$, with parameters depending on the Digraph and User, are presented in Table 5.4. DigraphTime durations will then be assumed to best follow either a gumbel, a normal, or a logistic, which parameters depend on the User and Digraph.

Datasets	Rank	d_0	χ^2	d_1	χ^2	d_2	χ^2	d_3	χ^2	d_4	χ^2	d_5	χ^2
CMU	1	normal (3σ)	0.685	gumbel (3σ)	0.530	gumbel (3σ)	0.533	gumbel (3σ)	0.533	gumbel (3σ)	0.544	normal (3σ)	0.680
	2	logistic (3σ)	0.677	gumbel	0.503	gumbel	0.511	gumbel	0.503	gumbel	0.515	logistic (3σ)	0.673
	3	logistic	0.668	logistic (3σ)	0.360	logistic (3σ)	0.352	logistic (3σ)	0.338	logistic (3σ)	0.363	logistic	0.665
	4	normal	0.668	normal (3σ)	0.348	normal (3σ)	0.347	normal (3σ)	0.327	normal (3σ)	0.360	normal	0.663
	5	gumbel (3σ)	0.604	normal	0.322	logistic	0.324	logistic	0.309	logistic	0.334	gumbel (3σ)	0.591
	6	gumbel	0.596	logistic	0.321	normal	0.324	normal	0.301	normal	0.324	gumbel	0.577
GREYC K	1	normal	0.011	gumbel (3σ)	0.305	gumbel (3σ)	0.357	gumbel (3σ)	0.287	gumbel (3σ)	0.310	normal	0.010
	2	normal (3σ)	0.011	gumbel	0.296	gumbel	0.350	gumbel	0.282	gumbel	0.302	normal (3σ)	0.010
	3	logistic	0.009	normal (3σ)	0.235	normal (3σ)	0.279	normal (3σ)	0.223	normal (3σ)	0.247	logistic	0.009
	4	logistic (3σ)	0.009	logistic (3σ)	0.229	logistic (3σ)	0.267	logistic (3σ)	0.214	logistic (3σ)	0.238	logistic (3σ)	0.008
	5	gumbel (3σ)	0.009	normal	0.209	normal	0.250	normal	0.201	normal	0.222	gumbel (3σ)	0.008
	6	gumbel	0.008	logistic	0.209	logistic	0.248	logistic	0.194	logistic	0.221	gumbel	0.008
GREYC W1	1	normal (3σ)	0.197	gumbel (3σ)	0.347	gumbel (3σ)	0.408	gumbel (3σ)	0.299	gumbel (3σ)	0.350	normal (3σ)	0.198
	2	normal	0.196	gumbel	0.342	gumbel	0.403	gumbel	0.292	gumbel	0.343	normal	0.196
	3	logistic	0.191	normal (3σ)	0.309	normal (3σ)	0.359	logistic (3σ)	0.257	normal (3σ)	0.297	logistic	0.193
	4	logistic (3σ)	0.186	logistic (3σ)	0.301	logistic (3σ)	0.357	normal (3σ)	0.255	logistic (3σ)	0.294	logistic (3σ)	0.188
	5	gumbel (3σ)	0.161	logistic	0.283	logistic	0.338	normal	0.240	logistic	0.281	gumbel (3σ)	0.163
	6	gumbel	0.155	normal	0.283	normal	0.333	logistic	0.237	normal	0.274	gumbel	0.158
GREYC W2	1	normal (3σ)	0.280	gumbel (3σ)	0.441	gumbel (3σ)	0.491	gumbel (3σ)	0.358	gumbel (3σ)	0.417	normal (3σ)	0.278
	2	logistic	0.267	gumbel	0.419	gumbel	0.462	gumbel	0.338	gumbel	0.408	logistic	0.264
	3	logistic (3σ)	0.265	logistic (3σ)	0.340	normal (3σ)	0.383	normal (3σ)	0.294	logistic (3σ)	0.350	logistic (3σ)	0.260
	4	normal	0.260	normal (3σ)	0.331	logistic (3σ)	0.370	logistic (3σ)	0.284	normal (3σ)	0.345	normal	0.253
	5	gumbel	0.245	logistic	0.319	normal	0.367	normal	0.267	logistic	0.311	gumbel	0.240
	6	gumbel (3σ)	0.239	normal	0.318	logistic	0.362	logistic	0.265	normal	0.307	gumbel (3σ)	0.237

Table 5.5: Top 6 results of χ^2 tests with 3 laws, using R_max estimator, with (3σ) and without exclusion of aberrant values. $\chi^2 = \mathbf{1}$ -$\chi^2(X, law)$

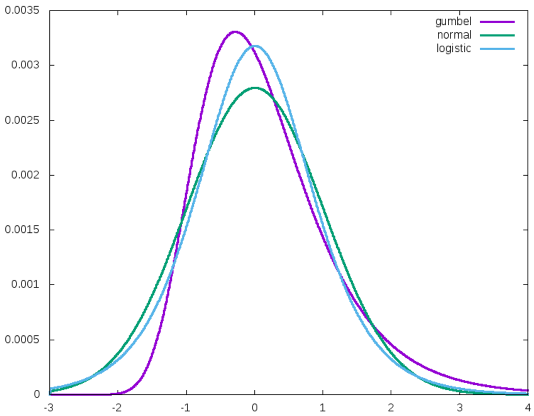


Figure 5.5: Density function (pdf) of several laws (with median=0, standard deviation=1).

These findings are confirmed in Table 5.5. The gumbel law seems to best fit d_1 to d_4 durations followed by either the normal or the logistic law. However, for d_0 and d_5 durations, the normal law seems to best fit them, followed by the logistic law and the gumbel law. The exclusion of aberrant values seems to increase the fitness of the law.

As shown in Figure 5.5, these three laws are quite similar. Contrary to the two others, the gumbel law is asymmetric and possesses a trail that match users' hesitations when typing.

We define the coverage as the headcount of sets for which $\mathbf{1}$ -$\chi^2(X, law) > 0.01$.

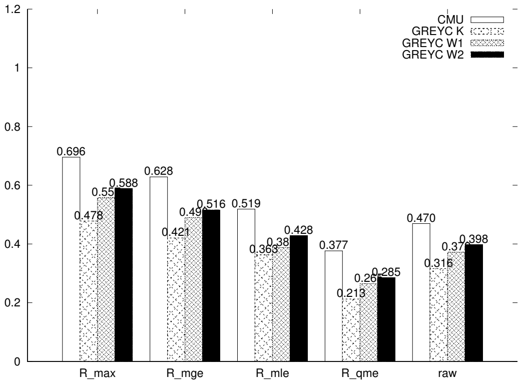


Figure 5.6: **1** -$\chi^2(law)$ for gumbel (3σ) with 23 elements per sets

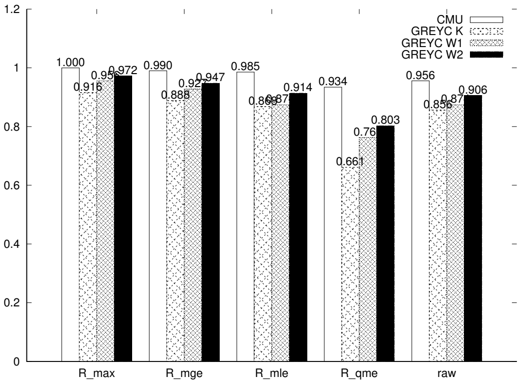


Figure 5.8: Coverage for gumbel (3σ) with 23 elements per sets

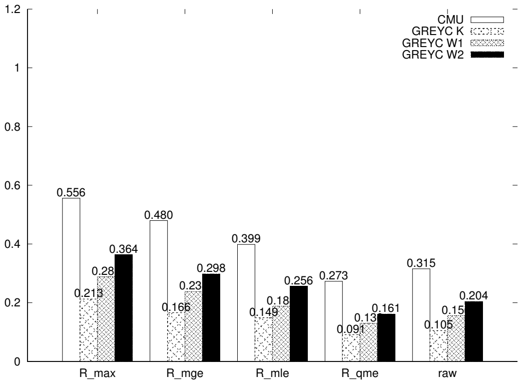


Figure 5.7: **1** -$\chi^2(law)$ for gumbel (3σ) with 45 elements per sets

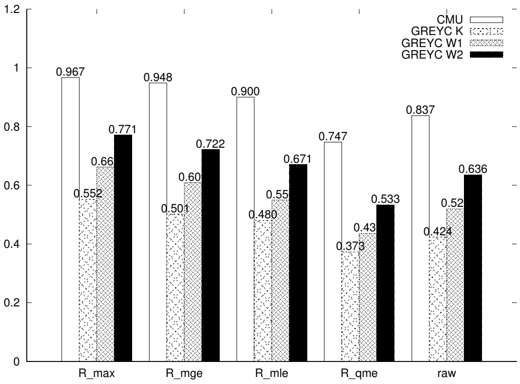


Figure 5.9: Coverage for gumbel (3σ) with 45 elements per sets

As shown in Figures 5.6 to 5.9, sets of 23 elements give better χ^2 scores than with 45 elements, that can be explained by the fact that sets of 45 elements have more classes, and thus the χ^2 test is more strict. GREYC K gives poor results, that can be explained to its d_0 and d_5 durations and the time precision of near 10ms. On the contrary, CMU gives the best results, followed by GREYC W2 and GREYC W1. As expected, the R_max fitness algorithm performs better than other fitness algorithms. Although, R_mle and R_qme perform poorly, they still give a significant increase to the R_max fitness algorithm. Surprisingly, raw fitness algorithm outperforms R_qme.

In order to reduce the number of possible combinations, each duration will be generated with by two laws X, Y (X being used for d_0 and d_5, and Y for d_1 to d_4), but with different parameters. The configuration will be noted X_Y. If X and Y are the same law, the configuration will be noted X.

In our study, we used 7 configurations obtained by combining the normal, and logistic law as X, and the gumbel, normal, and logistic law as Y, and adding the configuration gumbel_gumbel (i.e. gumbel). If the parameters of the laws have been estimated with exclusion of aberrant values, ”-3s” is appended to the configuration name.

We can see clearly in Tables 5.4 and 5.5 that the estimated laws and parameters for all DigraphTime durations are quite similar for the datasets we used in this study. Thanks to these statistical observations, we propose a generative model of keystroke dynamics data in the next section.

5.4 Keystroke dynamics generative model

5.4.1 Principles

As seen in the previous section, DigraphTime durations follow either a gumbel, a normal, or a logistic law which parameters can be estimated for each known User and Digraph. For a given User and Digraph, a DigraphGen can be then implemented as a set of 6 random engines generating the 6 DigraphTime durations with the chosen law and estimated parameters.

The full generative algorithm is thus the following:

- Select two laws, one for d_0 and d_5, one for d_1 to d_4;
- Estimate the parameters of the durations for each DigraphTime;
- Generate a new Keystroke by randomly generating durations from the chosen laws and estimated parameters;
- Apply a consistency strategy on the generated Keystroke.

We propose 10 consistency strategies, 1 for inconsistent DigraphTime, in which all durations are randomly generated (u), and 10 for partially-consistent DigraphTime, in which 3 durations are computed from the 3 others. The durations to compute can be chosen among the 8 following lists, and be used for all Digraph and User, or be randomly chosen (null) for each new DigraphTime to generate:

- 0: $d_3d_4d_5$
- 1: $d_2d_3d_5$
- 2: $d_2d_3d_4$
- 3: $d_1d_4d_5$
- 4: $d_1d_3d_4$
- 5: $d_1d_2d_5$
- 6: $d_1d_2d_4$
- 7: $d_2d_1d_3$

We also propose an 11th consistency strategy that perform the mean of the 8 strategies from the previous list (m). For each consistency strategy, we propose a fully-consistent version which first applies the consistency strategy, then set to 0 negative d_0, $d5$, and d_1 durations, before recomputing d_2, d_3, and d_4 from the 3 previous duration. Such strategies are suffixed by 'c'.

Once the DigraphGen created for a given User, the keystroke dynamics of a given Text T_n is generated through the following process:

1. $K[1] = T_n$

2. $\forall i \in [\![0, n[\![, K[0][i] = DT_{T_n[i]} = DG_{T_n[i]}()$.

Before the consistency strategy application, and if Keystroke is expected to be consistent (or partially consistent), the DigraphTime first duration $K[0][i][0]$ is settled, if exists (i.e. if $i > 0$), to the last duration of the previous DigraphTime $K[0][i-1][5]$.

3. If fully-consistent strategy, d_2 to d_4 recomputed after setting negative d_0, d_1, and d_5 to 0.

5.4.2 Synthetic dataset generation: protocol

20 synthetic datasets are generated for each real KD datasest, and each possible configuration, i.e. each law configuration L and each consistency strategy CS. The configuration is labelled L.CS. These synthetic datasets are generated so as to contain the same number of users and entries per user than the real dataset from which it is generated (as seen in previous section).

For each synthetic dataset, and each distance function DistFct (matching algorithm), 3 sub-datasets are computed:

- *DataSU:* to qualify the capacity of synthetic Keystroke dynamics to be indistinguishable from real Keystroke dynamics;
- *DataU:* to qualify the KDS performance with real Keystroke dynamics data;
- *DataS:* to qualify, in comparison with DataU, the capacity of synthetic datasets to match the KDS performance that would be expected with real Keystroke dynamics data.

These datasets are composed of legitimate and impostor scores, computed with the DistFct distance function. Legitimate scores are obtained by comparing the reference template with samples from the same user. The 10 first entries of each User are used as templates, and the other entries as samples. Impostors scores are obtained by comparing the reference template of users with samples from other users. DataU is computed from the real dataset, and DataS, from the synthetic one. In DataSU, legitimate scores are legitimate scores of DataU, and impostors scores are the distance, for each User, between real user templates, and its synthetic samples.

We consider the False Acceptance Rate (FAR) describing the ratio of accepted impostor data, the False Rejection Rate (FRR) describing the ratio of falsely rejected legitimate users. The Equal Error Rate (EER) corresponds to configuration of the biometric system when FAR equals FRR. Computed indicators across the 20 synthetic datasets are aggregated by generating the following values:

- mean: the mean of the indicators;
- error: the difference between the mean of the indicators and an expected value;
- prec: the maximal absolute difference between the mean and the second greater indicator, and between the mean and the second lesser indicator.

These values can then be aggregated with the following process:

- mean: by the mean of the mean indicators;
- error: by the absolute mean of the error indicators;

- prec: by the maximal prec indicators.

5.4.3 Synthetic dataset generation: results

We present the obtained results of the synthetic generation of KD datasets given real ones.

Indicators

In this study, the durations are assumed independent, and the laws parameters, assumed to be correctly estimated by the fitness algorithm. The equivalency between synthetic keystroke samples and real keystroke samples should be guaranteed by the proposed model, and has thus to be verified.

Three performance metrics are used to qualify the capacity of the generated synthetic samples to match samples that would have been expected:

- Area Between the Curves (ABC): qualify the capacity of the synthetic datasets to estimate the ROC curves of real datasets (the lesser, the better);
- EER estimation error (EEE): qualify the capacity of the synthetic datasets to estimate the EER of real datasets (the lesser, the better);
- EER of real against synthetic data (ERS): qualify the capacity of synthetic datasets to usurp users from real datasets (the greater, the better).

In order to compare our findings to the related work [Stefan et al., 2012, Stefan and Yao, 2010], we added one consistency strategy (6o) where the durations d_1, d_2, and $_4$ are computed, and all durations are positives. We used the normal law (StefN), and the uniform law (StefU), using the raw parameter estimation. As we work on fixed text, the Markov model is not used. We show in the following sections that the uniform law gives poor performances, as expected.

These three indicators are detailed in the following sections. As shown in Figures 5.10 and 5.12, best results for the configuration gumbel.5 are found for R_mge and R_qme fitness algorithm, while the raw fitness algorithm gives the worst results. The use of only 23 elements per set seems surprisingly to give slightly better results than using 45 elements. This might be due to the fact that users' ways of typing evolve with time. The use of R_mge fitness algorithm will thus be, by default, assumed in the following sections, as for the use of 45 elements per sets.

As shown in Figures 5.11 and 5.13, results highly depend on the dataset and the used distance function. For example, GREYC W1 dataset with Blesha distance gives an EER estimation error of 0.069 using gumbel.5, 45 elements per sets, and R_mge fitness algorithm while the best configuration, for this dataset and distance, is normal-3s.1c with an EEE of 0.026, which performs poorly, on the same dataset, with the Hocquet distance with an EEE of 0.148.

As shown in Figures 5.14 and 5.15, the selection of the configuration is a trade-off between EEE and ERS, although some configurations give both satisfying EEE and ERS.

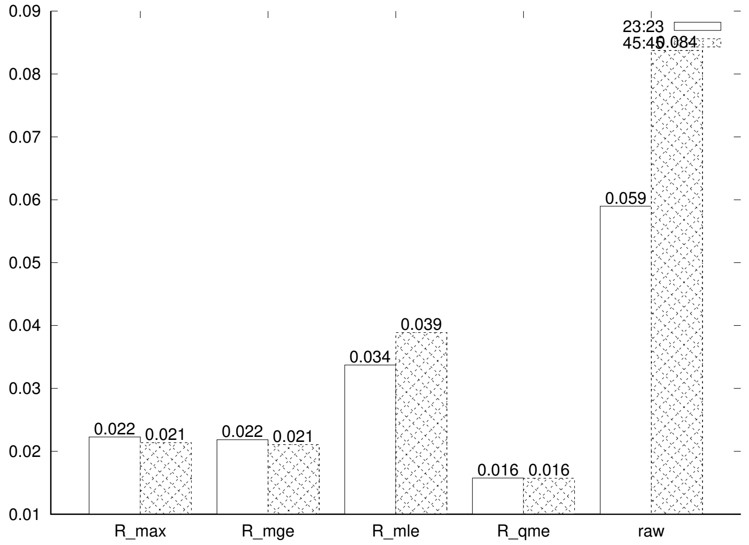


Figure 5.10: EER estimation error (EEE) using R_mge, gumbel.5, and 45 elements per sets.

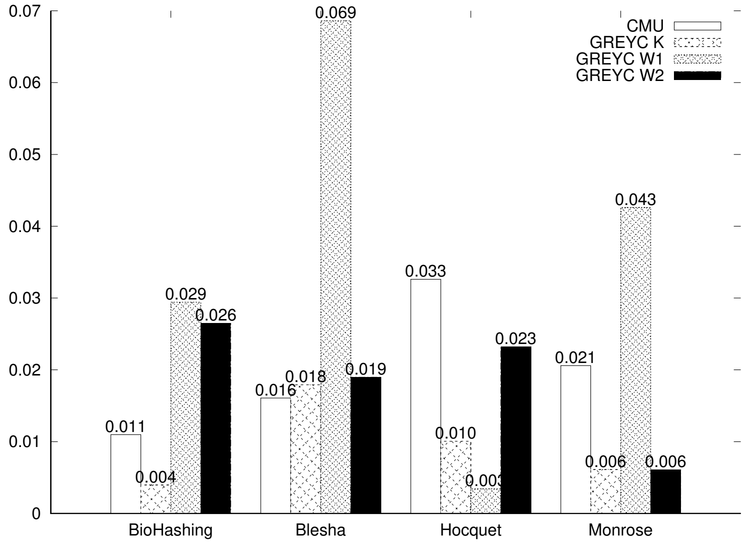


Figure 5.11: EER estimation error (EEE) using R_mge, gumbel.5, and 45 elements per sets.

ROC curve estimations

The Area Between the Curves (ABC), computed from the synthetic (DataS) and real (DataU) entries, qualify the capacity of the synthetic datasets to estimate the ROC curves of real datasets.

Table 5.6 shows the best configurations that minimize the ABC. For each configuration, the first line describes the mean distance between the synthetic and the real ROC curve, then the ABC, then prec, the maximal variation of the synthetic ROC curves relatively to its mean. The second line gives the ERS with its mean, error,

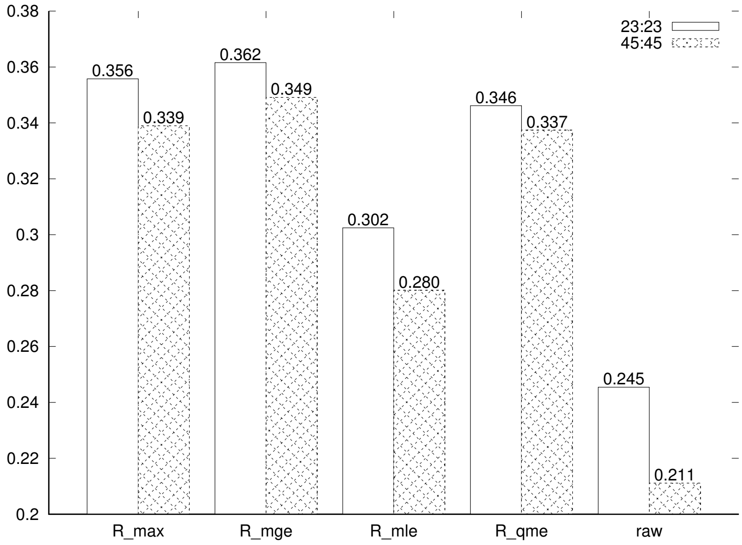


Figure 5.12: EER of real against synthetic data (ERS) using R_mge, gumbel.5, and 45 elements per sets.

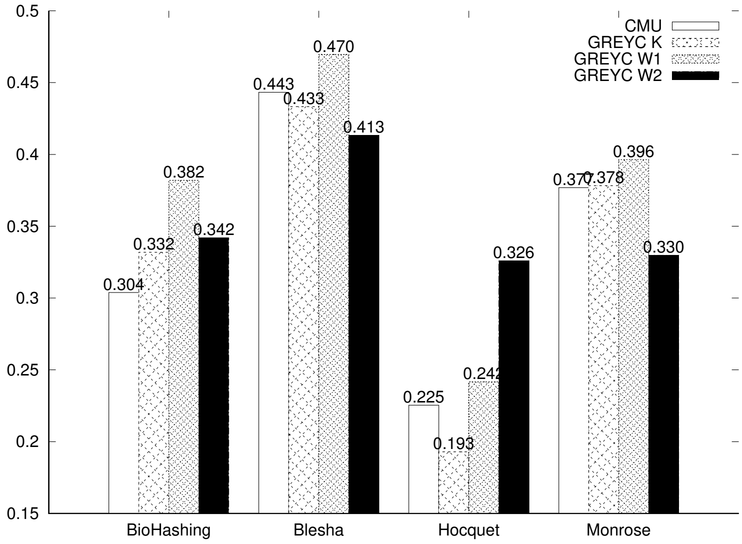


Figure 5.13: EER of real against synthetic data (ERS) using R_mge, gumbel.5, and 45 elements per sets.

and then prec.

As shown in Table 5.6, the ROC curve can be estimated with a great accuracy (ABC of 0.027 with a prec of 0.095). The best ABC values are obtained with the gumbel law, and with strategies 5, null, and 0 which, as said in the previous section does not generate d_5, but compute it from the other durations. Removal of aberrant values when estimating the parameters (-3s) does not seem to benefit the ABS. Fully consistant strategies are missing from this top. R_qme is over represented in this top. Best configurations in ABS have lower performances in ERS (> 0.10 instead of ~0.02).

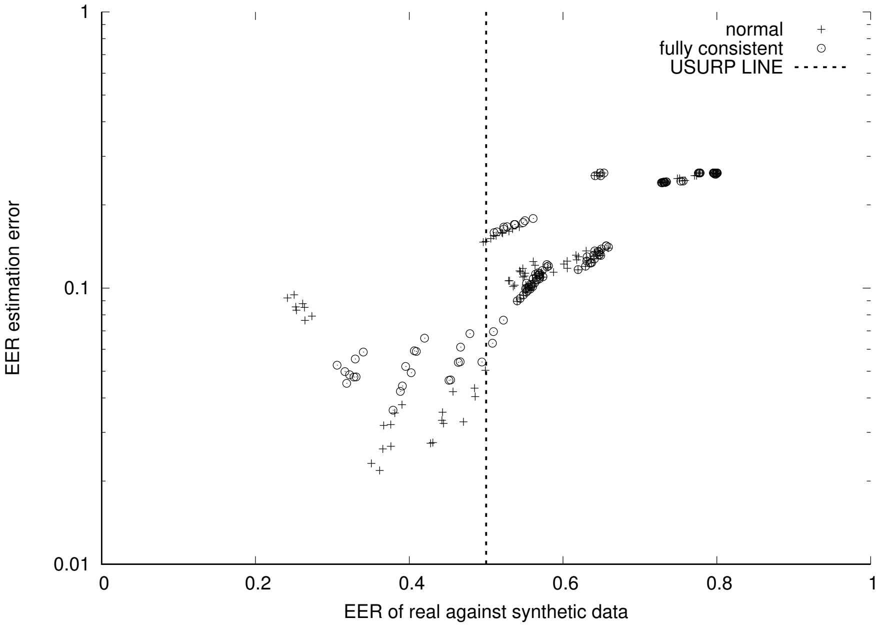


Figure 5.14: Performances of configurations with sets of 23 elements (using R_mge)

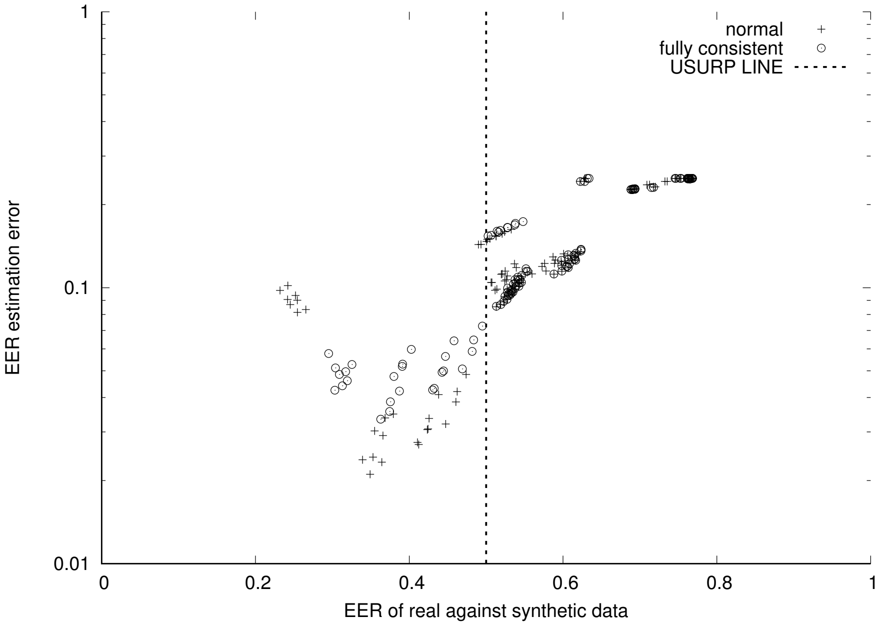


Figure 5.15: Performances of configurations with sets of 45 elements (using R_mge)

The best configurations to estimate the ROC curves of real datasets has been found to be gumbel.5 (using R_qme), followed by gumbel-3s.5 (using raw). The estimation of the ROC curves with gumbel.5 and R_qme is shown in Figure 5.16

	Sets of 23 elements		Sets of 45 elements	
1	R_qme:gumbel.null	0.045 (0.025±0.095) 0.415 (0.085±0.021)	R_qme:gumbel.5	0.045 (0.027±0.095) 0.337 (0.163±0.009)
2	R_qme:normal_gumbel.null	0.045 (0.026±0.089) 0.410 (0.090±0.021)	raw:gumbel-3s.5	0.045 (0.028±0.089) 0.326 (0.174±0.009)
3	R_qme:logistic_normal.null	0.044 (0.026±0.086) 0.437 (0.065±0.017)	R_qme:logistic_gumbel.null	0.053 (0.029±0.085) 0.433 (0.067±0.009)
4	R_qme:logistic_gumbel.null	0.049 (0.027±0.084) 0.454 (0.054±0.017)	R_qme:gumbel.null	0.049 (0.029±0.080) 0.400 (0.100±0.012)
5	R_qme:gumbel.0	0.045 (0.027±0.080) 0.344 (0.156±0.013)	R_qme:gumbel-3s.5	0.046 (0.029±0.086) 0.354 (0.146±0.008)
6	R_qme:normal.null	0.049 (0.028±0.085) 0.396 (0.104±0.018)	R_qme:logistic_normal.null	0.050 (0.029±0.087) 0.420 (0.080±0.013)
7	R_qme:normal_gumbel-3s.null	0.047 (0.028±0.089) 0.430 (0.071±0.017)	R_qme:normal_gumbel.null	0.049 (0.029±0.086) 0.396 (0.104±0.011)
8	R_qme:gumbel.5	0.046 (0.028±0.087) 0.346 (0.154±0.013)	R_qme:gumbel-3s.0	0.048 (0.029±0.090) 0.346 (0.154±0.012)
9	R_qme:normal-3s.null	0.048 (0.029±0.084) 0.418 (0.082±0.021)	R_qme:gumbel.0	0.047 (0.030±0.079) 0.329 (0.171±0.009)
10	R_qme:gumbel-3s.null	0.048 (0.029±0.085) 0.433 (0.068±0.013)	R_qme:normal.5	0.049 (**0.030**±0.077) 0.323 (0.177±0.010)

Comparison with the related work, using sets of 45 elements				
	With all values		With exclusion of aberrant values	
	raw:StefN.6o	0.057 (**0.035**±0.080) 0.389 (0.111±0.009)	raw:StefN-3s.6o	0.112 (**0.078**±0.093) 0.496 (0.029±0.008)
	raw:StefU.6o	0.261 (**0.165**±0.082) 0.640 (0.140±0.008)	raw:StefU-3s.6o	0.305 (**0.181**±0.072) 0.691 (0.191±0.010)

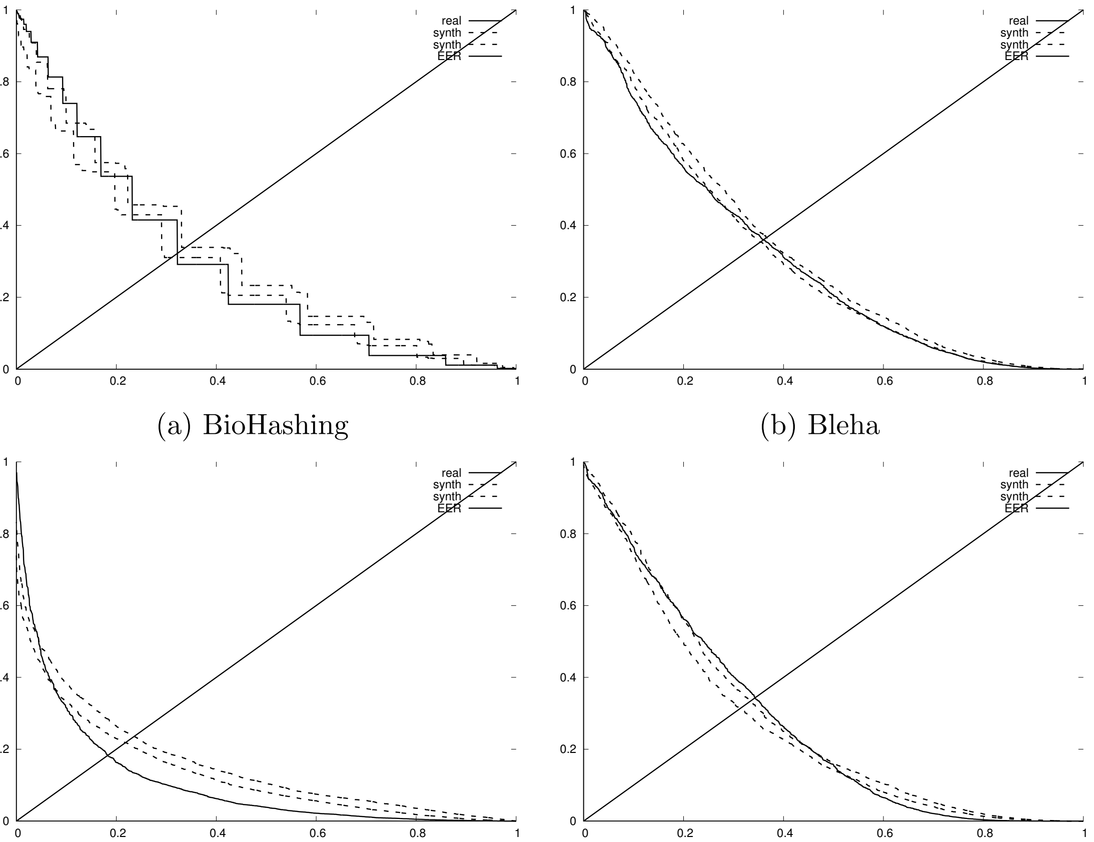
Table 5.6: TOP10 configurations that minimize the area between the ROC curves (ABC). In the first line, the ABC value, in the second line we present the ERS value is given. Each line contains the absolute value, the error, then the precision.

Usurpation of keystroke dynamics

The EER value computed from DataSU (ERS) is used to qualify the capacity of synthetic Keystroke dynamics data to be indistinguishable from real Keystroke dynamics data. As the EER corresponds to configuration of the biometric system when FAR equals the FRR, it is not possible to set a threshold enabling to reject less than EER % of genuine users, without accepting less than EER % impostors. Thus, with an EER of 50%, it is not possible to set a threshold that reject of accept users better than random. With a, EER > 50%, more impostors will be accepted than genuine users.

However, a biometric system with an EER < 50% can be trivially built from an existing one having an EER > 50%, simply by considering distance scores as similarity scores, i.e. by rejecting users below, instead of rejecting them over, a given threshold. Meaning that for each biometric system with an EER of X, one can build a biometric system with an EER of 1 - X.

In this study, we aim at building synthetic Keystroke dynamics data that are



(a) BioHashing

(b) Bleha

(c) Hocquet

(d) Monrose

Figure 5.16: ROC curves for the CMU dataset with 45 entries per users, using R_qme and gumbel.5.

indistinguishable (using the 4 distances functions we study) from real one, i.e. maximizing the minimum of ERS and 1 - ERS, i.e. getting an ERS as close as 50%. Obviously, if the Keystroke dynamics sample contains aberrant values, it would be easily detected. Thus, fully consistent strategies are desired.

Table 5.7 shows the best configurations that minimize the ERS error (i.e. $|ERS - 0.50|$). For each configuration, the first line describes the ERS with its mean, error, and then prec, and the second line the synthetic data EER with its mean, EEE, and then prec.

The best usurpations are obtained with either the gumbel or the normal law for strategies 6, 7, 4, and 2. None of these strategies recomputes d_5. Removal of aberrant values when estimating the parameters (-3s) seems to benefit the usurpation. R_max and R_mge are missing from these top. However, as already shown in the previous section, the best configurations in usurpation have poor results in EER estimation, with an EEE > 0.05, which is still high.

	Sets of 23 elements		Sets of 45 elements	
1	R_max:logistic_normal.nullc	0.498 (0.033±0.013) 0.199 (0.065±0.016)	raw:normal-3s.6	0.500 (0.021±0.011) 0.175 (0.075±0.011)
2	R_qme:normal.4	0.506 (0.034±0.021) 0.182 (0.080±0.013)	raw:normal-3s.6c	0.501 (0.022±0.010) 0.172 (0.077±0.010)
3	R_mle:normal_gumbel.6c	0.505 (0.034±0.019) 0.192 (0.070±0.013)	R_mle:normal-3s.6	0.499 (0.022±0.010) 0.175 (0.074±0.010)
4	R_mle:normal_gumbel.6	0.505 (0.034±0.019) 0.193 (0.069±0.012)	R_mle:normal-3s.6c	0.501 (0.022±0.010) 0.173 (0.077±0.010)
5	R_max:logistic_gumbel.nullc	0.489 (0.034±0.017) 0.212 (0.053±0.014)	raw:normal_gumbel-3s.7	0.510 (0.024±0.011) 0.172 (0.078±0.012)
6	R_qme:normal.6	0.513 (0.035±0.015) 0.178 (0.084±0.013)	raw:normal_gumbel-3s.7c	0.511 (0.024±0.011) 0.171 (0.078±0.012)
7	R_mle:normal-3s.4	0.519 (0.035±0.016) 0.179 (0.082±0.014)	R_qme:gumbel.4	0.500 (0.024±0.012) 0.169 (0.080±0.010)
8	R_qme:logistic_normal-3s.nullc	0.489 (0.035±0.015) 0.207 (0.057±0.015)	R_mle:normal-3s.7c	0.495 (0.025±0.010) 0.176 (0.074±0.009)
9	R_qme:normal.4c	0.510 (0.035±0.023) 0.175 (0.087±0.017)	raw:normal-3s.2c	0.499 (0.025±0.010) 0.173 (0.076±0.011)
10	raw:normal-3s.4	0.518 (0.035±0.013) 0.180 (0.082±0.015)	R_mle:normal-3s.2c	0.499 (**0.025**±0.008) 0.173 (0.077±0.010)

Comparison with the related work, using sets of 45 elements				
	With all values		With exclusion of aberrant values	
	raw:StefN.6o	0.389 (**0.111**±0.009) 0.246 (0.021±0.010)	raw:StefN-3s.6o	0.496 (**0.029**±0.008) 0.177 (0.072±0.010)
	raw:StefU.6o	0.640 (**0.140**±0.008) 0.062 (0.188±0.007)	raw:StefU-3s.6o	0.691 (**0.191**±0.010) 0.028 (0.221±0.006)

Table 5.7: TOP10 configurations that enables good usurpation (ERS). In the first line, the ERS value, the second line corresponds to the EEE value. Each line contains the absolute value, the error, then the precision.

As shown by the symmetric of the FAR/FRR curves in Figure 5.17, our proposed Keystroke generation method is thus able to produce synthetic samples that enable identity usurpation of a known user, by imitating its keystroke dynamics.

EER estimations

The difference between the EER values (EEE), computed from the synthetic (DataS) and real (DataU) entries, qualify the capacity of the synthetic datasets to estimate the EER value of the real one. Note that the threshold, in which the EER value is reached, is not taken into account.

Table 5.8 shows the best configurations that minimize the EEE value. For each configuration, the first line describes the mean of the synthetic dataset EER with its EEE, and then prec, and the second line the ERS with its mean, error, and then prec.

As shown in Table 5.8, the EER value can be estimated with a great accuracy (EEE of 0.016 with a prec of 0.012). The best EEE values are obtained with the

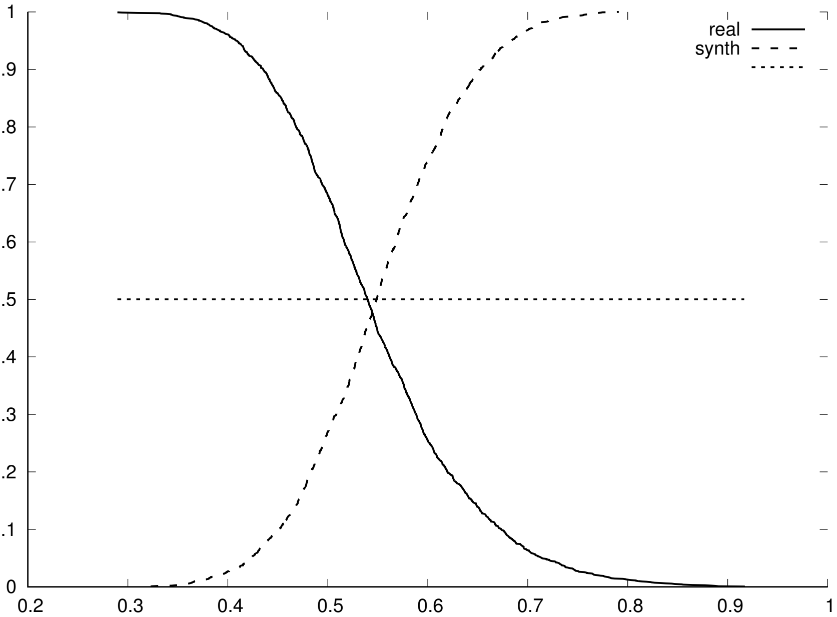


Figure 5.17: FAR/FRR curves using Hocquet distance of real samples against synthetics samples generated with the configuration logistic_gumbel-3s.nullc using R_mge, with sets of 45 elements.

gumbel law, and with strategies 5, 2, 7, and 6. Removal of aberrant values when estimating the parameters (-3s) does not seem to benefit to the EEE. R_mge and R_max are missing from this top. As already shown in previous sections, the best configurations in EEE have lower performances in ERS (> 0.12 instead of ~0.021).

The best configurations to estimate the EER value of real datasets has been found to be gumbel.5 (using R_qme), followed gumbel-3s.5 (using R_qme).

5.5 Conclusion and perspectives

In this chapter, we presented a method that enables the generation of synthetic keystroke dynamics data from known Users, to either usurp real user KD, or to estimate the EER value of a KDS. These methods have been tested on fixed text, but could be as well applied to free text.

We show that, the best estimation of the EER value of a KDS is met when using gumbel laws, without exclusion of aberrant values, and by computing durations d_5, d_1, and d_2 from other durations instead of generating them (gumbel.5). However, even though some configurations have satisfying performances in both usurpation and EER estimation, our findings show that the generation of synthetic keystroke dynamics is a trade-off between an optimal EER estimation, and an optimal usurpation capability.

	Sets of 23 elements		Sets of 45 elements	
1	R_qme:gumbel.0	0.258 (0.015±0.016) 0.344 (0.156±0.013)	R_qme:gumbel.5	0.251 (0.016±0.012) 0.337 (0.163±0.009)
2	R_qme:gumbel.5	0.255 (0.016±0.015) 0.346 (0.154±0.013)	R_qme:gumbel-3s.5	0.238 (0.017±0.012) 0.354 (0.146±0.008)
3	R_qme:normal.0	0.264 (0.018±0.015) 0.329 (0.171±0.015)	raw:gumbel-3s.5	0.262 (0.017±0.010) 0.326 (0.174±0.009)
4	R_qme:normal.5	0.260 (0.019±0.017) 0.331 (0.169±0.014)	R_qme:normal.5	0.254 (0.018±0.011) 0.323 (0.177±0.010)
5	R_qme:gumbel.null	0.259 (0.019±0.015) 0.415 (0.085±0.021)	R_mle:normal.2c	0.244 (0.019±0.013) 0.378 (0.122±0.009)
6	R_qme:gumbel-3s.0	0.246 (0.019±0.015) 0.361 (0.139±0.011)	raw:normal.7c	0.247 (0.019±0.012) 0.375 (0.125±0.012)
7	R_qme:normal_gumbel.null	0.261 (0.019±0.016) 0.410 (0.090±0.021)	raw:normal.6c	0.245 (0.019±0.013) 0.380 (0.120±0.008)
8	raw:gumbel-3s.5	0.257 (0.019±0.013) 0.348 (0.152±0.014)	R_mle:normal.6c	0.244 (0.019±0.013) 0.380 (0.120±0.008)
9	raw:gumbel-3s.0	0.261 (0.019±0.016) 0.342 (0.158±0.015)	raw:normal.2c	0.245 (0.019±0.013) 0.378 (0.122±0.009)
10	R_qme:logistic_normal.null	0.257 (0.021±0.012) 0.437 (0.065±0.017)	R_mle:normal.7c	0.247 (**0.019**±0.015) 0.376 (0.124±0.011)

Comparison with the related work, using sets of 45 elements				
	With all values		With exclusion of aberrant values	
	raw:StefN.6o	0.246 (**0.021**±0.010) 0.389 (0.111±0.009)	raw:StefN-3s.6o	0.177 (**0.072**±0.010) 0.496 (0.029±0.008)
	raw:StefU.6o	0.062 (**0.188**±0.007) 0.640 (0.140±0.008)	raw:StefU-3s.6o	0.028 (**0.221**±0.006) 0.691 (0.191±0.010)

Table 5.8: TOP10 configurations that minimize the mean of EER estimation error (EEE). In the first line, we show the EEE value, in the second line the ERS value is given. Each line contains the absolute value, the error, then the precision.

This work constitutes a first step towards the generation of large synthetic Keystroke dynamics datasets. The following step would be the generation of keystroke dynamics data for an unknown user. Such large synthetic Keystroke dynamics datasets could then be used to fairly compare KDS performances, as well as to improve KDS pre-processings. It constitutes a perspective of this work.

In short: *We generate entries for a given user. Our model assumes that Keystroke Dynamics does not evolve through time, and that durations are independent. However, this produces a trade-off between usurpation and EER estimation. Durations d_1 to d_4 seems to follow a gumbel law. Durations d_5 and d_0 are harder to generate, but can be computed from others. This model still needs to be improved.*

Chapter 6

Some examples of applications with proof of concept

This chapter explores applications of our previous contributions, as well as security enhancement through trusted devices.

Keywords: *Social Identity Proof, Social Networks; Trust; Hardware security; Security architectures; Offline Trusted Device Proxy Architecture; TLS switching.*

Contents

Contributions presented in this chapter

- Social identity proof;
- KDAS on trusted devices;
- PICRP on trusted devices;
- Protecting PICRP with trusted devices;

Publications

- Buccafurri, F., Lax, G., Migdal, D., Nicolazzo, S., Nocera, A., and Rosenberger, C. (2017a). Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *CyberWorlds (B - Core)*, Chester, United Kingdom.
- Buccafurri, F., Lax, G., Migdal, D., Nicolazzo, S., Nocera, A., and Rosenberger, C. (2018). Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *Italian Conference on CyberSecurity (ITASEC)*, Milan, Italy.
- Migdal, D., Johansen, C., and Jøsang, A. (2017). Offline Trusted Device and Proxy Architecture based on a new TLS Switching technique. In *International Workshop on Secure Internet of Things SIOT 2017 (ESORICS Workshop - A - Core)*, Oslo, Norway.
- Migdal, D. and Jøsang, A. (2017). OffPAD – Objet Personnel d'Authentification Hors-ligne appliqué aux hôpitaux et banques en ligne. In *RESSI 2017*, Grenoble, France.

6.1 Introduction

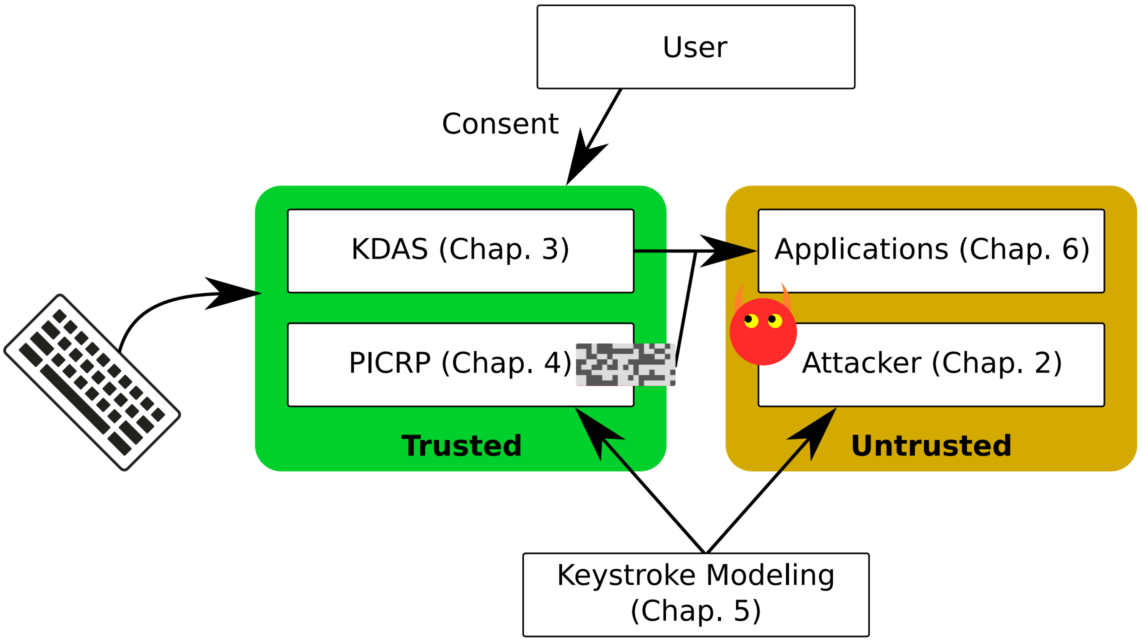
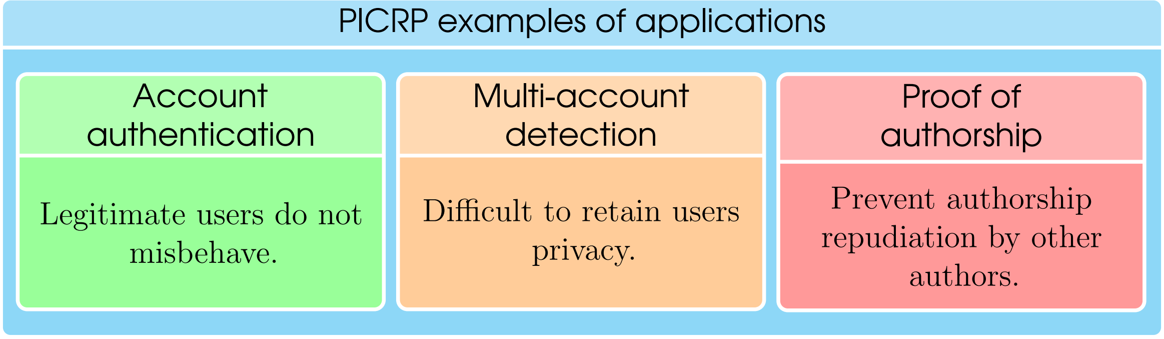


Figure 6.1: Complete pipeline.

Our contributions enable consented user authentication based on Keystroke Dynamics (with other modalities) while protecting user privacy. The full pipeline is presented in Figure 6.1. The Keyboard events are collected under a trusted component (could be a WebExtension, the browser, the OS, or a dedicated hardware) preventing the untrusted component (could be a web page, or the whole computer) to access them.

Keystroke events are anonymized with a Keystroke Dynamics Anonymisation System (see Chapter 3 in order to transmit the meaning of the keyboard event without their true timing information, thus preventing user profiling (authentication, identification, soft biometrics) without its consent (see Chapter 2). Under the user consent, PICRP are computed (see Chapter 4) and sent to the applications in order to enable user authentication. Keystroke Modeling (see Chapter 5) is a first step towards improvement of PICRP and attacks performances. In this chapter, we focus on the application of our PICRP contribution, its integration with dedicated hardware and the associated proof of concept we developed.

In Chapter 4, we presented an authentication protocol that can be used to straighten *Account authentication.* For example, Same/Fixed-Text can be used during login processes while the user types its credentials. Once the user logged in into its account, Free-Text can be used, either for continuous or punctual authentication while the user type a message, e.g. on a chat, forum, or blog, thus preventing lunch-time attacks. If the authentication fails, either a challenge can be issued for



Legitimate users do not misbehave.

Difficult to retain users privacy.

Prevent authorship repudiation by other authors.

Figure 6.2: PICRP examples of applications

the user to prove its identity, or the user can be automatically disconnected. In such scenarios, attackers will try to impersonate legitimate users while legitimate are consenting to the authentication and have no interest into misbehaving.

As shown in Figure 6.2, we can consider two more applications. *Multi-account detection* where the service provider aims to detect users possessing several accounts, and *Proof of authorship* where users tries to prove authorship of a written text, but can also misbehave.

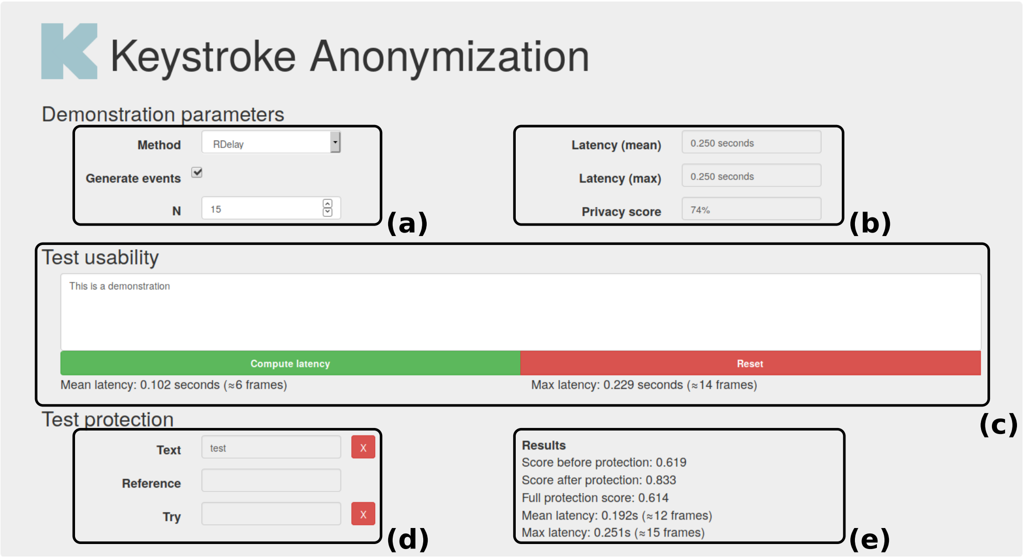
In the scope of this PhD thesis, several proof of concepts have been developed. They are available online on `https://trust.greyc.fr`. A WebExtension gathering all functionalities will also be made available.

Note: *Presented softwares might be subject to modifications.*

6.2 Anonymization

First, we consider the Anonymisation of keystroke dynamics to protection users privacy. We implemented a proof of concept of Keystroke Dynamics Anonymization System techniques presented in Chapter 3. The Graphical User Interface (GUI) is presented in Figure 6.3.

The KDAS can be configured in the area (a) by selecting the type of KDAS and its parameter p. The performances of the KDAS are given in the area (b), the privacy score is computed as $2 * EER$. Performances are computed from the known fixed-text datasets. Usability of the selected KDAS can be experienced through the area (c) where the user can type a free-text while its Keystroke Dynamics is anonymized. The latency observed during typing of the free-text is shown in real-time. The level of protection offered by the KDAS can also be experienced through the area (d). The user selects a fixed-text and types it several times, 3 as references, and 3 as samples. Performances are then shown in area (e).



Keystroke Anonymization

Demonstration parameters

Method: RDelay

Generate events: ☑

N: 15

Latency (mean): 0.250 seconds

Latency (max): 0.250 seconds

Privacy score: 74%

Test usability

This is a demonstration

Mean latency: 0.102 seconds (≈6 frames)

Max latency: 0.229 seconds (≈14 frames)

Test protection

Text: test

Reference:

Try:

Results
Score before protection: 0.619
Score after protection: 0.833
Full protection score: 0.614
Mean latency: 0.192s (≈12 frames)
Max latency: 0.251s (≈15 frames)

Figure 6.3: Anonymization demonstration interface.

As for the WebExtention version, Keystroke Dynamics is anonymized on all web pages. Default behaviors are not entirely implemented, and modifiers keys are not delayed, see Section 3.3.1.

6.3 Account authentication

The PIRCP signature allows the user authentication for logical access control applications. In this developed proof of concept, we show the computation and comparison of two PICRP proposed in Chapter 4. The GUI is presented in Figure 6.4.

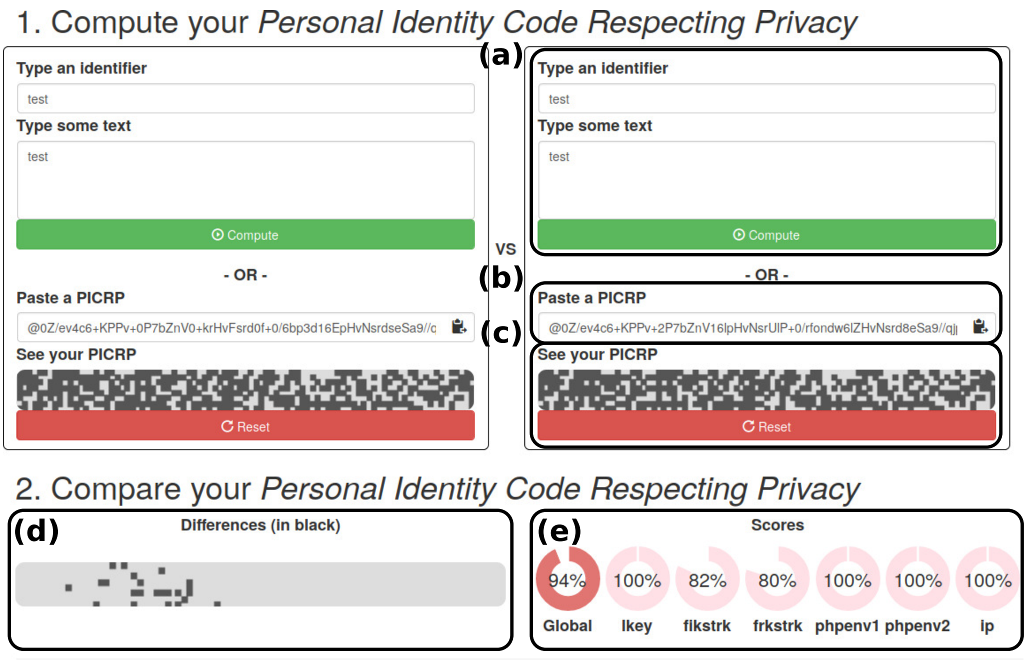
Users are invited to give two PICRP in order to compare them. They can either generate them (a) or paste one (b). A visual representation is then given in the area (c). In order to generate them, users type a fixed-text that is used as a secret. The generated PICRP is then printed in field (b).

The differences between the two PICRP are shown in the area (d), while similarity scores computed from the PICRP Hamming pseudo-distance are shown in area (e). Similarity scores can only be computed if the PICRP are generated, or otherwise, if the first version of PICRP is used.

6.4 Proof of authorship

Instead of being used for account authentication, PICRP could also be used as proof of authorship on a written text.

We developed a proof of concept to verify the authorship of a document. It is implemented as a continuous authentication solution based on PICRP and Free-Text



1. Compute your *Personal Identity Code Respecting Privacy*

Type an identifier: test

Type some text: test

Paste a PICRP: @0Z/ev4c6+KPPv+0P7bZnV0+krHvFsrd0f+0/6bp3d16EpHvNsrdseSa9//c

VS

Type an identifier: test

Type some text: test

Paste a PICRP: @0Z/ev4c6+KPPv+2P7bZnV16lpHvNsrUlP+0/rfondw6lZHvNsrd8eSa9//qj

2. Compare your *Personal Identity Code Respecting Privacy*

Global	Ikey	fikstrk	frkstrk	phpenv1	phpenv2	ip
94%	100%	82%	80%	100%	100%	100%

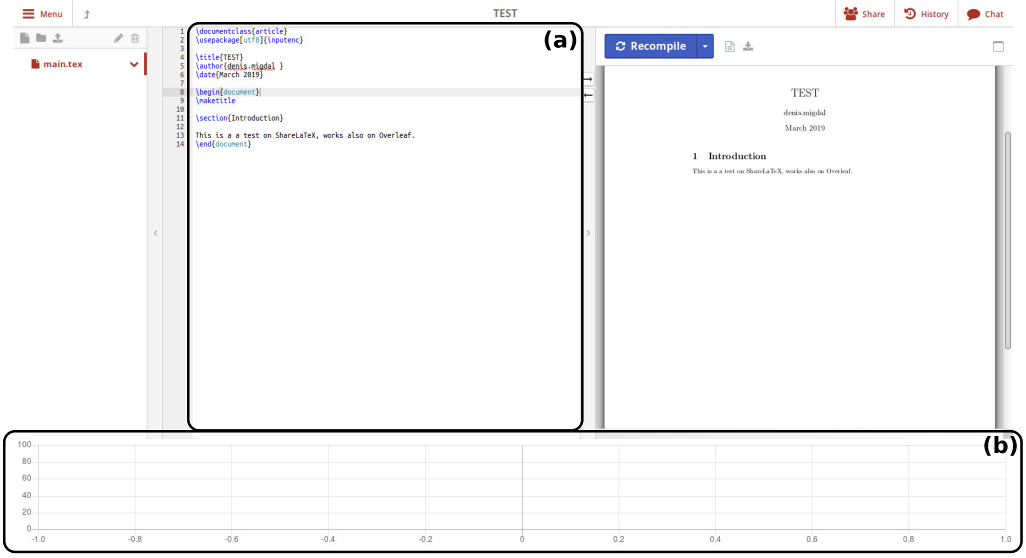
Figure 6.4: Authentication demonstration interface.

Keystroke Dynamics. The GUI is presented in Figure 6.5. The shown interfaces are based on ShareLaTeX, an online collaborative platform used to edit LaTeX documents.

The online demonstration is based on a self-hosted ShareLaTeX instance. The demonstration integrated to the WebExtension works on any ShareLaTeX and Overleaf instances. As we do not save any information in this demonstration, we do not use these fields to straighten authentication by using Fixed-Text-based PICRP or BioCode. As shown in Figure 6.5, Keyboard Events are collected on the area (a). A BioCode is then computed from the typed Free-Text, using only the last 125 typed digraph. A similarity score is then computed using a sliding window. The score of all users typing in the current document is shown in the area (b).

In this demonstration, we thus use author continuous authentication to prevent repudiation of one author contribution by other malicious authors, as well as identity theft. However, this does not protect against ghostwriting, or fake accounts. Ghostwriting might be detected as, either a large insertion of copied text, a failed authentication, or through the ghostwriter identification. However, as the legitimate owner of the account is complicit, it can automatically produce authentication proofs for the ghostwriter.

The use of a trusted device authenticated and trusted by the service could harden such complicit usurpation. Else, if PICRP are computed on the service, and the keystroke sent in real-time, the service could ensure PICRP are computed from the typed content, and might try to detect synthetic or replayed keystroke, as well as the manual transcript of a text.



```latex
\documentclass{article}
\usepackage[utf8]{inputenc}

\title{TEST}
\author{denis.migdal }
\date{March 2019}

\begin{document}
\maketitle

\section{Introduction}

This is a a test on ShareLaTeX, works also on Overleaf.
\end{document}
```

TEST

denis.migdal

March 2019

1 Introduction

This is a a test on ShareLaTeX, works also on Overleaf.

Figure 6.5: Online Editor demonstration interface.

6.5 Social identity proof

As presented in the previous sections, P2P authentication requires Trust On First Usage (TOFU). Indeed, any peer must first assume that it communicates with the appropriate user before being able to authenticate him in future exchanges. We tackle down this issue by proposing in this section a social proof of identity enabling existent peers to certify the user identity to new peers.

We illustrate this proposition in the context of a social network, where new users want to befriend users they know. However, they are unsure whether the account is legitimate or if the claimed identity has been usurped. The social proof of identity solve this issue.

Note: *This social proof of identity originates from a collaboration with the University of Reggio Calabria. A paper [Buccafurri et al., 2017b] was published in the scope of this collaboration.*

6.5.1 Background

Motivations

In some cases, being able to authenticate past interlocutors is enough. In such cases, the identity of the interlocutor is built as the way the user perceive its interlocutor through their previous exchanges. *De facto*, the identity of an interlocutor varies from one user to another. In simpler terms, the interlocutor's identity is defined through its acts.

However, as described in Figure 6.6 other cases requires to ensure the interlocutor identity, properties, or trust. Ensuring the interlocutor is who it claims to be is

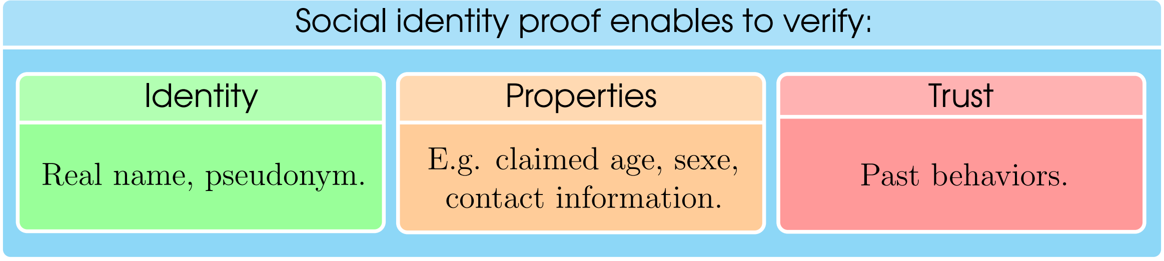

Social identity proof enables to verify:
Identity
Real name, pseudonym.
Properties
E.g. claimed age, sexe, contact information.
Trust
Past behaviors.

Figure 6.6: Social proof usages

required when the user seeks to communicate e.g. with a specific acquaintance it met or known in real-life or on another service. Impersonation of an acquaintance might lead to various scams, s.a. fake president fraud, or breach of trust. By giving private information to a stranger, this can also lead to thief, kidnapping, or to an abduction of a minor. Moreover the person impersonated might be hold liable, or have its reputation damaged.

Ensuring the interlocutor is what it claims to be might be mandatory for the user security or to prevent user deception. For example, sexe verification could prevent some of the romance scams, in which attackers seduce users to extort money. Age verification could ensure that grown-up adults does not pose as minor when exchanging with minors. Contact information are also necessary to make the interlocutor liable and to initiate legal proceedings.

Finally, ensuring the interlocutor past behaviors enables to trust well-behaved interlocutors while being wary of misbehaving interlocutors. Interlocutor reputation is essential in some contexts, s.a. on online markets. In such a way, bad and good behaviors are punished or rewarded through peer reputation, thus creating an incentive to good behaviors.

Formalism

As shown in Table 6.1, we distinguish between the theoretical model (abstraction), and the model implementation (implementation). For example, a node in the theoretical model represents an account/user, while a trust relation is implemented as a certificate. The concepts of trust/certificate chains/graphs are defined in the next sections. A confirmed node/account is a node/account whose owner identity has been confirmed by our proposed social identity proof.

Security requirements

Our proposed approach aims at reinforcing trust in social entities interactions, by ensuring that an entity who and what it claims to be. We thus aim at preventing attacks s.a. *identity usurpation* (i.e. using an existing identity), *typo-squatting* (i.e. using a very similar name), and *fictitious identities*.

Abstraction	**Implementation**
Node	Entity/Account/User
Edge/Trust Relation	Certificate
Claim to recognize/trust	Certify
Recognized/trusted	Certified
Recognizer/truster	Certifier
Trust chain	Certificate chain
Trust graph	Certificate graph
Confirmed	Confirmed

Table 6.1: Formalism

Reputations systems sanction users past behaviors through a gain or loss of trust. They are based on the assumption that users stay in the system, expect future interaction requiring trust, that treason would cost more than the expected profits, and that users looks to their long terms interests (requiring trust) instead of their short terms benefices (treason).

Reputations systems are also vulnerable to several attacks, s.a. the use of several identities, trust manipulations, or competitive misbehavior. Misbehaving users might recreate a new identity to start with a clean reputation after a treason (whitewashing), or use dedicated accounts to their dishonest activities, s.a. stolen accounts. Trust can be manipulated through the creation of fake identities (Sybil attacks), or through social engineering techniques. Competition might also lead users to attack others reputations, e.g. through slanders or false flag attacks.

Our Social Proof of Identity is a form of reputation system as users behaviors are sanctioned through certifications, representing trust. However, we enforce users proper behavior, not only through the use of trust, but also by enabling moderators to revoke users anonymity when they misbehave.

Our approach being based on peers recognition/certification, it necessitates the following security properties in order to be effective:

- *Integrity:* one should not be able to forge or modify certificates in the name of another certifier.
- *Performance:* computations should be computed in a reasonable time and memory occupation.
- *Availability:* computations of trust should be possible even if some nodes are unavailable or corrupted/malicious.
- *Accountability:* entities should be held accountable of the certificates they issue, implying non-repudiation of the issued certificates.
- *Revocation:* entities should be able to revoke the certificates they issued.
- *Uniqueness:* an entity should not be able to certify a same entity several times, e.g. by creating multiple accounts.

Accountability prevents entities from certifying unrecognized, untrusted, or misbehaving accounts, e.g. in exchange of money. Revocation enables an entity to revoke certificates in cases of misbehaving entity, change of an account ownership (e.g. stolen/sold account) or a mistakenly issued certificate.

Our proposed approach should be protected from (or at least mitigate) the following well-known reputation attacks:

- *Collusion:* system should be resilient to collusions.
- *Sybil attacks:* see uniqueness.
- *Slanders:* malicious entities should not be able to remove an account confirmation.
- *Whitewashing:* a misbehaving entity should not be able to create a new account to start with a new reputation.
- Identity usurpation, typo-squatting, and fictitious identities.
- *Hacking:* a hacked account should not be able to issue or revoke certificates, and should no longer be considered as confirmed.

We also require our approach to respect users privacy. The system should minimize the amount of information it discloses, even in case of misbehaving accounts, e.g. the users biometrics, and who certifies who, should not be disclosed.

6.5.2 Social Identity Proof principles

Certificate chain

Servers are typically authenticated by browsers through certificate chains. When a browser establishes a new TLS connection with a server, the server typically sends a certificate chain as a proof of its identity. The certificate chain prove that a given public key is associated to a given identity. Then any entity knowing the private key is assumed possessing the certified identity.

A simplified version of a certificate chain is shown in Figure 6.7. The certificate, associated to a server, certifies the server public key and identity, while intermediate certificates, associated to a Certification Authority (CA), certify the validity of other certificates. The Root CA certificate, associated to a Root Certification Authority (Root CA/RCA), is installed in, and trusted by, browsers.

When receiving a certificate chain, the browser verifies, among other things, that the server certificate is indirectly certified by a known RCA certificate. Meaning that, considering certificates as nodes, and certify relations as oriented edges from the certified to the certifier, there exists a path from the server certificate node to the RCA certificate node. Certification is performed by signing the certified certificate with the private key whose public key is contained in the certifier certificate.

Certificate graph

Although certificate chains are used to verify server identities, they remain vulnerable to, e.g., typo-squatting, malicious certifiers, or private key thieves. In our scheme, biometrics is used as an additional authentication modality mitigating the impact of private key thieves. Indeed, if the certifiers revoke their certification if they do

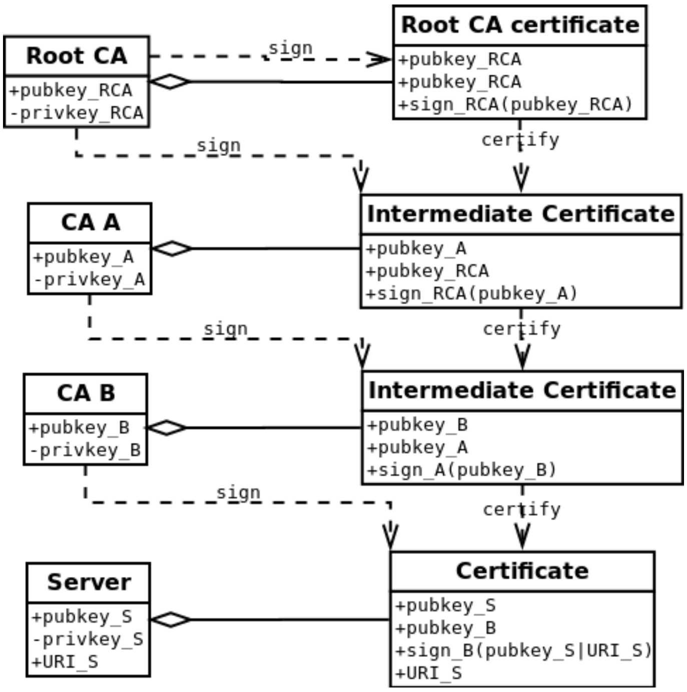


Figure 6.7: Certificate chain

not recognize the certified entity biometrics. In our context, typo-squatting attacks are assumed inefficient as certification is done by peers, upon recognition of the account/entity.

Still, attackers might become confirmed due to the presence of a misbehaving certifiers. For example a RCA belonging to a country can be used to issue dubious certificates enabling to spy on users by usurping servers identities. The private key of a CA might also be leaked, enabling attackers to forge certificates in its name, a misconfiguration might enable servers to be considered as a CA, enabling it to issue certificates, or a certifier might have little regards on the servers they certifies.

The accountability properties enable to punish misbehaving certifiers, however this does not prevent the consequences of such misbehavior. We thus require each nodes, in order to become confirmed, to be trusted, not by only one truster, but by t already confirmed trusters, as shown in Figure 6.8. We thus replace the trust chain by a trust graph, as shown in Figure 6.9.

Minimal trust subgraph

We defined the minimal trust subgraph as a subgraph containing only the confirmed nodes with the minimal number of trust relation so that the nodes remain confirmed, i.e. deletion of any trust relation, would make a node unconfirmed. This implies that as least t trust relations has to be deleted in order to disconnect the minimal trust subgraph. This means that each nodes needs t trust relations from the minimal trust subgraph in order to be confirmed.

As a node cannot be, at a given time, both confirmed and unconfirmed, an

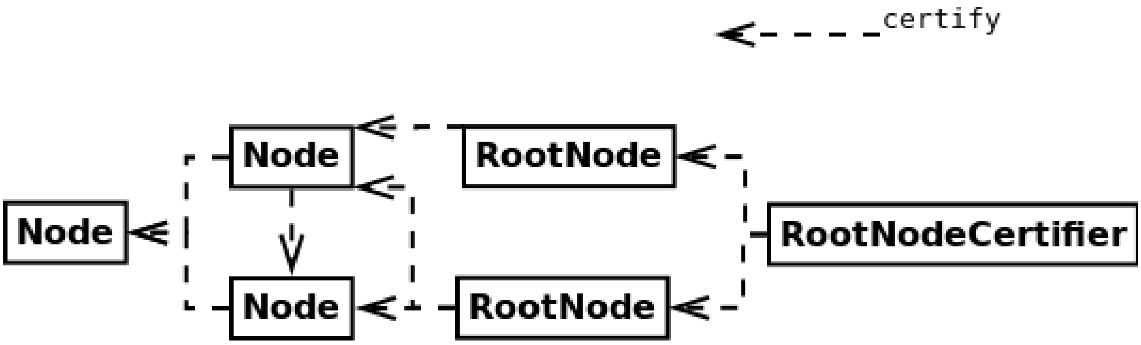


Figure 6.8: Trust graph with $t = 2$

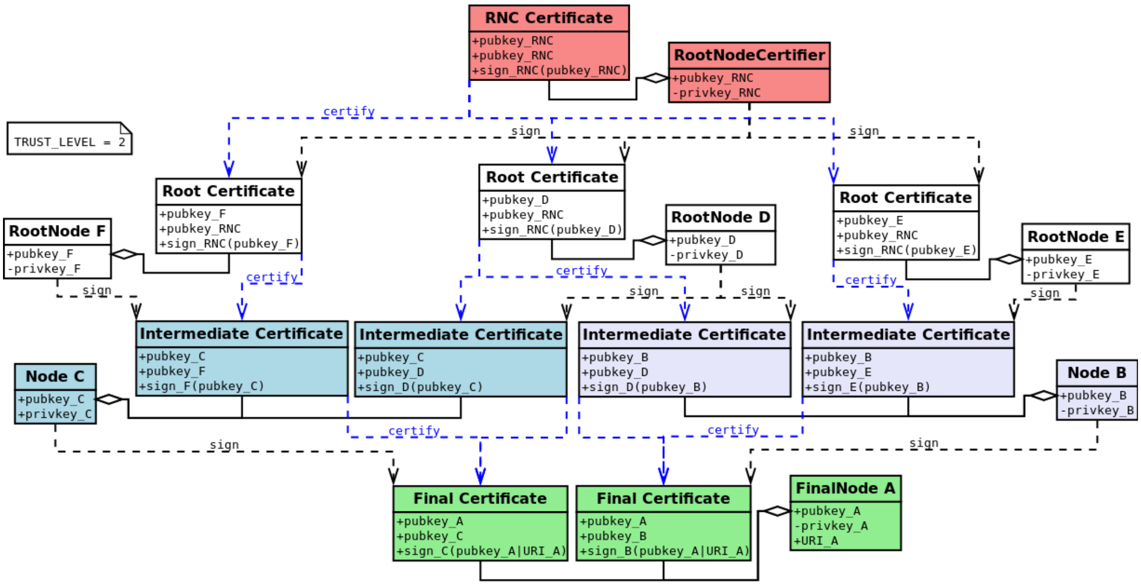


Figure 6.9: Certificate graph with $t = 2$

unconfirmed node cannot be in a minimal trusted subgraph, i.e. an unconfirmed node cannot become confirmed thanks to, or partially thanks to, the nodes it certifies, i.e. by construction, the minimal trust subgraph cannot contain loops. The minimal trust subgraph is thus a *directed acyclic graph.*

Resilience

As a node requires t trust relations from the minimal trust subgraph to be confirmed, and as each nodes can only trust a given node once, in order to arbitrary confirm nodes, at least t misbehaving confirmed nodes have to collude. This thus increases attacks cost and feasibility.

Our proposed method is vulnerable to sybils attacks were t attacker colludes to certify arbitrary nodes. However, the system is resilient and such account could be easily identified and deleted, once done, the system come back to its original state.

6.5.3 Social Identity Proof considerations

We present here some consideration about the Social Identity Proof. Algorithms are proposed in the next section.

Accountability

We assumed that Root Nodes are nodes trusted by a Root Node Certifier (RNC), whose task is to ensure Root Nodes real identities, and ensuring that they remains trusted. RNC can perform heavier verification for RN recognition, s.a. background checks, requiring an ID card, requiring a real-life appointment. We assumed that a node trusted by at least one RNC became a confirmed RN. However this work can easily be extended by requiring instead of one, at least t' RNC truster in order to become a RN.

In case of misbehavior, the certifier is required to disclose the certified node identity to the proper authority in order to held the certified node liable of its behavior. Any refusal to cooperate is in itself a misbehavior, thus requiring the certifier's certifier to disclose the certifier node identity. The certificate chain does not only represent a trust chain, but also a responsability chain. At the end, the RN, whose identity is verified by the RNC, will be held liable if no certifiers cooperate. Thus any victim of misbehavior would be able to legally recourse against at least one entity. Being able to sue somebody for the loss or prejudice suffered is very important in law in order to protect entities rights.

Minimizing disclosed information

When responding to a proof of identity query, a confirmed node sends a trust graph to the requester. However, a node cannot trust another several times, and cannot be present several times in a trust chain (i.e. loop in the minimal trust subgraph). In consequence, node linkability must be ensured in order to verify the trust graph. Meaning that the trust graph leaks the topology, and thus the trust relation between some identified nodes.

We define the minimal trust subgraph for a node (MTSN) as a trust graph in which the removal of any edges or nodes would make the node unconfirmed. i.e. the number of truster disclosed for a node is equal to the trust level t, thus limiting the amount of disclosed information.

A distinction should be made between intermediary and final certificates. When final certificates must contain the confirmed node identity, e.g. an URL, it is not required in intermediary certificates. Indeed, in the MTSN only the identity of the requested node matter. This means that each trust relation will be represented by two certificates, one enabling the trusted node to emit certificate, and one used as proof of its confirmed status.

Another measure is to prevent attackers from reconstructing the whole trust graph by asking all confirmed nodes for them MTSN. For example, when asking the MTSN, a requester must first send its own to the requested that will explicitly accept, or not, to respond to the requester query. As shown in Figure 6.10, for a friendship query on a social network, the requester (i) send a friendship query and its MTSN to the node it wants to befriend; (ii) the requested accept/reject the query; (iii) if the friendship query is accepted, the requested send its MTSN; (iv) the requester validate the friendship relation.

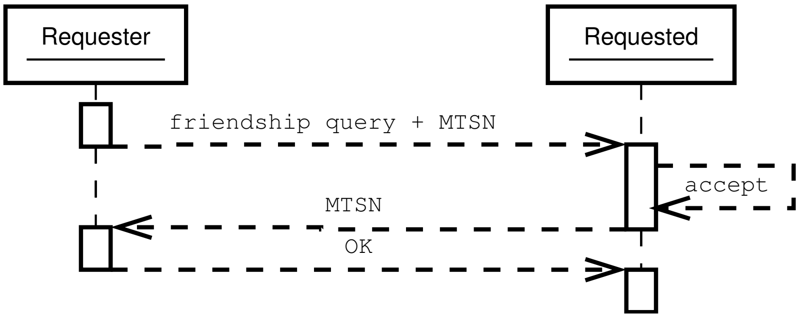


Figure 6.10: A friendship request

Unlikability of nodes across MTSN (MTSN unlikability), as well as between intermediary and final certificates, can also be used to prevent reconstruction of the trust graph from several MTSN. This can be achieved by issuing MTSN-dependant certificates, i.e. certificates, with different public keys would be issued for each MTSN.

Ergonomics

Users are an important part of a system security. If the system is not ergonomics or too costly for the users, they are likely to bypass the system, leading to breaches of security.

First, an incentive to certification should not be created, e.g. by giving extra-features to either the certifier or certified accounts. Indeed, such incentive may lead to dubious certifications if the entities estimate that the taken risks is worthy of the gain offered by such features. Such risk does not only impact the security of such misbehaving account, but also the security of all accounts present in the system.

In order to prevent false security feelings as well as putting too much trust into confirmed status, the absence of the confirmed status should be highlighted, instead of its presence. Moreover, the certification process should encourage and incite challenges through another medium, e.g. exchanging a short secret in real-life in order to validate the certification.

Finally, as previously stated, entities might have legitimate needs to posses several accounts. However, this would enable such entity to produce several certificates for a same account. We thus propose that secondaries account only need the certification of the confirmed primary account to become confirmed, however, such accounts would not be authorized to be certified by other accounts, and to certify an account already certified by the primary account, or another secondary account. The certificate produced by the primary account must be distinguishable from other certificates, e.g. by adding a field stating the secondary status of the certified account.

Certificates should contain a proof that no illegitimate multi-accounts are present in MTSN. For example through a GREYCHashing-based PICRP computed on, and signed by, a trusted device.

Trust graphs

Several variants of trust graphs and MTSN can be considered. For example certifications could include context(s) or level(s) of confidence. For example, certifying that a pseudonymous account whose identity originate from a specific community (e.g. forum, online game), or a type of account, s.a. institution, company, association, person. The level of confidence refers to the nature of the relation between the certified and certifier entities, e.g. colleague, friend in real-life, online friend, family member.

Entities might also set a maximal depth to MTSN and reject taller MTSN, i.e. only accepting confirmed nodes close to root nodes.

In the same way that users can add or remove RCA certificate on their browser, they should be able to do the same on MTSN, i.e. to add or remove RNC (RNC selection). Removal can be done if the RNC is not trusted anymore, and addition could be made for local MTSN, e.g. on a company, school, association, or other institution.

However, requester should not be able to choose which nodes or root nodes they trust, as they would be able to enforce the presence or absence of nodes the the MTSN they receive, and thus, to deduce information.

6.5.4 Social Identity Proof operations

Certification

One of the main operation of our system is to issue certificate for a certified-to-be account. Certificates issuing can be performed upon queries (e.g. MTSN unlikability, RNC selection), or only once, at the trust relation establishment. In both cases, confirmed nodes save their MTSN in order to give it upon queries, without regenerating it.

As shown in Figure 6.11 Certification consists in two phases,(i) an ascending phase, where the query is transmitted to the certifier nodes, up to the root nodes, and (ii) a descending phase, where the MTSM is built from the root nodes to the requester. The ascending phase is only necessary when certificates are issued upon queries.

The ascending phase starts when an account have enough truster in order to be confirmed. The query might contain a nonce (e.g. an url) dependant on the requester node (e.g. for MTSN unlikability), or a set of accepted RNC (e.g. for RNC

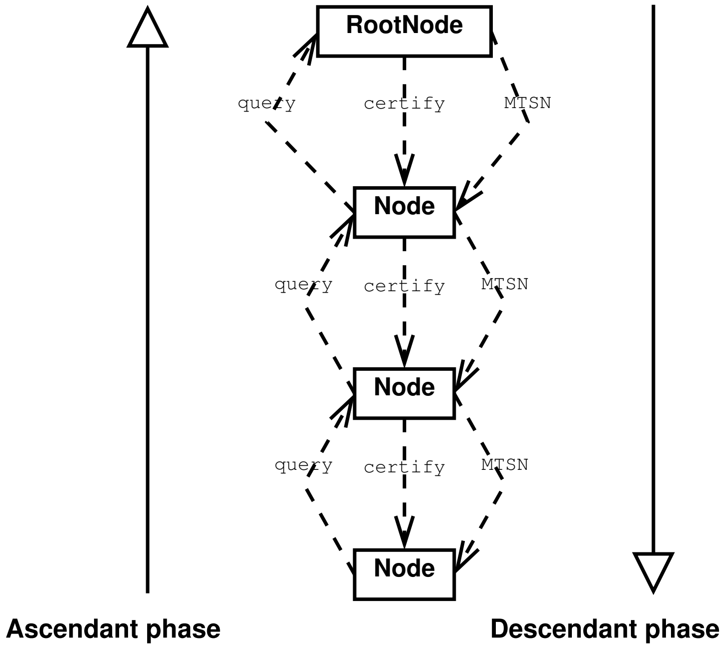


Figure 6.11: Certification process

selection). The nonce will be used to diversify the asymmetric keys of each entity in order to enable unlikability.

In the descendant phase, confirmed certifier accounts send its MTSN with its final and intermediary certificates, alongside with the intermediary and final certificate it issues for the certified account. The certified node verifies the received MTSN and certificates (cf nexts sections). The certified node stores the received elements. Then, if it becomes confirmed, i.e. having at least t certificate, the certified node merges t received MTSN and certificates, sorts the certificates (by the certified key) and removes duplicates. The selection of the MTSN to use can be random, e.g. for MTSN unlikability, be e.g. the smallest MTSN or the t oldest MTSN. Then the newly confirmed account issues certificates to the accounts it trusts, and in the case of upon-query certification, only to the nodes that queried him.

Merging of MTSN and certificates is illustrated in Figure 6.12. Final certificates are represented as a circle, while intermediate certificates are squares. The letter indicates the public key certified by the certificate. In this example, $t = 2$. E is certified by A and C; A by B and D; C by B and F; F and D are root nodes; and B is not certified.

Revocation

Revocation of a certificate can be achieved in two ways. Either the certificate is explicitly revoked, or has an expiration time and is not renewed.

The first solution requires to store a list of revoked certificates on a platform that could be queried and trusted. As for the second solution, it requires an expiration

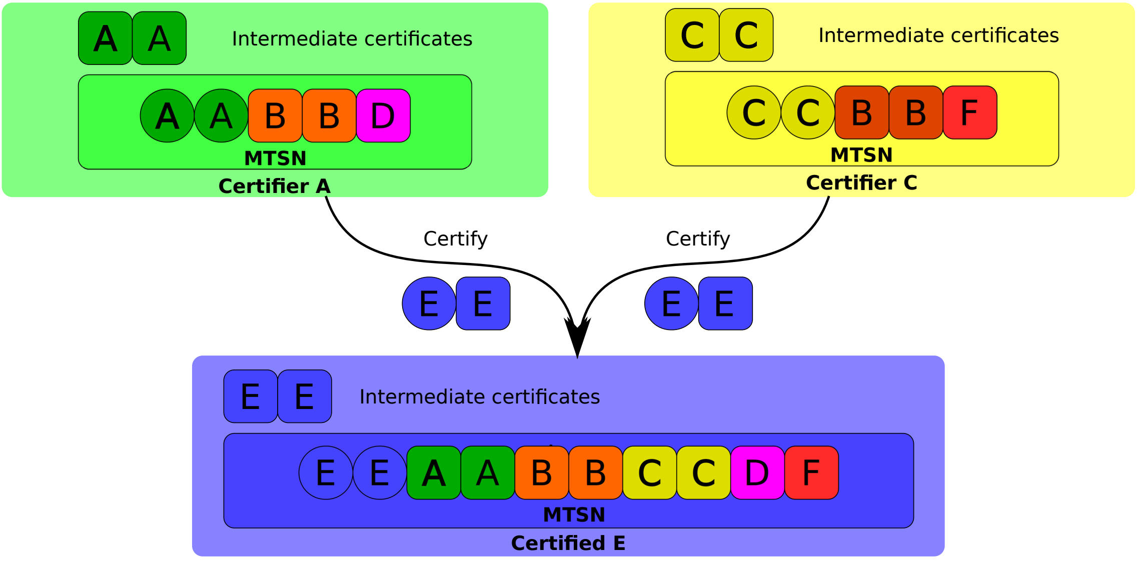


Figure 6.12: MTSN Merging

time long enough to not have to constantly regenerate certificates, and short enough to be able to quickly revoke misbehaving nodes.

We propose to use the root nodes to store a list of revoked certificates (or their hash). Thus enabling, when verifying MTSN, to query its included root nodes for the revoked certificates.

Revocation can thus be performed by contacting directly the root nodes and giving him a hash of the revoked certificate with the certificate expiration timestamp, both signed by the certifier asymmetric keys. The root node then respond by signing such query as a proof the revocation has been taken into account. Alternatively, the query and proof can be transmitted through ascendant and descendant phases, as in the certification.

This thus requires to corrupt at least t nodes in order to prevent certificate revocations. Expired certificates should be removed from the list of revoked certificates.

The certifier account then inform (in case of explicit revocation) the revoked account of the revocation, for it to be able to rebuild its MTSN and send it, if still confirmed, to the accounts it certify. Otherwise, it inform them that it is no longer confirmed. The account certified by the revoked account can then in turn update their MTSN. The whole MTSN does not need to be sent, only the new certificates, and the list of certificates to delete.

Query for identity disclosure

In case of misbehavior from a given node, its MSTN being known, the contained root nodes are contacted by a competent authority. a query for identity disclosure is then issued to reveal the misbehaving node identity.

The query is composed of the MSTN signed by the recognized competent authority, and is transmitted from the root node to the misbehaving node certifiers following the paths indicated in the MSTN.

The identity of the misbehaving node is encrypted using the authority public key, and signed by the certifier node. The respond is then transmitted to the root node following the same path, and signed by each nodes on the path.

If a node refuse to respond in a reasonable time, its certifiers must respond to the query revealing the identity of the non-responding node(s). If a node modifies the answer, invalid signature(s) would immediately reveal the position of the modification in the path, and can be reported to the authority.

Alternatively, certifier nodes can choose to reveal their identity to the authority by directly contacting it. The authority them produces a proof of answer the node can send its certifiers nodes.

In order to prove the revealed identity is the one that was certified, the certificate should contain a proof of the certified identity, only readable by the certified, or other nodes knowing a secret. For example, a hash of the certified node identity with a nonce stored by the certifier. The nonce and the hash can then be transmitted with the answer.

Biometric authentication

Certifiers ensure the account remains in the hand of its legitimate owner through biometric authentication (e.g. using PICRP). If a change in ownership is detected, certifiers can revoke their certificates. However, the more the account have certifiers, the more it is likely, that the attacker is likely to find t certifiers that continue to recognize it, thus enabling it to remain confirmed.

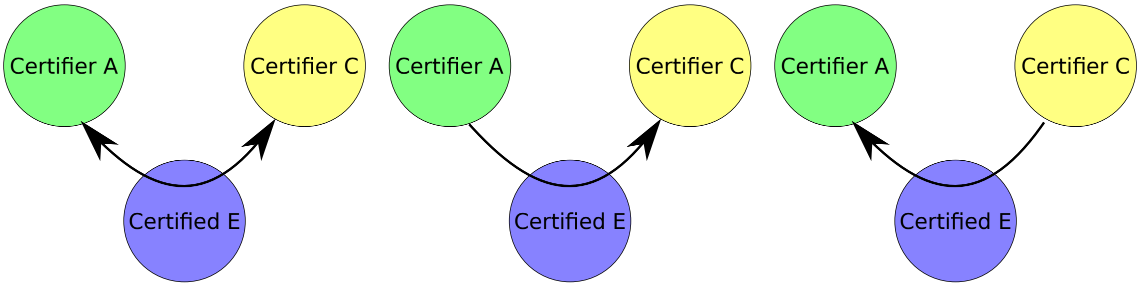
When used by its legitimate owner, we assume that the certified node has no interest into manipulating its biometric authentication. A secure communication can thus be established between the confirmed certified nodes through the certified node without revealing the certifiers node identities (e.g. with a Diffie-Hellman symmetric key negotiation).

When authenticating the user, a certifier node can then query the other certifiers for their decisions (authenticated/rejected). The hacked certified node, can only stop transmission of the query or the answer, that would be interpreted as a rejection. Misbehaving certifiers might also lie on their decision. The certifier node is thus able to take a final decision, given the other nodes decisions and the biometric authentication score.

In order to prevent dubious certifier addition and legitimate certifier removal, the user should be authenticated when adding or removing a new certifier. Only one certifier can be removed at a time.

Moreover, the certifiers should send their decisions to the other certifiers, before revealing their final decision (certificate revocation or renewal). i.e. the query sent to other certifiers should contain the decision, and answers to the query would thus be acknowledgement of the decision reception.

The whole protocol is illustrated in Figure 6.13.



(a) Secure channel (b) Query + A Decision (c) Ack + C decision

Figure 6.13: Biometric authentication

Trust graph centralized verification

Our model being assumed decentralized, centralized verification of the trust graph is not possible in practice. However this might be useful when testing and evaluating the performances of such system. This verification is thus used in our experiments and is presented in Listing 1.

This algorithm only go through the nodes certified by the lasts confirmed nodes instead of going through all nodes. This requires only $h * |C|$ nodes checks, with $h << len(n)$, the mean number a certification a node can issue (generally $h = 3$ [Barabási and Albert, 1999]). Which corresponds, in the worst case, only to $2 * nb\_edges$ nodes checks. In worst case, using 16,000,000 nodes, this algorithm takes less than 72 seconds. A limitation on the number of certification a node can easily prevents denial of service attacks.

MTSN verification

Upon queries, confirmed nodes send their MTSN in order to prove their identity. The MTSN is easily verified through a depth-first graph traversal. The validity of each certificate is verified as well as the absence of loops. Algorithm is given in Listing 2.

Certificate verification

Certificate verification is achieved by verifying the certificate signature, and that the certifier public key is certified by t different certifiers. The different fields of the certificate should also be verified, e.g. that the creation timestamp and the expiration timestamp are respectively lesser and greater than the current timestamp.

Algorithm 1 Trust graph centralized verification

```
procedure COMPUTE CONFIRMED NODES $C$
    Variable: $RN$, set of nodes;                              ▷ Set of root nodes
    Variable: $n_1, \ldots, n_{|N|}$, array of nodes;          ▷ The set of all graph nodes
    Variable: $fifo$, array of nodes;    ▷ The FIFO list of nodes to be processed
    Variable: $idx$, index;                      ▷ Index of the next node to process
    Variable: $cur$, node;                          ▷ Current node being processed
    Variable: $conf_1, \ldots, conf_{|N|}$, array of boolean;      ▷ Whether a node is confirmed or not
    Variable: $t_1, \ldots, t_{|N|}$, array of integer; ▷ The set of integers representing the computed level of trust of each node
    Procedure: $Cert(x)$                              ▷ the nodes certified by x

    $t.fill(0)$;                ▷ Initialization: trust level is 0 for all nodes
    $conf.fill(false)$;            ▷ Initialization: no nodes are confirmed
    $idx = 0$;                                     ▷ Initialization: first index

    for all nodes $n_i \in N$ do                         ▷ Search for root nodes
        if $(n_i \in RN)$ then
            $conf_i = true$;                              ▷ the node is confirmed
            $fifo.push(n_i)$;          ▷ the confirmed node has to be processed

    while ( $idx! = fifo.size()$ ) do  ▷ Iterative computation of node trust level
        $cur = fifo[idx++]$;
        for all nodes $n_j \in Cert(cur)$ do
            if ( $conf_j$ && $t_j++ \geq t$) then ▷ An unconfirmed node with enough certification has to be processed.
                $trusted_j = true$;
                $fifo.push(n_j)$
    $C = fifo$;                                    ▷ Building $C$ to be returned
```

6.5.5 Experiments

Implementation Issues

We assume that a node confirmed status verification is performed by users client while browsing. As previously said, the centralized algorithm is not adapted to such verification, as it would requires to leak all topology and biometric data, thus being a treat to confirmed users privacy. Moreover the quantity of information required to such computation make it not realistic to be downloaded by users while browsing.

Unfortunately, decentralized algorithms cannot be used for research purposes as it is unrealistic to dispose of billions of clients for tests and demonstration purposes. Thus requiring at best to simulate decentralized algorithms in an iterative way. In such cases, costs induced by communications cannot be measured, but can be

Algorithm 2 MTSN verification

```
procedure VERIFY_MTSN
    Argument: $final\_certs$, array of final certificates;
    Variable: $cur\_path$, array of certificates;           ▷ Current traversed path
    Variable: $cur\_cert$, certificate;                      ▷ Current certificate

    if ($len(final\_certs) < t$) then
        throw;
    for all certificat $f\_cert \in final\_certs$ do
        $cur\_path = [certificate]$;
        $go\_end\_of\_path(cur\_path)$;
        while ( $cur\_cert = next\_cert(cur\_path)$ ) do
            $verify\_cert(cur\_cert)$;

procedure NEXT_CERT
    Argument: $cur\_path$, array of certificates;           ▷ Current traversed path
    Variable: $cur\_cert$, certificate;
    Variable: $next\_cert$, certificate;

    $cur\_cert = cur\_path.pop()$;
    if ($!cur\_cert$) then
        Return: $cur\_cert$
    if ($next\_cert = cur\_path.front().next\_certifier(cur\_cert)$) then
        if ($cur\_path.contains(cur\_cert)$) then
            throw;
        $cur\_path.add(next\_cert)$;
        $go\_end\_of\_path(cur\_path)$;
    Return: $cur\_cert$

procedure GO_END_OF_PATH
    Argument: $cur\_path$, array of certificates;           ▷ Current traversed path
    Variable: $cur\_cert$, certificate;

    $cur\_cert = cur\_path.front()$;
    while ($cur\_cert = cur\_cert.first\_certifier()$) do
        if ($cur\_path.contains(cur\_cert)$) then
            throw;
        $cur\_path.add(cur\_cert)$;
```

estimated from the number and length of such communications.

As the decentralized algorithm is simulated on a unique computer, cryptographics operations generation and verification of signature were also simulated, as such operations tend to be pretty slows. As for the communication costs, Cryptographics costs can be estimated from the number of required signatures generation and verification. Cryptographics operation still can be easily implemented in JavaScript using the WebCrypto API [1].

Datasets

We assume that each certified node send the same biometric data to its certifiers that give it the same biometric score, and therefore take the same decision, to revoke, or not, their certification. We assume use of Fixed-Text and Free-Text Keystroke Dynamics, with a threshold corresponding to the EER of the used KDS. As we consider more users than we have in the Keystroke Dynamics datasets, we simulate the certifiers decision as a Bernoulli trial, with a probability for an attacker to be accepted, and for a legitimate user to be rejected equal to the KDS EER value. This corresponds to 24.2% for Fixed-Text and to 13.7% for Free-Text.

It is well known that the degree of social-network graphs follows a power-law distribution [Kumar et al., 2010, Cha et al., 2010, Buccafurri et al., 2013]. This can be obtained by using the Barabási–Albert model [Barabási and Albert, 1999], one of the most famous algorithms for generating random scale-free networks using *preferential attachment*. Starting from a single-node graph, each new node is connected to the existing nodes by following the law: the more the node degree is, the more the probability to receive new link is.

Instead of using such synthetic social-network graph, we use a real-life graph from the Stanford Large Network Dataset Collection (SNAP)[2], the ego-facebook dataset, with 4,039 nodes, and 88,234 edges.

Attacker model

We assume that PICRP is used in order for the certified to be authenticated by their certifiers. Attackers thus cannot impersonate users without knowing their secret, as well as having access to their computer. In such a case, we can assume that the attacker is able to obtain/use users GPS and IP location, security thus exclusively remain on Keystroke Dynamics. Such attack is quite extreme, but could be e.g. a user relative knowing its secret and having access to its client, s.a. partner or childs.

We consider 4 scenarios:

- None: when the biometric authentication is unable to distinguish attackers and legitimate users.
- Fixed-Text: security remains exclusively on Fixed-Text;
- Free-Text: security remains exclusively on Free-Text;
- Perfect: ideal case where the biometric authentication is error-free.

[1] https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API

[2] https://snap.stanford.edu/data/index.html

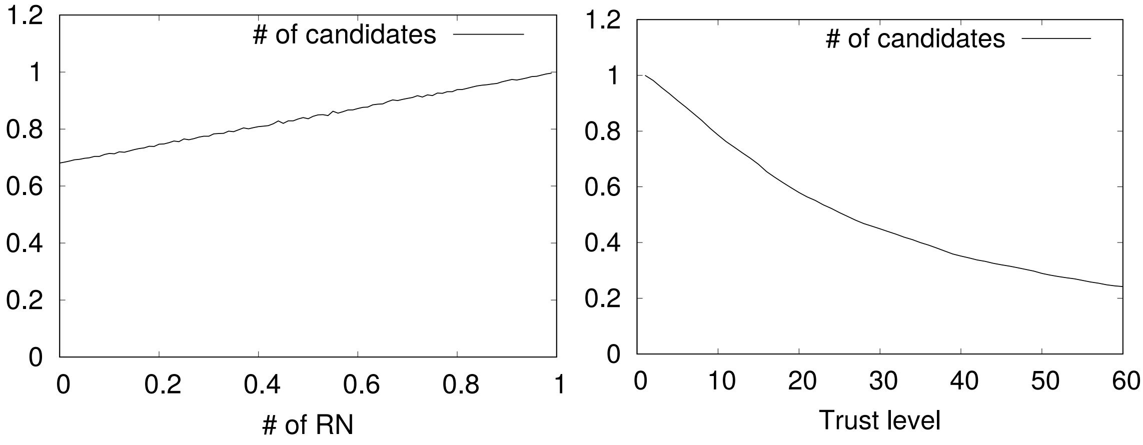
Protocol

We randomly select root nodes among nodes with degree higher than 0, assuming that isolated nodes has little to no interest into being certified by RNC. Attacker nodes are randomly selected from all nodes. The social network graph, the biometric performances, and the rate of RN and attacker constitute a context. The trust level corresponds to the system configuration.

For this experiments, we set the rate of attackers to 5%, as well as the trust level t to 15. The social network graph depends on the dataset used, and the biometric performance to the attack scenario.

Biometric authentication schemes are usually evaluated though their EER (Equal Error Rate) where the FAR (False Acceptance Rate) equals the FRR (False Rejection Rate). In our case, we will use FCR (False Confirmation Rate) and FNCR (False Non-Confirmation rate), corresponding to the rate of attackers falsely confirmed, and to the rate of users falsely not-confirmed. FNCR rate is computed among the set of candidates nodes. Candidates nodes are defined as nodes having at least t certifiers or being root nodes.

Results



(a) In function of the root nodes headcount, with the trust level $t = 15$.

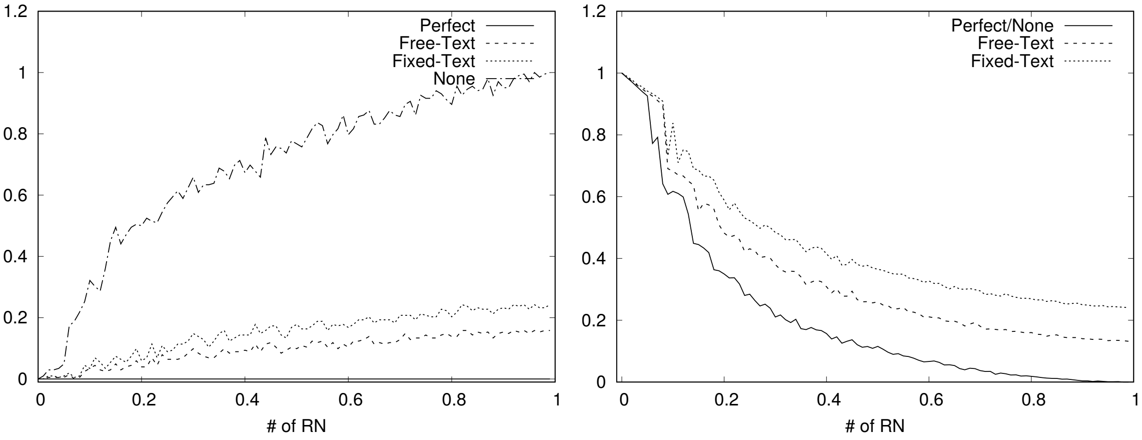
(b) In function of the trust level t, without RN.

Figure 6.14: Number of candidates in function of RN headcount, and trust level.

As shown in Figure 6.14, even with an high trust level t, the number of candidates in the graph is high. Assuming that 5% of nodes are root nodes, the rate of candidates is $\simeq 70\%$.

As shown in Figure 6.15, use of biometric data, significantly decreases the False Confirmation Rate in the most extreme scenarios (Free-Text and Fixed-Text). However, this comes at the cost of an higher False Non-Confirmed Rate.

In this experiment, we used an high The trust level ($t = 15$), a lower trust level should decrease the FNCR, specially when the RN rate is low.



(a) FCR

(b) FNCR

Figure 6.15: FCR and FNCR in function of the RN headcount, with trust level $t = 15$, and 5% of attackers.

6.5.6 Proof of concept

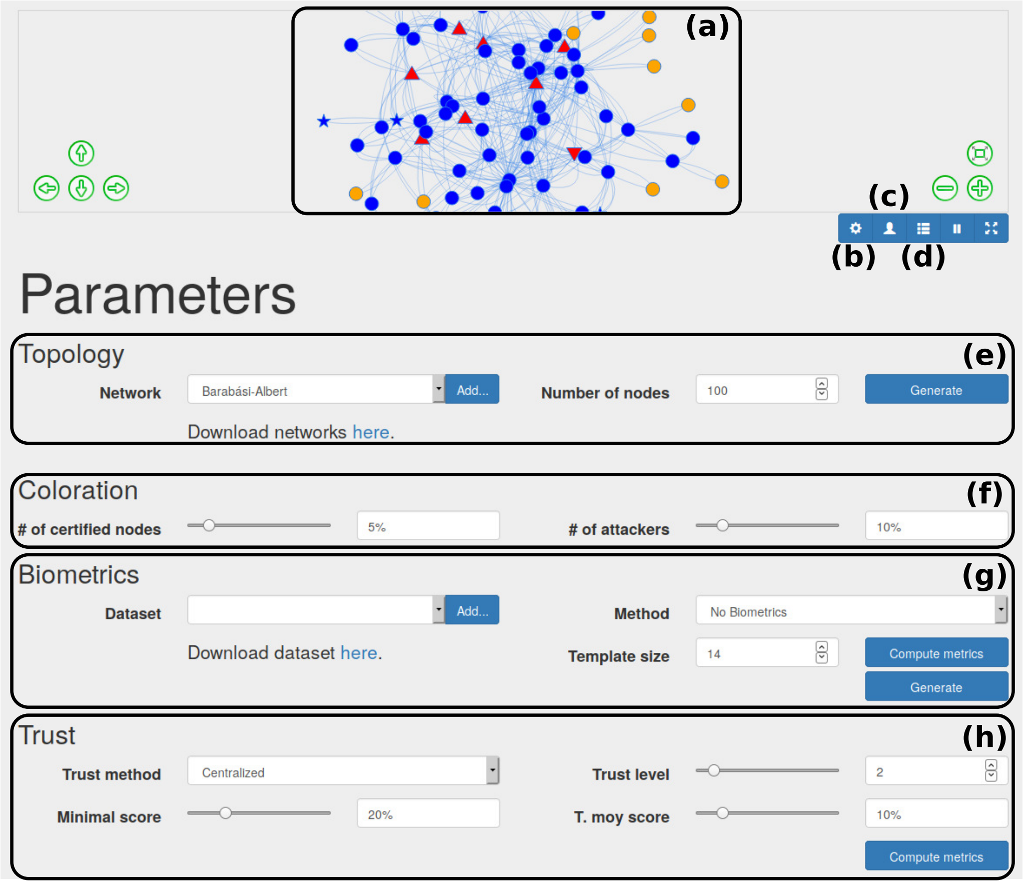
We developed a proof of concept using the two PICRP proposed in Chapter 4. The GUI is presented in Figure 6.16.

The area (a) gives a visual representation of the trust graph. Icon (b) enables to tweak parameters while icon (d) prints a list of the trust graph node. By clicking on a node in the list, or in the visual representation, a synthetic account, associated to the node, is shown. Icon (c) enables to visualize the last shown account.

In the parameters, topology (e) enables to change the trust graph topology, i.e. the number of nodes, and the trust relations. They can either be randomly generated with the Barabási-Albert algorithm, or through real-life datasets. Coloration (f) enables to select the rate of root nodes (certified nodes), as well as the rate of attackers.

Biometrics scores (g) can be generated from real-life datasets. The method gives the algorithm used to generate the Biometric scores, while the template size gives the number of references to use per users. The trust area (h) enables to tweak the trust computation. The trust level, t, indicates how many confirmed certifiers is required to become certified. Thresholds are used by the certifier to decide, given a biometric score, whether revoking or not a given certified node. Several metrics are then computed and shown below the parameters.

In short: *We propose a Social Proof of Identity enabling accountability of misbehaving users while respecting their privacy.*



Parameters

Topology

Network: Barabási-Albert

Number of nodes: 100

Download networks here.

Coloration

# of certified nodes: 5%

# of attackers: 10%

Biometrics

Dataset:

Method: No Biometrics

Download dataset here.

Template size: 14

Trust

Trust method: Centralized

Trust level: 2

Minimal score: 20%

T. moy score: 10%

Figure 6.16: Social Network demonstration interface.

6.6 Enhancing security with dedicated Hardware

Until then, it was assumed that the attacks were performed by JavaScript codes injected into web pages. However, the user computer (the client), might as well be complicit, corrupted, and/or malicious. This section reveals how to protect user privacy in such cases.

Previously, its was assumed that the trusted component, providing KDAS and PICRP features where a WebExtension, but could be integrated to the browser, a driver, or the Operating System. With correct right management, integration to a driver or the OS could requires administrator privilege to access users Keystroke Dynamics, thus preventing some malicious code from accessing it.

However, this does not protect against complicit client, or malicious code with administrator privilege. When the client is assumed untrusted, it is common, in order to increase security, to use an external or internal trusted component. Internal components s.a. Secure Element (SE), or Trusted Execution Environment (TEE) are hardware components integrated to the client that are assumed secure and trusted. Though such internal trusted components often requires a blind trust to the chips manufacturers, and generally does not provides a secure path to the keyboard.

We thus suggest the use of an external trusted device that will be plugged between the keyboard and the client, in order to protect secure KDAS and PICRP features even though the client is complicit, corrupted or/and malicious. Such external trusted device could even be integrated to the keyboard.

6.6.1 KDAS on trusted device

Assuming a trusted device between the keyboard and the client, implementing KDAS feature is trivial, as the trusted device just has to forward keyboard events with some delay. However one challenge remains to be addressed: synchronizing the trusted device with the screen frame refresh rate.

Note: *One simple proof of concept has been implemented during Benjamin Graindorge-Lamour's bachelor internship. This proof of concept is only able to receive keyboard events (from the keyboard), and to send arbitrary events (to the client). The hardware and the reception/emission of keyboard events are presented in the following sections.*

Hardware

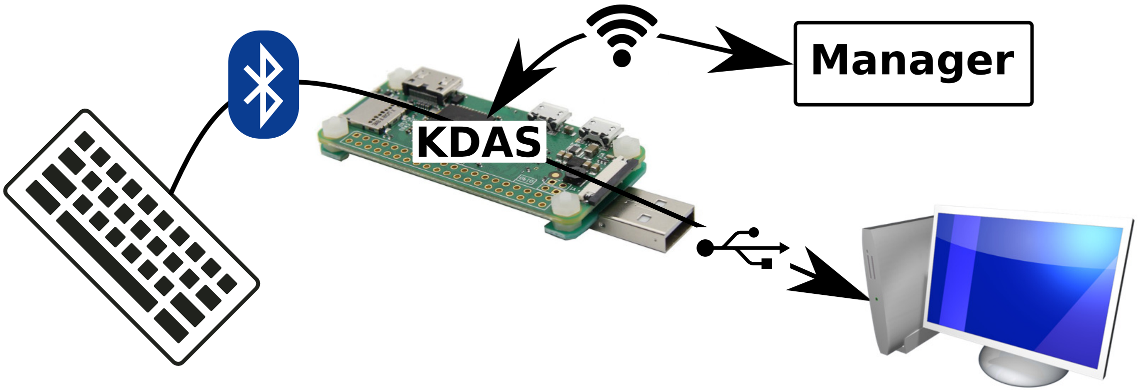


Figure 6.17: KDAS on Raspberry Pi (prototype).

A Raspberry Pi were used as a trusted device in order to intercept and emit keyboard events. As shown in Figure 6.17, a Raspberry Pi Zero W[3] is plugged to the client thanks to an USB-A addon board[4], while a keyboard is connected to the Raspberry Pi through Bluetooth. The Raspberry Pi is manageable through WiFi.

As the Raspberry Pi is already connected to the client through USB, the keyboard has to be connected to the Raspberry Pi using Bluetooth. Indeed the Raspberry Pi

[3] https://www.kubii.fr/les-cartes-raspberry-pi/1851-raspberry-pi-zero-w-kubii-3272496006997.html

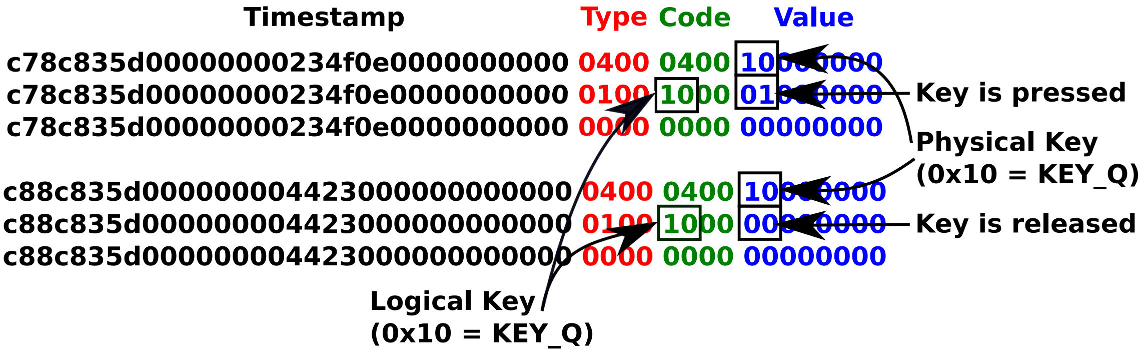
[4] https://www.kubii.fr/cartes-extension-cameras-raspberry-pi/2063-adaptateur-zerokey-usb-pour-pi-zero-kubii-3272496009271.html

cannot act, at the same time, as slave (to send keystroke to the client) and as master (to receive keystroke from the keyboard), as it only possess one USB controller. Indeed, the USB-A addon board doesn't add any USB controller, but uses Pogo pins to connect onto the circuit board, directly on the conductive tracks of the Raspberry Pi micro-USB female port. This implies that if both the keyboard (using the micro-USB port) and the client (using the USB-A addon board) were connected to the Raspberry Pi through USB, the keystroke would be directly forwarded from the keyboard to the client without any means of intercepting it. Bluetooth would not be required if the device would have at least two USB controllers. WiFi is also not required for the device management, and could be done, e.g. through the keyboard.

Although the used hardware is high priced 21€40 (≃ $23.69), production of a dedicated hardware (or integration to keyboards) could be expected to be way cheaper.

Receiving Keyboard events

The received keyboard events are read on the Raspberry Pi thanks to `/dev/input/eventX`[5]. An example of a keystroke on a 64-bits OS is shown in Figure 6.18.



```
c78c835d00000000234f0e0000000000 0400 0400 10000000
c78c835d00000000234f0e0000000000 0100 1000 01000000
c78c835d00000000234f0e0000000000 0000 0000 00000000

c88c835d000000004423000000000000 0400 0400 10000000
c88c835d000000004423000000000000 0100 1000 00000000
c88c835d000000004423000000000000 0000 0000 00000000
```

Figure 6.18: Description of keyboard input events.

Input events are structured as follow:

- **Timestamp** (8 bytes on 32-bits OS, 16 bytes on 64-bits OS);
- **Type** (2 bytes);
- **Code** (2 bytes);
- **Value** (4 bytes).

Each Keyboard event generates the following input events:

- **EV_MSC (Type 0x04) MSC_SCAN (Code 0x04)** is used (here) to describe a keyboard physical key.
  - **Value**: the physical key released, pressed, or repeated.
- **EV_KEY (Type 0x01)** is used to describes a keyboard event.
  - **Code**: the logical key released, pressed, or repeated.
  - **Value**: 0 for release, 1 for keypress, and 2 for autorepeat.
- **EV_SYN (Type 0x00)** is used to separate events and does not contains any information.

[5] `https://www.kernel.org/doc/Documentation/input/input.txt`

Input event values are described in `input-event-codes.h`[6].

Sending Keyboard events

Emission of keyboard events from the Rapsberry Pi to the client is performed by writing reports into `/dev/hidgX`[7]. A report is an 8-bytes structure:
0b00000000 00 00 00 00 00 00 00, with:

- **Modifiers** (1-byte), describes the state of modifiers keys (s.a. the control key);
- **Reserved** (1-byte), always 0x00;
- **Keys pressed** (6x 1-byte), describes up to 6 pressed keys.

The keys pressed field describes 0 to 6 key pressed, each byte contains the code of a pressed key, or otherwise 0x00. Meaning that only up to 6 keys can be pressed at the same time. All keys that does not appears in a report are assumed released. Thus, after each press events, a report without any pressed key is written in order to release the pressed key.

6.6.2 PICRP on trusted device

Assuming that a KDAS is performed on a trusted external device, the PICRP obviously cannot be computed on the client. Indeed, it would be absurd if the Keystroke Dynamics were protected with a KDAS, only to be revealed at each PICRP computations on the client.

Moreover, PICRP cannot be computed at the sole request of the client, and must requires the user consent. Indeed, the client could asks for PICRP whenever it wants, without the user knowledge and consent, in order to continuously identify him.

KDAS will be assumed to be applied by the trusted device on the keystroke events it receives, before transmitting it to the client. Indeed, computing a PICRP on the trusted device while exposing the user keystroke dynamics would have little to not interest.

Hardware

We assume that the trusted device does not have any button, has only two communication channels (one from the keyboard, and one to the client), and behaves as a keyboard. Bluetooth could be used to control the trusted device, and to connect wireless keyboards, however it would raise the price of such device. We thus assume that the trusted device does not have any wireless capabilities.

As shown in Figure 6.19, communications are assumed unidirectional. The user types on the keyboard, the keyboard sends keyboard events to the trusted device, the trusted device delays and forwards them to the client, and the client prints them on its screen, to be viewed by the user.

As the client is assumed untrusted, and is able to print arbitrary content on the screen, the user cannot trusts what he sees on the screen. In order to ensure a secure

[6] https://github.com/torvalds/linux/blob/master/include/uapi/linux/input-event-codes.h

[7] https://www.kernel.org/doc/Documentation/usb/gadget_hid.txt

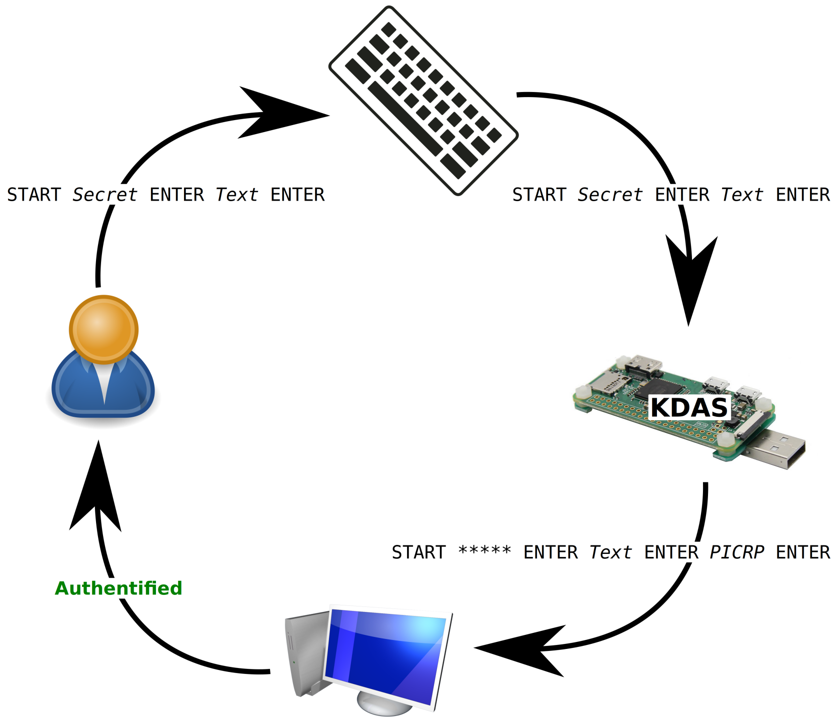

START *Secret* ENTER *Text* ENTER
START *Secret* ENTER *Text* ENTER
KDAS
START ***** ENTER *Text* ENTER *PICRP* ENTER
Authentified

Figure 6.19: PICRP on Raspberry Pi.

and trusted information printing, a screen could be added to the trusted device. E.g. for the prototype, an e-Ink screen could be used. This costs near 10€43 (≃ \$11.53) for a 200x200 pixels screen[8] or a 250x122 pixels screen[9], raising the prototype price to 31€83 (≃ \$35.22).

In such case, the user has to be notified of any new printed message on the trusted device, for him to read it. The notification could be displayed on the client (e.g. on the visited web page), or be a sound played by the trusted device (s.a. a bip), requiring an additional component, either a speaker or a buzzer. The keyboard event could also be blocked until the user has acknowledged the notification.

In the following, it will be assumed that information are printed on the client, and therefore untrusted.

Control messages

In order to give consent, users have thus to type additional content on the keyboard, generating keyboard events that will need to be interpreted by the trusted device and the client. We call this *control messages*.

For Same/Fixed-text we propose the following format:
`START` *`Secret`* `ENTER` *`Text`* `ENTER [`*`PICRP`* `ENTER]`, with:

[8] https://thepihut.com/collections/raspberry-pi-screens/products/eink-screen-1-54-200x200-display-only

[9] https://thepihut.com/collections/raspberry-pi-screens/products/eink-screen-2-13-250x122-display-only

- START, a combination of keys (e.g. Ctrl+P) to start the control message.
- *Secret* and *Text*, texts typed by the user;
- ENTER, a key or combination of keys (e.g. Enter);
- *PICRP*, the computed PICRP encoded in base 64.

The control message is transmitted to the client as the user type it, with a KDAS protection. The typed characters for Secret are replaced by, e.g., '*' when transmitting it to the client. After the second ENTER, the trusted device send the PICRP encoded in base 64 to the client. At any moment the user can CANCEL the control message using e.g. the backspace key. Please note that *Secret* or *Text* might be empty, therefore the PICRP should be computed without the keystroke dynamics modality associated to the missing element.

Even though the client is untrusted, it can be used to improve usability by printing information to the user. For example, to highlight fields where a PICRP is required, or to show the status of the current control message.

Free-text are a little more complex to handle as several Free-Text authentication can be performed simultaneously. It requires 4 to 5 different control messages:

- START_FREE *name* ENTER: starts the Free-Text authentication named *name*.
- STOP_FREE *name* ENTER: stops the Free-Text authentication named *name*.
- STOP_ALL_FREE: stops all Free-Text authentications.
- NEW_PICRP *name* ENTER *PICRP* ENTER: sent regularly by the trusted device to the client for each running Free-Text authentications.
- LIST: for the trusted device screen, show the running Free-Text authentications.

The client could be used to keep track of the running Free-Text authentications, but using the trusted device screen would be more secure, as the client might lie.

A simple JavaScript code, embedded either in a web page or in a WebExtension, cannot prevent the client from printing the PICRP where it does not belong, e.g. outside the web page, or outside the browser. A native application (i.e. running outside the browser) must thus either to collect and dispatch PICRP, using their name, or to explicitly asks the trusted device to produce PICRP (using then a bidirectional communication with the trusted device). The native application could then communicate with a WebExtension using *native messaging*[10].

In case of bidirectional communication with the trusted device, the client could send additional modalities to use to compute the PICRP. Please remind that as the client is untrusted, all information received from the client, s.a. other modalities, is therefore untrusted and should be verified with care on the trusted device.

In short: *Security can be increased using external trusted devices, however this requires a cost that users may not be willing to pay. Solutions must thus be cheap, ergonomics, and optional.*

[10] https://developer.mozilla.org/fr/docs/Mozilla/Add-ons/WebExtensions/Native_messaging

Chapter 7

Conclusion and future works

This short chapter gathers the contributions of this PhD thesis and associated research perspectives that could be exploited to pursue this work.

Contents

7.1 What do websites know about you?

This chapter demonstrated that Keystroke Dynamics constitutes a threat to users privacy. The threat was quantitatively estimated through a fair KDS comparison, taking into account the influence of the KDS context and configuration.

In our attacker model, attackers collect each user Keystroke Dynamics, to then be able to recognize their future typing. This model could be completed by enabling attackers to recognize users past typing. The attacker could also collect, for each users, non-consecutive Keystroke Dynamics, and then try to recognize users on their past, concurrent, and future typing.

We tested KDS with theoretical user-dependent thresholds without any references updates, realistic methods s.a. in [Mhenni et al., 2019] could be tested and compared to our theoretical results. References freshness or/and quality could also be exploited by template update and users-dependent thresholds mechanisms. For identification, references-dependent thresholds on rank and/or distance scores could also be tested.

KDS configuration could also be improved by using other distances, e.g. based on the merging of several distances. Better pre-processing would improve performances, either through better laws estimations/fittings, or by estimating laws followed by unknown digraphs. Different weights could also be applied on durations or digraphs, either before the distance computation, or by computing a weighted mean from distances computed for each durations/digraphs. In the same way, different weights could be applied when merging references or scores, e.g. based on the reference freshness or quality. Deep learning could also be exploited, e.g. for Soft-Biometrics, or to convert Keystroke Dynamics into vectors of optimal features for a given distance.

Other modalities could also be explored in depth, s.a. mouse dynamics, the browser fingerprinting, or recognition of sound (microphone) and video (webcam), not only for identification/authentication, but also for Soft Biometrics. For example, using browser fingerprinting to assert users age or gender. Larger datasets should also be constituted and tested, especially for Free-Text authentication/identification, and Free-Text Soft Biometrics.

7.2 How to protect my information from malicious websites

This chapter proposed several techniques enabling to protect users against unwanted Keystroke Dynamics collection Our findings showed a trade-off between privacy and usability (latency).

KDAS were not fairly compared as only the best configurations and context were tested without any guarantee that they remain the best after protection. All configuration and context should be tested for a given KDAS and expected latency. Moreover, users shared the same threshold. Theoretical user-dependent thresholds should be used to estimate a lower boundary to attackers capabilities.

Usability has been evaluated in terms of seconds, but should also be evaluated in terms of users acceptations, i.e. at which point the latency is noticed, bothering, or unaccepted by users. Website compatibility with such techniques should also be tested, especially due to the lost of events trusted status.

Privacy preservation across time should also be studied, especially on long Free-Text. Indeed, non-blocking KDAS sill expose users typing speed. Anonymisation of mouse dynamics could also be considered. Contrary to keyboard events, mouse events not only can be delayed, but can also be suppressed, modified, on created.

7.3 Using personal data in a privacy compliant scheme

This chapter described our Personal Identity Code Respecting Privacy (PICRP) enabling the comparison of several arbitrary biometric information without revealing them.

In this chapter, we presented a user authentication protocol based on the indirect comparison of PICRP. Other avenues should be studied, s.a. the use of 0-Knowledge protocols to verify PICRP, the creation of One-Time Passwords from PICRP, or the signature of arbitrary contents using the PICRP as a private key.

Our PICRP used Fixed-Text Keystroke Dynamics, GPS location and IP location. Free-Text and some Browser Fingerprinting information Keystroke Dynamics were presented and pre-processed, but were not included in the final PICRP. Other biometric modalities could also be tested, s.a. the mouse dynamics, soft biometrics information (s.a. age, gender), or the user travel behavior. These personal data are merged using weights set empirically through an exhaustive search. Better strategies for searching the optimal weights should also be proposed. Other merging strategies could also be proposed (e.g. mixing pre and post merges).

PICRP is based on BioHashing, a keyed Local Preverving Hashing (LPH). Other key-based LPH could be tested, s.a. GreycHashing. However, such protection introduces a cost in performances. Better pre-processing solutions could help to solve this issue, e.g. by using deep-learning to produce optimal feature-vectors.

As always, a better understanding of the Keystroke Dynamics model could help to improve performances, e.g. by improving KD laws estimations or being able to estimate laws followed by unknown digraphs. For Free-Text, only one configuration were tested, other configurations should be considered, and might offers better performances. XYZ locations could be also uniformized in such a way that each triplet as an equal probability to appear.

Finally, due to the failure of initial collects, we opted for synthetic and chimeric datasets. Our proposed PICRP should be tested using real-life data, specially for the GPS and IP location. However, in such context, a single reference might not be enough, thus requiring the use of template update strategies.

7.4 How to use this data for research while respecting privacy?

This chapter modeled Same-Text Keystroke Dynamics thus enabling dataset augmentation, or user usurpation through syntheticly generated data.

We assumed that user KD model does not evolve through time, which is obviously not the case in real-life. We speculate that the observed trade-off between ERS and EEE might be due to this assumption. Futur models should take into account the user KD evolution. Evaluation of the generated synthetic data should be done in a faster way without depending on a particular context and configuration. In a same way, the capacity to estimate laws parameters should not influence the model performances, and should be evaluated separately. Moreover, discrimination of real and synthetic data should be tested, e.g. using deep learning. Estimation of unknown digraphs parameters on Free/Fixed-text could also be considered, as well as real-time usurpation.

7.5 Some examples of applications of these findings

This chapter explored applications of the proposed contributions, as well as security enhancement through trusted devices. Implementation of the proposed applications should be pursued and tested with real-life data.

We presented the issue of multi-account detection. GreycHashing could be considered to tackle this issue and its security in such context should be evaluated.

We proposed to use the PICRP to build proof of authorship of an online co-written document. However, the pertinence of each author contribution should be evaluated, and not judged only on the number of keyboard events. Synthetic data should as well be detected.

We tackled the issue of Trust On First Use (TOFU) with a Social Proof of Identity. Our experiments assumed an automatic decision based on biometrics, with no evolution of the trust graph through time. Future experiments should aim at reproducing real-life behaviors, to demonstrate if the produced indicators helps manual users decisions. Disclosed information should also be reduced, and the security of the biometric verification improved.

Trusted devices were then considered in order to bring security if the client is assumed corrupted or malicious. The usability of each approach should be determined through experiments. Prices of dedicated hardware should also be considered. The device plugged on the client, and viewed as a Keyboard. However, proposed KDAS approaches require a synchronisation of the device with the client screen refresh rate. This issue should be considered in future works.

French synthesis

This short chapter provides a synthesis of this thesis in French.

Contents

7.6 Position de la thèse

7.6.1 L'environnement: Internet et ses acteurs

Internet n'est pas seulement un ensemble de techniques et d'ordinateurs, il est aussi constitué d'acteurs ayant divers intérêts et motivations. Les solutions techniques déployées ne sont pas une fin en soit, mais sont au service des acteurs et de leurs besoins. La compréhension de cet environnement, des intérêts des acteurs, leurs besoins et motivations, aussi bien que les conflits émergeant des divergences d'intérêts entre acteurs, sont ainsi requises à la conception de solutions techniques adaptées.

Dans la suite, nous présentons ainsi rapidement les acteurs d'Internet et explorons leurs buts et motivations. Dans la section suivante, nous discutons les problématiques inhérentes à Internet, pour ensuite établir les objectifs de cette thèse. Cependant, la présentation des acteurs reste superficiel, la compréhension des tenants et aboutissants des interactions d'acteurs requièrent de solides connaissances interdisciplinaire, principalement dans le domaine de l'économie (Théorie des jeux), la psychologie, la sociologie, ainsi qu'en sciences juridiques.

Les acteurs d'Internet

Internet est défini par le dictionnaire de Cambridge comme un *"système mondial de réseau d'ordinateurs utilisés pour échanger de l'information"*, soulignant les 3 composants principaux d'Internet :

- ***ordinateurs***, constituant un réseau physique (de réseaux) ;
- ***utilisateurs***, personnes physiques ou juridiques interagissant sur Internet ;
- ***information***, qui sont échangées sur Internet.

Les ordinateurs n'agissent pas de leur propre initiative, mais au nom d'utilisateurs. Ce sont des outils instrumenté par les utilisateurs afin de poursuivre la réalisation de leurs besoins. Les ordinateur n'étant pas des personnalités juridiques, ils ne peuvent être tenus responsables de leurs actions. En revanche, les utilisateurs sont responsables des actions exécutées en leur nom par les ordinateurs.

Nous distinguons les utilisateurs (*Service Provider*) instrumentant des ordinateurs (*server*) à fin de fournir des services, des utilisateurs (*users*) accédant ces services via d'autres ordinateurs (*clients*). Ces rôles sont non-exclusifs, e.g. un ordinateur peut être, en fonction du contexte, soit un client, un serveur, ou les deux.

Comme illustré par la Figure 7.1, nous considérons les rôles non-exclusifs joués par les acteurs d'Internet suivants :

- ***Auteur***, source/créateur de l'information ;
- ***Sujet***, ce dont l'information parle ;
- ***Lecteur***, consomme l'information ;
- ***Fournisseur de Service***, héberge et distribue l'information ;
- ***Modérateurs***, régule l'information en appliquant des politiques ;
- ***Société***, définie les politiques qui devraient être appliquées.

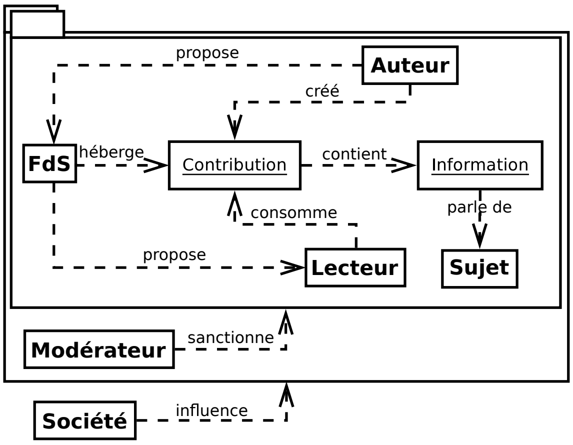


Figure 7.1: Acteurs d'Internet

Les auteurs créent des contributions contenant de l'information à propos d'un sujet. Ils proposent leurs contributions aux Fournisseurs de Services (FdS), qui, en retour, les propose aux lecteurs pour leur consommation. Ces interactions sont régulées par les modérateurs dans un cadre influencé et défini par la société. Ces rôles sont présentés et détaillés dans la suite.

En bref : *Les acteurs d'Internet jouent des rôles non-exclusifs : ils peuvent être auteurs/lecteurs/sujets d'informations, un fournisseur de service, un modérateur, ou/et être simplement parti prenante de la société.*

Sujets

Le sujet est ce dont l'information échangée parle. Son but est de contrôler son image et sa réputation, i.e. comment les lecteurs les perçoivent suite à la consommation d'informations. Pour une personne physique ou juridique, cette image constitue leur identité virtuelle.

Les sujets visent des interactions de différentes natures en fonction du contexte. Pour chaque ils maintiennent une image et identité différente [al, 2014, Boyd, 2014]. Par exemple, dans un contexte professionnel, un sujet aura tendance à viser une image plus sérieuse et professionnelle que dans un contexte familiale. Afin de maintenir leur image, et ainsi la manière dont ils interagissent avec les autres, les sujets visent, dans une certaine mesure, à contrôler les informations les concernant, i.e. qui a accès à quoi, indépendamment de la véracité de l'information.

Tristan Nitot[1] identifie dans [Nitot, 2016, p.32-37] cinq causes des fuites ou des mauvais usage de l'information :

[1] Foundateur de Mozilla Europe, membre (2013-2015) du *Conseil National du Numérique*, une institution étatique consultative française, et membre du comité de la prospective de la CNIL depuis 2015 (`https://www.cnil.fr/fr/les-membres-du-comite-de-la-prospective`)

- Une entité peut volontairement dénoncer un utilisateur ;
- Un employé peut outrepasser ses fonctions et devoirs ;
- Un ordinateur/serveur peut être hacké ;
- Une entité (e.g. une agence étatique comme la NSA) peut espionner un individu ;
- L'auteur, lui-même, peut malencontreusement dévoiler une information.

Une sixième raison peut être ajoutée : une entité peut vendre, ou partager, l'information à un autre en vue d'un gain financier [Fox et al., 2000].

Bien qu'en soit l'information, ou la connaissance de l'information ne constitue pas une menace de quelque manière que ce soit, leur utilisation et conséquences peuvent l'être. L'information non-contrôlée constitue ainsi une menace sérieuse pour les sujets :

- La détérioration de l'*image* d'un sujet impact la manière dont il interagit avec les autres, résultant, e.g. à une perte de confiance, de crédibilité, ou d'influence (e.g. boycott d'une entreprise, voire même d'une personne physique).
- Les sujets peuvent *perdre du pouvoir*, e.g. du pouvoir de négociation (e.g. pour des frais bancaires ou d'assurances).
- Un tiers peut *gagner du pouvoir sur* les sujets, e.g. chantage, extorsion, ingénierie sociale.
- Les sujets peuvent être *attaqués*, s.a. vols, usurpation d'identités, enlèvements.
- Les sujets peuvent être *sanctionnés* par les modérateurs, s.a. banni du service, privé d'Internet pour leurs responsables légaux, recevoir une amande, mis en prison, ou renvoyé d'un emploi [Fox et al., 2000].
- Les sujets peuvent être *sanctionnés* par la société, e.g. calomnies, harcèlement, menaces physiques.
- La *peur* des conséquences listées ci-dessus peuvent conduire le sujet à l'auto-censure, ou, si une information sensible a été dévoilée, au suicide.

Cependant, le contrôle de l'information sur Internet est difficile. Premièrement, il est difficile pour un sujet d'être conscient de chaque informations dévoilées, ainsi que de tous les tenants et aboutissants de telles divulgations, à la fois à court et long termes. De plus, ses identités ont de fortes chances de se recouper et ainsi être liées par des entités tierces, i.e. déterminer qu'elles appartiennent bien à la même personne. La possibilité de lier des identités peut conduire à des "collisions" de contextes incompatibles [Boyd, 2014], i.e. des contextes où le sujet maintient des images contradictoires.

Deuxièmement, l'information peut être facilement trouvée sur Internet via l'utilisation de moteur de recherches standards, plus avancés/spécifiques. De tels outils permettent à un tiers de retrouver une aiguille dans la botte de foin immense qu'est Internet. Ainsi, l'information publiquement accessible sur Internet doit être assumé connu par tous.

Enfin, l'information peut difficilement être totalement retirée d'Internet car pouvant facilement être copié et partagé. Toute tentative de forcer la suppression d'une information est souvent contre-productive à cause de l'effet Streisand [Jansen and Martin, 2015]. L'information peut aussi être archivée par le FdS lui-même, quand bien même l'information aurait été officiellement "supprimée", ou par des sites spécialisés, s.a. `https://archive.org/web/`. Ainsi, toute information connue à un temps donné, le sera probablement par la suite. Dans [Nitot, 2016], Tristan Nitot recommande l'approche POSSE[2] pour augmenter le contrôle des sujets sur l'information qu'ils dévoilent. Cependant, cela n'empêche pas un tiers de copier une telle information sur d'autres sites.

À la lumière de ce qui a été dit précédemment, le contrôle de l'information par les sujets repose principalement sur leur choix de dévoiler, ou non, eux-même l'information.

En bref : *Les sujets ont besoin de contrôler, dans une certaine mesure, l'information les concernant, ces dernières pouvant constituer de sérieuses menaces à leur encontre, s.a. chantages, boycott, usurpation d'identité, harcèlements. Cependant, ce contrôle est limité, donné que l'information peut être facilement trouvée sur Internet, et peut très difficilement en être retiré.*

Auteurs

Les auteurs sont une personnalité physique ou juridique créant de l'information afin qu'elle soit consommée par un ensemble de lecteurs donné, pour un usage donné.

Cependant, ils peuvent difficilement s'assurer que l'information créée sera consommé de la manière qu'ils souhaitent, par les lecteurs qu'ils visaient. En effet, une fois l'information connue, les lecteurs et les fournisseurs de services sont capables de la partager, ou d'effectuer des calculs arbitraires sur leurs clients ou serveurs, sans que les auteurs en aient connaissance ou y ai consenti. De plus, toute création d'information contient aussi des informations rebuts, s.a. le pseudonyme de l'auteur, un timestamp, une adresse IP, un champ lexical utilisé. Certaines de ces informations concernent l'auteur, qui est ainsi à la fois auteur et sujet. Ainsi, l'auteur a besoin, avant chaque création d'information, d'arbitrer entre ses intérêts en tant qu'auteur et en tant que sujet.

Comme nous l'avons vu dans la section précédente, il est très difficile de comprendre tous les tenants et aboutissants des informations qu'on dévoile. De plus, il est difficile pour un auteur de connaître avec précision les informations qu'ils dévoilent en contribuant. En effet, quand bien même les fournisseurs de services fournissent des Conditions Générales d'Utilisations (CGU), les auteurs ne sont pas tous au fait des tenants et aboutissants du service offert, et les CGU peuvent ne pas tout à fait correspondre à la réalité des collectes et traitements effectués par le fournisseur de service.

[2] Publish on your Own Site Syndicate Elsewhere - `https://indieweb.org/POSSE`

En bref : *Les auteurs produisent de l'information afin de le faire consommer par un ensemble de lecteurs donnés. Cependant, comme vu dans la section précédente, l'information peut difficilement être contrôlée. Aussi les auteurs son les sujets de l'information rebut sans nécessairement en avoir pleinement conscience.*

Lecteurs

Les lecteurs sont une personnalité physique ou juridique visant à accéder et consommer l'information qu'ils souhaitent, de la manière qu'ils souhaitent, potentiellement afin de, ou pour effet de, prendre une action, prendre une décision, ou se forger leur opinion sur un sujet donné.

Cependant, les lecteurs sont physiquement incapables de consommer toute l'information créée. En effet, en 2014, 300 heures de vidéos *par minutes* ont été téléchargées sur Youtube[3], et 220 330 livres ont été publiés par les membres de l'IPA[4], i.e. un livre toutes les 6 minutes. En 2006, près de 1 350 000 articles scientifiques ont été publiés d'après [Bjork et al., 2009], i.e. un article toutes les 24 secondes. Les lecteurs ont donc besoin de sélectionner l'information qu'ils veulent, ou non, consommer.

Les lecteurs sélectionnent l'information avec leur propres critères, e.g. la qualité perçue, la véracité, les sujets, la réputation. Selon le contexte certaines informations peuvent être indésirables pour les lecteurs, s.a. spam, porn, hors sujets, publicités. Cependant, il peut être difficile pour les lecteurs de trouver ce qu'ils veulent consommer [Cordier, 2015], et d'évaluer la qualité de l'information et sa pertinence avant de la consommer. Dans un sens, certaines information peuvent tromper les lecteurs afin d'être consommées.

La qualité, véracité, et la confiance que les lecteurs placent dans une information est nécessaire à la correcte consommation de l'information. Si une information est incomplète, déformée, ou fausse, les lecteurs peuvent être amenés à prendre des décisions non-éclairées, pouvant conduire à des conséquences tragiques, s.a. ruine financière, absence de soins médicaux, abus de confiance.

En bref : *Les lecteurs visent à accéder et consommer l'information. Avec la multitude d'informations continue sur Internet, les lecteurs doivent sélectionner celles qu'ils souhaitent consommer, au risque d'être déçus ou trompés.*

[3] `https://www.cnet.com/news/youtube-music-key-googles-stab-at-taking-paid-streaming-songs-mainstream/`

[4] `http://www.internationalpublishers.org/images/annual-reports/ipa_ar_online.pdf`, page 17

Fournisseurs de Services

Les fournisseurs de services servent d'intermédiaires entre les auteurs et lecteurs et distribuent l'information. Leur objectif est souvent d'ordre financier, e.g. générer un profit en distribuant de l'information, mais peut aussi être idéologique, et centré autour de leurs propres valeurs.

La problématique des fournisseurs de services est décider la manière dont les lecteurs et auteurs doivent interagir, en particulier, de quelles informations devraient être mises en avant et/ou distribuées à qui. Les fournisseurs de services doivent ainsi construire une politique interne qui sera traduite, e.g. en des conditions générales d'utilisations, des processus sous-jacents (s.a. censure automatique de certains types de mots, mise en avant des informations populaires), la structure du service et ses algorithmes, résultants en un cadre pour les interactions auteurs/lecteurs.

La construction d'un tel cadre est loin d'être trivial. Évidemment, pour être utilisé, le service doit remplir les besoins de ses utilisateurs, i.e. les auteurs et lecteurs. De plus, le fournisseur de service peut être, en fonction de la législation, partiellement responsable de l'information qu'ils hébergent et distribuent. De telles contraintes légales sont d'autant plus importantes que le fournisseur de service est souvent connu, sans la possibilité d'être entièrement anonyme, contrairement aux lecteurs ou auteurs qui sont généralement au moins pseudonymes sur Internet. Les fournisseurs de services peuvent ainsi moins facilement échapper aux sanctions légales.

Un tel cadre doit aussi correspondre à l'éthique de ses utilisateurs et à la morale de la société afin de ne pas être désapprouvé et condamné, e.g. via un boycott. L'information hébergée et partagée par le fournisseur de service va influencer son image, e.g. par association, ou en pensant que le FdS approuve les informations hébergées. Le tout, bien évidemment, sans perdre en vue leurs meilleurs intérêts.

En bref : *Les fournisseurs de services distribuent l'information des auteurs aux lecteurs pour atteindre leurs intérêts et besoins. Cependant, le cadre d'interaction construit par le fournisseur de service doit aussi correspondre aux voeux de ses utilisateurs, ainsi que ceux des modérateurs et de la société.*

Société

La société définie quelles consommations are légales, "morales", ou "éthiques". Cependant, plusieurs points de vues opposés peuvent être exprimés sur quelques questions morales ou éthiques. Les réponses sont souvent pondérées et nuancées en fonction de l'exact contexte de la situation [Allen, 1996]. Si les règles ou loi ne sont pas reconnues comme légitimes ou justes, elles risquent de se faire transgresser ou de ne pas être appliquées.

Plusieurs lois régulent les interactions en ligne, nous nous concentrons dans cette section aux lois européennes sur la vie privée. En Europe, le droit à la vie privée est déclarée par :

- l'article 12 de la Déclaration Universelle des Droits de l'Homme (1948): *"Nul ne sera l'objet d'immixtions arbitraires dans sa vie privée, sa famille, son domicile ou sa correspondance, ni d'atteintes à son honneur et à sa réputation. Toute personne a droit à la protection de la loi contre de telles immixtions ou de telles atteintes."*[5]
- l'article 8 de la Convention Européenne des Droits de l'Homme (1953) : *"1. Toute personne a droit au respect de sa vie privée et familiale, de son domicile et de sa correspondance.*
  2. Il ne peut y avoir ingérence d'une autorité publique dans l'exercice de ce droit que pour autant que cette ingérence est prévue par la loi et qu'elle constitue une mesure qui, dans une société démocratique, est nécessaire à la sécurité nationale, à la sûreté publique, au bien-être économique du pays, à la défense de l'ordre et à la prévention des infractions pénales, à la protection de la santé ou de la morale, ou à la protection des droits et libertés d'autrui."[6]
- les articles 7 et 8 de la Charter of Fundamental Rights of the European Union (2000) :
  Article 7. Respect de la vie privée et familiale. Toute personne a droit au respect de sa vie privée et familiale, de son domicile et de ses communications. Article 8. Protection des données à caractère personnel. 1.Toute personne a droit à la protection des données à caractère personnel la concernant. 2. Ces données doivent être traitées loyalement, à des fins déterminées et sur la base du consentement de la personne concernée ou en vertu d'un autre fondement légitime prévu par la loi. Toute personne a le droit d'accéder aux données collectées la concernant et d'en obtenir la rectification. 3. Le respect de ces règles est soumis au contrôle d'une autorité indépendante.[7]

La vie privée a été une problématique relativement tôt dans l'histoire des ordinateurs et d'Internet, et est toujours d'actualité de nos jours. Les lois sur la vie privée on précédemment été définies, en France, par la "Loi informatique et liberté" (1978), modifiée par décrets en 1991 et en 2004. Cette loi a inspirée la *Convention pour la protection des personnes à l'égard du traitement des données à caractère personnel*[8] (1981), ainsi que la *Directive 95/46/CE sur la protection des données personnelles*[9] (1995). Les lois sur la vie privée sont maintenant définies au niveau Européen par le *Règlement Général sur la Protection des Données (* (RGPD) qui est entré en vigueur le 28 Mai 2018, durant cette thèse.

Le RGPD requiert que les FdS obtiennent le consentement explicite et positif des utilisateurs avant de traiter leur données personnelles. En conséquences, les options de retraits (opt-out) ne peuvent désormais plus être utilisées pour obtenir le consentement des utilisateurs. Les Conditions Générales d'Utilisations (CGU)

[5] https://www.un.org/fr/universal-declaration-human-rights/index.html

[6] https://www.echr.coe.int/Pages/home.aspx?p=basictexts&c=fre

[7] https://www.europarl.europa.eu/charter/pdf/text_fr.pdf

[8] https://www.coe.int/fr/web/data-protection/convention108/modernised

[9] https://eur-lex.europa.eu/legal-content/FR/TXT/HTML/?uri=CELEX:31995L0046

doivent aussi être compréhensibles pour tout à chacun, et informer les utilisateurs sur les traitements de données personnelles effectués. Les CGU ne sont souvent pas adaptés aux utilisateurs néophytes qui veulent juste utiliser le service rapidement. Elles utilisent souvent des termes techniques ou juridiques non-compréhensibles par un utilisateur lambda, et leur longueur découragent les utilisateurs de les lire [Nitot, 2016, page 86]. Les CGU visent plus à protéger juridiquement le FdS que d'informer les utilisateurs sont les traitements effectués.

Le RGPD exige aussi la protection de la vie privée dès la conception, ainsi que l'obligation d'être sécurisé afin de protéger les données personnelles traitées. Cela demande ainsi de concevoir de nouvelles solutions techniques permettant d'améliorer la vie privée et la sécurité sur Internet. Le RGPD a une portée extra-territoriale, signifiant que toute entité peut être tenue responsable des traitements impliquant des données personnelles d'un citoyen Européen. Cependant, l'application de telles lois sont difficiles dans un contexte international, e.g. pour le droit à l'oubli. Les utilisateurs Européens peuvent en effet demander aux moteurs de recherches de retirer des résultats les concernant, mais ces entrées ne sont retirées que des versions du site à destination du publique Européen[10].

En bref : *La société définie les loi, la morale et l'éthique. La nouvelle réglementation Européenne RGPD défini un cadre légal afin de protéger la vie privée des utilisateurs.*

Modérateurs

Les modérateurs sont des personnalités physiques ou juridiques font appliquer les règles et/ou loi. Ils peuvent être e.g. étatiques, les parents d'un enfant, un employeur, ou le service modération d'un fournisseur de service, faisant appliquer les règles au niveau d'un état, d'une maison, d'une entreprise, ou d'un site Internet. Leurs responsabilités sont définies par la loi (e.g. les responsabilités parentales) et/ou par contrats (e.g. CGU).

Les interactions entre entités requirent un cadre structurant permettant la confiance entre les entités. Ce cadre balance le droit de chaque entité par l'établissement de limites, a.k.a. des règles et lois. Le rôle des modérateurs est de faire appliquer un tel cadre et de sanctionner les contrevenants, s.a. avec des amendes, la privation de droits, ou des bannissements temporaires ou permanent du service. Les modérateurs ont le pouvoir de censurer et de contrôler ce que les lecteurs peuvent lire, pour des fins légitimes (e.g. contenus inadaptés pour les mineurs) ou non (s.a. motivé par des idéologies religieuses, politiques, ou philosophiques). Donner trop de pouvoir aux modérateurs peut conduire à une application arbitraire de la loi, ainsi qu'à des abus,

[10] `http://curia.europa.eu/juris/document/document.jsf?docid=218105&text=&dir=&doclang=FR&part=1&occ=first&mode=DOC&pageIndex=0&cid=4477289`

au détriments des droits des utilisateurs, *"Quis custodiet ipsos custodes?*[11]*", Juvénal.*

Par exemple, l'anonymat et le pseudonymat permettent aux auteurs d'échapper aux conséquences de leurs actions. Cela peut être désirable, l'anonymat permettant les lanceurs d'alertes, les sources journalistiques, ainsi que les témoins, et dissidents politiques de s'exprimer sans craintes. Cependant, dans le même temps, cela peut être non-désirable, comme l'anonymat permettant aussi de transgresser les lois, l'éthique, ou la morale en toute impunité. En effet, e.g. porter plainte contre un utilisateur pour son mauvais comportement requiert que le plaignant, ou les modérateurs, aient connaissance du vrai nom et adresse de l'utilisateur au mauvais comportement.

En général, les systèmes démocratiques ne peuvent exister sans secret, d'où le Droit de L'Homme à la vie privée. En effet, les processus de votes requiert fréquemment le secret du butin afin d'empêcher la coercition de votants, l'achat de votes, ainsi que d'autres fraudes électorales. Le secret de l'instruction est aussi requis au bon fonctionnement de la Justice. Une application trop forte de la loi prive aussi les citoyens de leur Droit de l'Homme à la résistance à l'oppression. Une trop forte surveillance prive aussi les citoyens de leur Liberté d'expression ainsi que de leur Liberté de pensé, du fait de l'effet panoptique [Simon, 2005], où les citoyens se censurent eux-mêmes, et changent leur comportement lorsqu'ils se sentent observés ou surveillés. D'autres secrets sont aussi requis afin d'empêcher des discriminations illégales, e.g. basé sur l'état médical de l'utilisateur, son orientation sexuelle, ou sa religion.

Bien que la sphère privée doit être protégée afin de garantir les droits des citoyens, les modérateurs ont cependant besoin d'entrer dans cette sphères pour sanctionner des infractions graves qui menacent aussi le droit d'autres citoyens. Les modérateurs ont ainsi besoin d'un accès spécial à l'information, mais uniquement dans le cadre d'une procédure encadrée par la loi, avec des gardes-fous permettant la préservation des droits fondamentaux des utilisateurs.

Cela a pour conséquence que les modérateurs n'ont souvent pas le pouvoir d'appliquer entièrement les règles et lois, par exemple, en recourant aux signalements des utilisateurs des contenus ou comportements inappropriés. Cela permet de réduire les ressources assignées aux services de monitorats, mais d'un autre côté, les mauvais comportements non-signalés restent non-sanctionnés.

En bref : *Les modérateurs font appliquer les règles pour permettent les interactions entre utilisateurs. Cependant ils leur manque souvent les moyens de faire correctement appliquer de telles règles, en parti pour empêcher les abus de la part des modérateurs eux-mêmes.*

[11] "Qui va [modérer] les [modérateurs] ?"

7.6.2 Sécurité et vie privée

La sécurité et la vie privée sont souvent présentées comme des concepts opposés, la sécurité protégeant le système contre les utilisateurs, et la vie privée protégeant les utilisateurs contre le système. Cela conduit à un faux-dilemme, demandant aux concepteurs du système de choisir entre sécurité et vie privée [Schneier, 2001].

La sécurité garanti le bon fonctionnement du système qui doit rester *disponible* et *efficace* afin de pouvoir servir ses utilisateurs. Le système doit s'assurer de l'*authenticité* (i.e. véracité d'une déclaration) de l'identité de l'utilisateur (e.g. pour donner l'accès à une ressource), de l'*origine* et *intégrité* des messages (i.e. le message a été envoyés par l'utilisateur déclaré, et n'a pas été modifié par un tiers), ainsi que de l'authenticité de l'information ou des attributs de l'utilisateur (e.g. la véracité des déclarations s.a. l'âge ou le sexe déclaré par l'utilisateur). Les systèmes doivent aussi garantir l'*identification* des utilisateurs, que ce soit pour détecter des multi-comptes, ou pour *empêcher la répudiation* (i.e. nier un acte) afin d'engager la *responsabilité* des utilisateurs. Le système a aussi besoin de la *confidentialité* de l'information afin de rester sécurisé, même s'il a été compromis à un moment donné. Les échanges passés doivent restés sécurités (*Backward Secrecy*) de même que les échanges futurs (*Forward Secrecy*).

La vie privée garanti le droit des utilisateurs d'être laissé seuls. La *confidentialité* des données à caractère personnel doit être garantie, de même que la *non-correspondance* de leur différentes informations ou comptes. La non-correspondance signifie qu'un tiers ne devrait pas être capable de déterminer si deux informations ou comptes appartiennent, ou non, à une même entité. La non-correspondance implique l'*anonymat* ou la *pseudonymat* des utilisateurs, i.e. de ne pas être capable de lier une information à la réelle identité d'un utilisateur. Les utilisateurs ont aussi un *droit à l'oublie*, ainsi que de pouvoir *répudier* leurs propres actions. De plus, ils doivent être capables de contrôler leur information, i.e. de corriger des informations fausses, et de choisir les informations qu'ils révèlent.

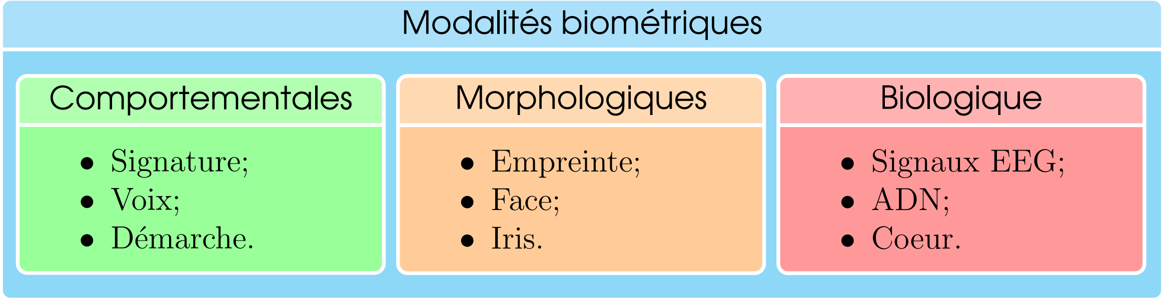
La sécurité et la vie privée peuvent sembler opposées de premier bord, ils ne sont pas incompatibles par nature. Par exemple, la sécurité requiert la non-répudiation quand la vie privée requiert la répudiation, cependant, les deux peuvent être atteint en même temps. En effet, les entités auprès desquelles la non-répudiation doit être atteinte pour la sécurité e.g. modérateurs étatiques dans le cadre d'une procédure spécifique, ne sont pas les même que dans la vie privée, i.e. toutes les autres entités.

7.6.3 Biométrie

Les fournisseurs de services peuvent collecter des données biométrique à travers d'une simple page web. Non seulement ils peuvent identifier et profiler les utilisateurs, ils peuvent aussi vérifier l'essence des utilisateurs et pas uniquement leur connaissance d'un secret. Cela rend les données biométriques typiquement plus difficile à partager, copier, et usurper que e.g. la connaissance ou une possession.

Contrairement aux modalités basées sur une connaissance ou une possession, l'usage de la biométrie est probabiliste, i.e. dans un système d'authentification, les utilisateurs légitimes ont une certaine probabilité d'être rejeté (FRR), et pour un attaquant, d'être accepté (FAR). Cela est due aux variations lors des acquisitions biométriques d'un même utilisateur (variations intra-classes), ainsi qu'aux similarités entre utilisateurs (variations inter-classes).

La biométrie peut aussi constituer des problèmes de vie privée comme ils sont souvent difficilement révocables et renouvelables, et peuvent être utilisés pour déduire des informations personnelles sur les utilisateurs, s.a. leur âge, ou sexe. Cela fait de la biométrie une modalité d'authentification particulière, comme le démontre les articles régulant son usage dans le RGPD.



Modalités biométriques

Comportementales
- Signature;
- Voix;
- Démarche.

Morphologiques
- Empreinte;
- Face;
- Iris.

Biologique
- Signaux EEG;
- ADN;
- Coeur.

Figure 7.2: Modalités biométriques

Nous présentons dans la Figure 7.2 les trois types de modalités biométriques, avec des exemples pour chaques. La biométrie biologique, aussi appelée biométrie cachée, sont des particularités des utilisateurs, invisibles sans les capteurs appropriés, tandis que la biométrie morphologique peut être facilement vue par tous. La biométrie comportementale est basée sur la manière dont les utilisateurs se comportent, s.a. leur façon de marcher [Bours and Denzer, 2018]. Dans cette thèse, nous nous concentrons sur la biométrie comportementale.

Nous nous concentrons sur la biométrie comportementale car elle ne requiert généralement pas d'action supplémentaires de la part de l'utilisateur, et permenttent ainsi une authentification continue et/ou transparente. Malheureusement, l'absence d'action spécifique de l'utilisateur permet aussi son utilisation sans son consentement ou sa connaissance. La biométrie comportementale est aussi sujette à d'importante variations intra-classes, le comportement de l'utilisateur changeant au fil du temps, et dépendant aussi de son état actuel (e.g. fatigué, irrité, triste).

La dynamique de frappe au clavier ainsi que la dynamique d'utilisateur de la souris, i.e. la manière d'utiliser la souris et le clavier, peuvent être facilement collectés sur navigateurs via un simple code JavaScript embarqué dans les pages visitées par l'utilisateur sans nécessiter de capteurs additionnels autre que le clavier et la souris de l'utilisateur. Dans cette thèse, nous nous concentrons sur la dynamique de frappe au clavier, l'une des spécialité du laboratoire GREYC avec plusieurs thèses conduites dans ce domaine [Mhenni et al., 2019, Idrus et al., 2013, Giot, 2012].

7.6.4 Objectifs de la thèse

Cette thèse vise à répondre aux besoin des acteurs d'Internet via l'utilisation de la dynamique de frappe au clavier. Nous permettons aux lecteurs/fournisseurs de service d'utiliser la dynamique de frappe au clavier pour des applications de sécurité, principalement pour l'authentification des utilisateurs, tout en garantissant la vie privée et le consentement explicite des sujets/auteurs, sans nécessiter la supervision des modérateurs. En considérant le RGPD, nous proposons plusieurs contributions à la sécurité et la vie privée sur Internet :

- Nous proposons une anonymisation en temps réel de la dynamique de frappe au clavier afin de prévenir des collectes non-consenties. Cela permet de s'assurer du consentement explicite des utilisateurs pour le traitement de telles données. Nous présentons plusieurs techniques pour protéger la dynamique de frappe dans le Chapitre 3, ainsi qu'une preuve de concept dans le Chapitre 6.

- Nous proposons une authentification biométrique multi-modale compatible avec le RGPD sans dévoiler d'informations privées, les données biométriques n'étant pour dévoilées au fournisseur de service. Cela garanti ainsi que de telles données ne seront pas utilisées par le fournisseur de service pour d'autres finalités que l'authentification. Nous présentons dans le Chapitre 4 un tel protocole d'authentification utilisant la dynamique de frappe au clavier, ainsi que la position de l'utilisateur, et la configuration de son navigateur. Une preuve de concept est ainsi présentée dans le Chapitre 6.

- Une preuve de paternité basée sur les précédentes contributions est ensuite proposée avec une preuve de concept dans le Chapitre 6.

- Une preuve d'identité sociale est proposée dans le Chapitre 6 afin de vérifier l'identité des utilisateurs via une reconnaissance par les pairs, ainsi que pour rendre responsable les utilisateurs au comportement inapproprié envers les modérateurs, via un protocole dédié. La vie privée des utilisateurs reste garantie.

- Nous proposons ensuite l'utilisation de matériels sécurisés dans le Chapitre 6 à fin de protéger la dynamique de frappe des utilisateurs, et ainsi leur vie privée, contre un client corrompu ou malicieux.

- Nous proposons une modélisation de la dynamique de frappe au clavier, ce pour faciliter la recherche et améliorer les performances des systèmes de dynamique de frappe au clavier. La modélisation de la dynamique de frappe au clavier permet leur génération synthétique rendant ainsi possible d'augmenter des bases de données existantes, à terme, permettant de partager des bases de données de dynamique de frappe au clavier pour des finalités de recherches, sans dévoiler aucune information personnelle d'utilisateurs réels.

- Nous présentons dans le Chapitre 2, que l'information peut être déduite à partir de la dynamique de frappe au clavier collectée via le navigateur des utilisateurs, et en quoi elle menace la vie privée des utilisateurs. Nous quantifions cette menace et étudions l'impacte du contexte et de la configuration sur les performances du système de dynamique de frappe au clavier. Dans ce cadre, nous présentons un système de comparaison équitable des systèmes de dynamique de frappe au clavier.

7.7 Contributions de la thèse

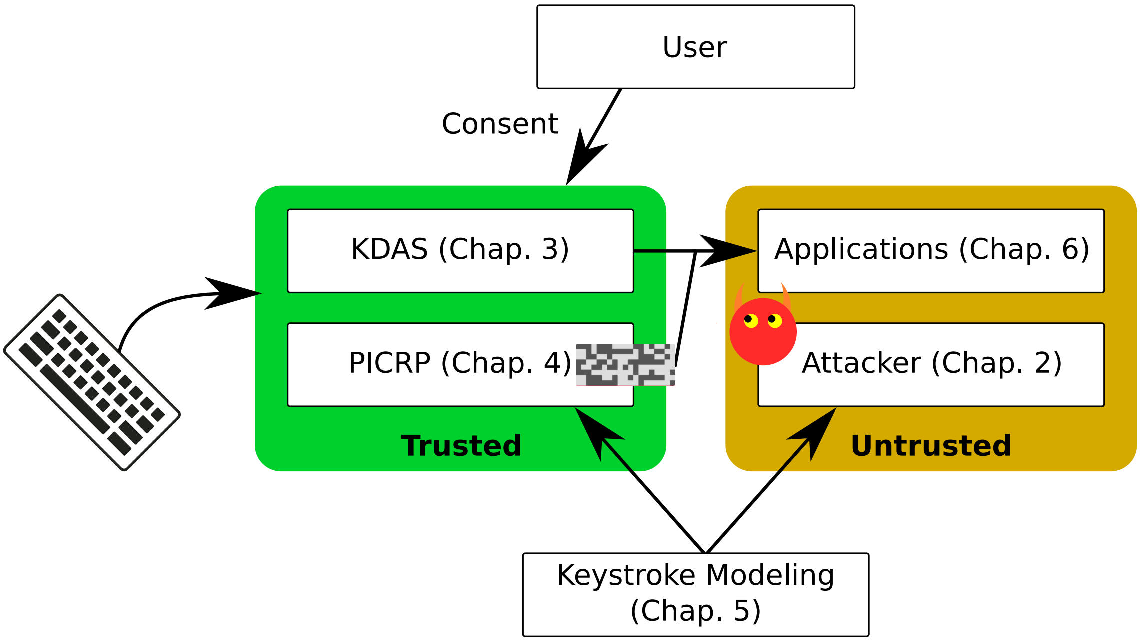


Figure 7.3: Pipeline complet.

Nos contributions permettent une authentification utilisateur consentie basée sur la dynamique de frappe au clavier (en association d'autres modalités) tout en protégeant la vie privée de l'utilisateur. Le pipeline complet est présenté dans la Figure 7.3. Les événements claviers sont collectés au sein d'un composant sécurisé (peut être une WebExtension, le navigateur, le système d'exploitation, ou un matériel dédié) empêchant les composants qui ne sont pas de confiances (peut être une page web, ou le client) d'y avoir accès.

Les événements claviers sont anonymisés avec un Schémas d'Anonymisation de Dynamique de Frappe au Clavier (voir Chapitre 3) afin de transmettre le sens de l'évènement sans sa réelle information temporelle, empêchant ainsi de profiler l'utilisateur sans sont consentement (voir Chapitre 2). Avec le consentement de l'utilisateur, un PICRP est calculé (voir Chapitre 4) et est envoyé aux applications afin de permettre l'authentification de l'utilisateur. La modélisation de la dynamique de frappe au clavier (voir Chapitre 5) est une première étape vers l'amélioration à la fois du PICRP et de l'évaluation de la capacité des attaquant.

7.7.1 Que savent les sites web à votre propos ?

La dynamique de frappe au clavier permet de profiler les utilisateurs (s.a. identification, authentification, détermination de l'âge/sexe/émotions de l'utilisateur) en analysant sa manière de taper au clavier, e.g. lorsque naviguant sur Internet. De nombreuses études visent à améliorer les performances de tels systèmes de dynamique de frappe au clavier [Umphress and Williams, 1985, Monrose and Rubin, 2000, Revett et al., 2007b, Lee and Cho, 2007, Giot et al., 2011], cependant elles ne sont pas comparables, ou difficilement, car elles utilisent différentes bases de données ou différents protocoles [Giot et al., 2011].

Afin d'estimer la menace que représente la dynamique de frappe au clavier vis à vis de la vie privée des utilisateurs, nous avons ainsi besoin d'établir un système de comparaison équitable permettant de sélectionner les meilleurs systèmes de dynamique de frappe au clavier. Dans le cadre de ce travail nous nous concentrons sur les temps de pressions et de relâchement des touches reçues par le système d'exploitation et/ou le navigateur.

Système de comparaison équitable des systèmes de dynamique de frappe au clavier

Nous distinguons 4 composants de notre système de comparaison :

- Le modèle de l'attaquant : décris les capacités de l'attaquant ;
- Métriques : quantifie le succès de l'attaque ;
- Contexte : décris les données obtenues par l'attaquant ;
- Configuration : décris le système de dynamique de frappe au clavier.

Nous assumons un attaquant collectant la dynamique de frappe d'utilisateurs identifiés afin de les reconnaître par la suite ensuite. Nous ne prenons pas en compte les mécanismes de mises à jours de références ainsi que les systèmes réalistes de seuils dépendants de l'utilisateur.

Afin d'évaluer les performances du système de dynamique de frappe au clavier, nous utilisons le taux d'erreur (ER) qui correspond au taux de fausses prédictions sur le nombre total de prédictions. Pour les systèmes dépendant d'un seuil, nous prenons l'*Equal Error Rate* (EER) qui est l'ER correspondant au seuil où le taux de faux positifs (FAR) est égal au taux de faux négatifs (FRR).

Le contexte regroupe plusieurs paramètres allant des bases de données utilisées (dont le nombre d'utilisateur de la base, le texte tapé, sa taille, etc.), ainsi que la manière dont les références et les échantillons sont sélectionnés.

La configuration décrit le système de dynamique de frappe au clavier, i.e. la sélection des caractéristiques, les pré-traitements, la fonction de distance utilisée, ainsi que les processus de fusions de scores ou de références.

Ces paramètres étant soumis à des phénomènes de synergies, l'ensemble des combinaisons doivent être comparées afin d'établir l'influence de la valeur d'un paramètre sur les performances.

Dynamique de frappe au clavier et vie privée

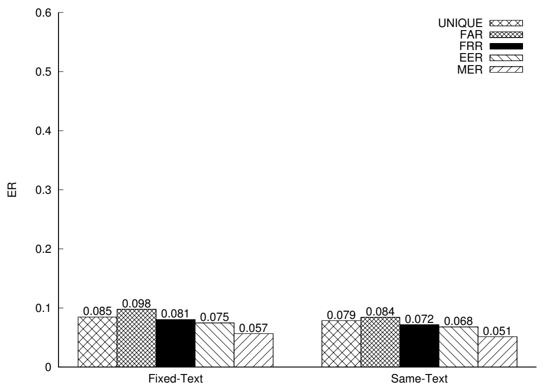


Figure 7.4: Impact du seuil sur les performances d'authentification.

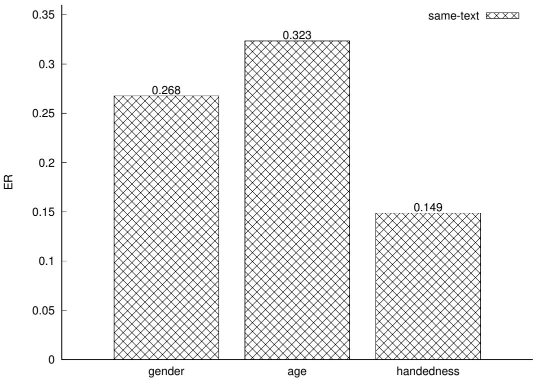


Figure 7.5: Performances du profilage via dynamique de frappe au clavier.

Nous présentons dans la Figure 7.4 les performances des meilleurs systèmes d'authentification de dynamique de frappe au clavier basés sur une fonction de distance en fonction de la capacité de l'attaquant à établir un seuil dépendant de l'utilisateur. Dans la Figure 7.5, nous présentons les performances des meilleurs systèmes de profilage de dynamique de frappe au clavier basé sur un SVM.

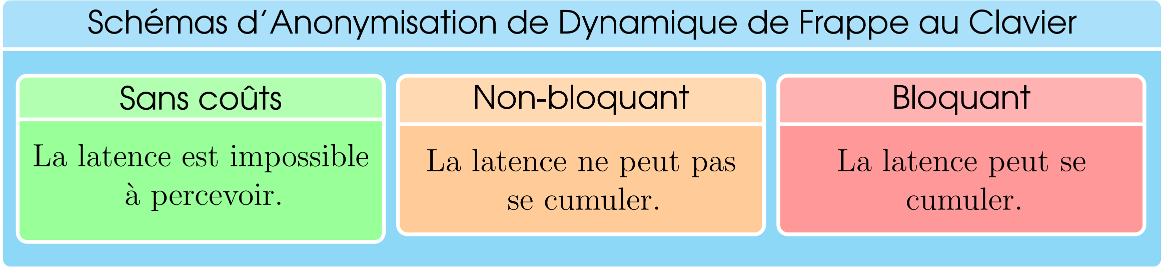
Nous démontrons ainsi que la dynamique de frappe au clavier constitue une menace majeure sur la vie privée des utilisateur qui a besoin d'être adressé. Pour un texte fixe, la valeur d'EER en authentification est de 8,5% et peut descendre à 5,7% avec des seuils dépendant de l'utilisateur. En identification, les performances sont encore meilleure avec un EER de 3,8%.

7.7.2 Comment protéger mes informations de sites web malicieux ?

Afin de protéger la vie privée des utilisateurs, nous proposons différentes stratégies permettant de perturber la capture de la dynamique de frappe au clavier, tout en permettant son exploitation pour des usages légitimes et consentis par l'utilisateur.

Plusieurs stratégies existent afin de protéger les informations sensibles, les différentes méthodes proposées s'en inspirent. Les données peuvent être simplement supprimées, la précision diminuée, les valeurs randomisées, de fausses informations ajoutées, ou standardisées.

Bien évidemment, la protection de la dynamique de frappe au clavier s'accompagne de plusieurs contraintes. Le sens du contenu tapé ne doit pas changer. La protection s'effectuant en temps réel, la latence doit être minimale et aussi imperceptible que possible. Les évènements de pressions ne peuvent aussi qu'être retardés, et ne peuvent pas être prédis (sauf en cas d'utilisation de fonctionnalités d'auto-complétions), l'ordre des évènements de pression doit être conservé.



Schémas d'Anonymisation de Dynamique de Frappe au Clavier

Sans coûts

La latence est impossible à percevoir.

Non-bloquant

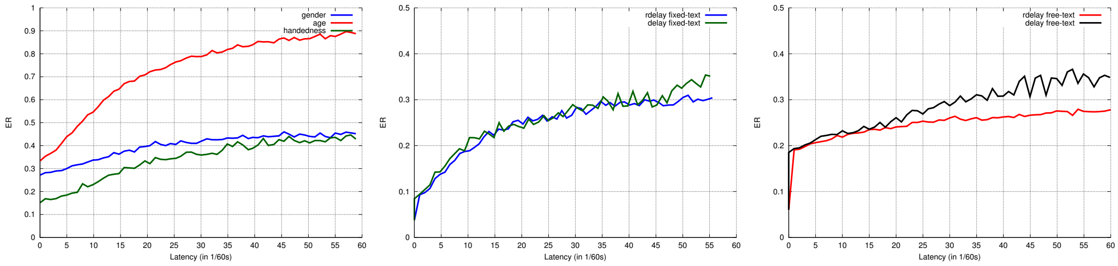
La latence ne peut pas se cumuler.

Bloquant

La latence peut se cumuler.

Figure 7.6: Schémas d'Anonymisation de Dynamique de Frappe au Clavier.

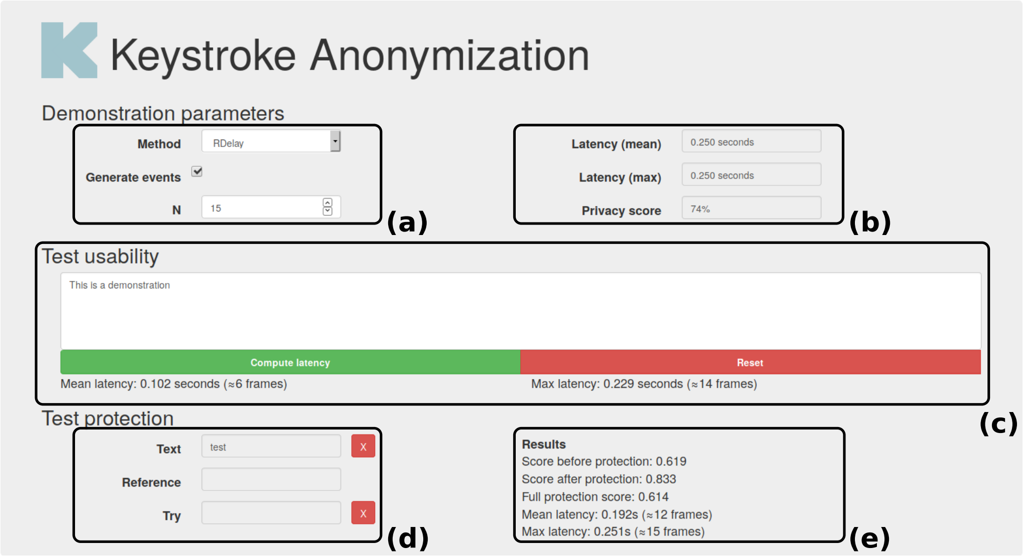
Nous proposons ainsi des Schémas d'Anonymisation de Dynamique de Frappe au Clavier (SADFS) dont nous distinguons 3 familles comme montré par la Figure 7.6. Nous nous concentrons dans cette synthèse sur les SADFS non-bloquant.



(a) Biométrie douce (rdelay) (b) Identification sur texte fixe (c) Ident. sur texte libre

Figure 7.7: Performances de SADFS non-bloquants.

La Figure 7.7, montre le gain de vie privée offert par les SADFS. Les gains sont significatifs, cependant une forte protection de la vie privée vient au prix d'une diminution de l'utilisabilité, i.e. d'une forte latence.



Keystroke Anonymization

Demonstration parameters

Method: RDelay

Generate events: ☑

N: 15

(a)

Latency (mean): 0.250 seconds

Latency (max): 0.250 seconds

Privacy score: 74%

(b)

Test usability

This is a demonstration

Mean latency: 0.102 seconds (≈6 frames)

Max latency: 0.229 seconds (≈14 frames)

(c)

Test protection

Text: test

Reference:

Try:

(d)

Results

Score before protection: 0.619

Score after protection: 0.833

Full protection score: 0.614

Mean latency: 0.192s (≈12 frames)

Max latency: 0.251s (≈15 frames)

(e)

Figure 7.8: Interface de la démonstration d'anonymisation.

Nous proposons une démonstration dont l'interface est présentée dans la Figure 7.8. Le SADFS est configuré dans la zone (a), ses performances sont données dans la zone (b). La zone (c) permet de tester l'utilisabilité (latence) du SADFS. La zone (d) permet d'en tester la sécurité, les résultats sont présentés dans la zone (e).

L'implémentions d'un SADFS doit se faire avec prudence, d'autant plus s'il est implémenté sous la forme d'une WebExtension. En effet, le mécanisme de protection peut être facilement contourné si mal configuré ou implémenté. Notamment le gestionnaire d'évent du SADFS doit être appelé avant tout autre afin de retarder les évènements claviers. De plus, ce retardement doit se faire via une attente passive, ce qui requiert la re-création des évènements qui perdent alors leur statu de "confiance", i.e. les comportements par défauts des touches claviers doivent être simulées.

7.7.3 Utiliser des données personnelles dans un schéma d'authentification respectueux de la vie privée.

Nous proposons le concept de Code d'Identité Personnel Respectueux de la Vie Privée (PICRP), une signature non-cryptographique irréversible calculée à partir de données arbitraires. Dans nos expériences, nous utilisons la dynamique de frappe au clavier, l'adresse IP, et la géo-localisation GPS. Les données sont pré-traitées puis fusionnées suivant différentes stratégies.

Cette signature a la particularité d'être multi-modale, de ne pas être dépendant d'une modalité donnée, d'être irréversible et renouvelable, avec la possibilité d'adapter le niveau de sécurité à la volée, le tout sans requérir d'éléments sécurisés. Nous l'utilisons principalement à finalité d'authentification, la comparaison de deux signatures permettant de comparer les données d'origines sans les dévoiler. En effet, tout type de modalité peut être utilisées dans notre signature tant qu'elle peut être représentée sous la forme d'un vecteur de réel de taille fixe. Le changement d'un mot de passe connu par l'utilisateur permet la renouvelabilité de notre signature.

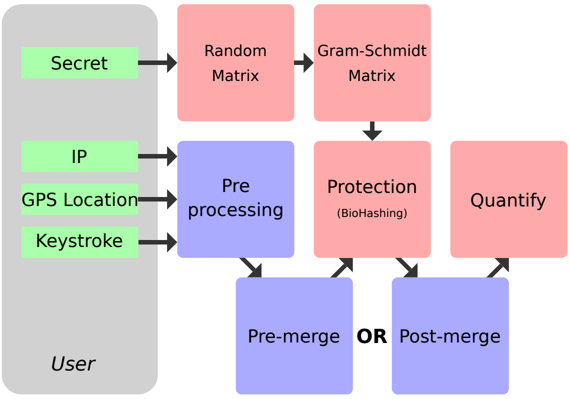
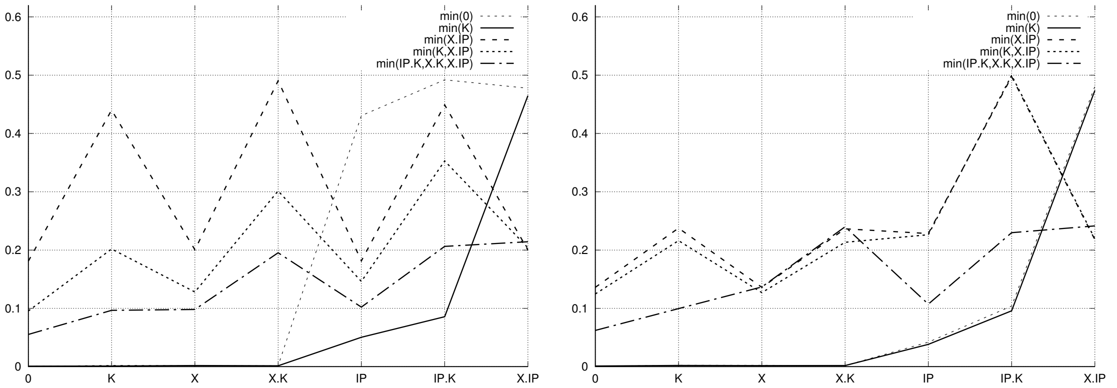


Figure 7.9: Pipeline du calcul du PICRP

Le calcul du PICRP est montré dans la Figure 7.9. En rouge, les étapes correspondant au calcul du BioHashing, une méthode permettant de transformer un vecteur de réel en un vecteur binaire non-inversible. En vert les données utilisées pour le calcul du PICRP, et en bleu les étapes ajoutées pour pré-traiter et fusionner les données.

Le BioHashing est calculé comme le produit matriciel entre un vecteur de réel, et une matrice aléatoire orthogonalisée via l'algorithme de Gram-Schmidt. Le vecteur de réel résultant est binarisé afin de produire un vecteur binaire. Le secret de l'utilisateur est utilisé comme une graine afin de générer la matrice aléatoire.

Les méthodes proposées sont évaluées à travers 7 scénarios, nommées comme la concaténation des modalités assumées connues par l'attaquant, dont la dynamique de frappe au clavier (K), la position GPS (X), et l'adresse IP (IP). Toutes les attaques assument que l'attaquant a connaissance du secret de l'utilisateur utilisé pour le BioHashing. 0 correspond au scénario où l'attaquant ne possède pas de connaissance supplémentaires. Les résultats sont présentés dans la Figure 7.10.



(a) Pre-fusion.

(b) Post-fusion.

Figure 7.10: Meilleures configurations sous 7 scénarios.

Les configurations minimisant la valeur de l'EER sous le scénario (0) produisent de très bon EER ($< 1\%$), mais sont peu performantes (EER $> 40\%$) si l'adresse IP ou la localisation GPS est connue des attaquants. D'autres configurations garantissent une valeur d'EER $\lesssim 20\%$ sous tous les scénarios, au coût d'une valeur d'EER plus mauvaise sous le scénario (0).

Un exemple de protocole d'authentification utilisant PICRP est proposé dans la Figure 7.11, mais n'est pas détaillé ici. Le principe général étant d'utiliser le PICRP pour reconstruire un aléa et d'utiliser cet aléa pour s'authentifier, empêchant ainsi le service d'accéder à la moindre information.

Une démonstration permettant de comparer des PICRP a été développée. Son interface est présentée dans la Figure 7.12. Cette démonstration permet de créer (a) ou de copier (b) deux PICRP (c) puis de visualiser leurs différences (d), ainsi que d'obtenir un score de similarité pour chaque modalités (e).

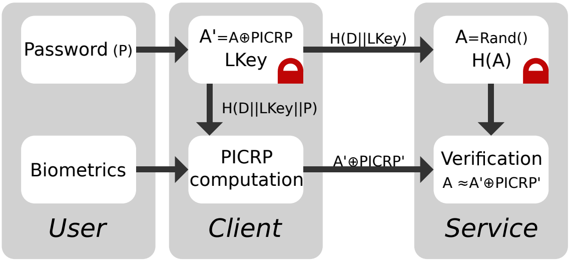
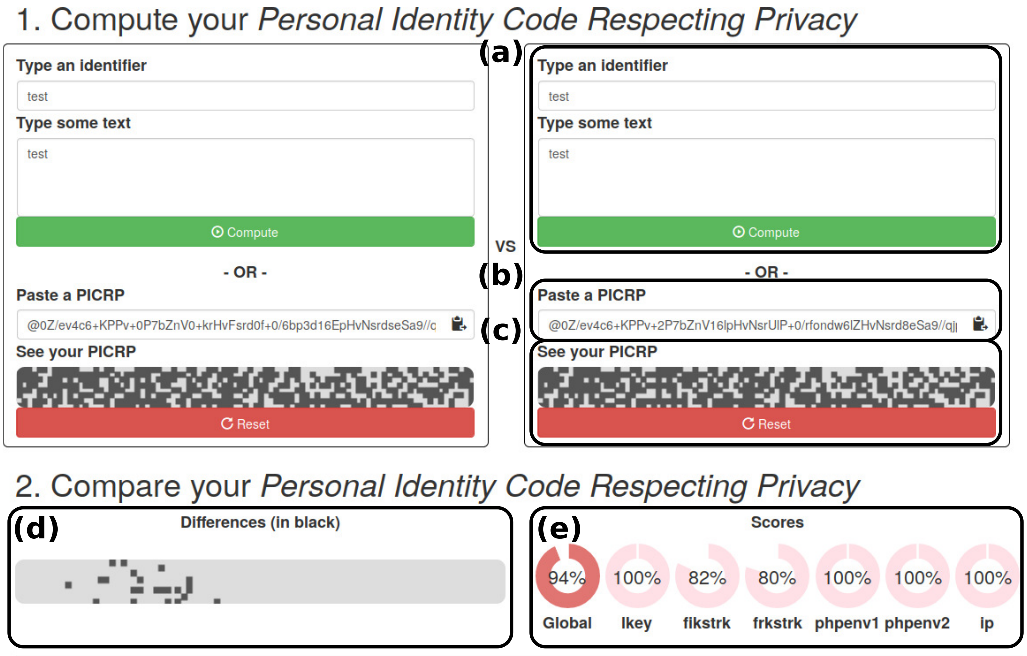


Figure 7.11: Schéma d'authentification basé sur PICRP.


1. Compute your *Personal Identity Code Respecting Privacy*

Type an identifier: test
Type some text: test
Compute
- OR -
Paste a PICRP: @0Z/ev4c6+KPPv+0P7bZnV0+krHvFsrd0f+0/6bp3d16EpHvNsrdseSa9//c
See your PICRP
Reset

Type an identifier: test
Type some text: test
Compute
- OR -
Paste a PICRP: @0Z/ev4c6+KPPv+2P7bZnV16lpHvNsrUlP+0/rfondw6lZHvNsrd8eSa9//qj
See your PICRP
Reset

2. Compare your *Personal Identity Code Respecting Privacy*

Differences (in black)

Scores

Figure 7.12: Interface de la démonstration d'authentification.

7.7.4 Comment utiliser cet données à finalité de recherche tout en respectant la vie privée des utilisateurs ?

La construction de systèmes biométrique nécessitant de grandes quantités de données. Cependant, la collecte de dynamique de frappe au clavier étant très chronophage et contraint, e.g. par le RGPD. Pour ces raisons il est crucial de générer des bases de données synthétiques de dynamique de frappe au clavier. Nous proposons ainsi une méthode de génération de la dynamique de frappe au clavier pour un utilisateur connu, comme première étape à la génération de telles bases synthétiques. La seconde étape étant alors la génération d'utilisateurs.

La méthode proposée, sur texte fixe, assume l'indépendance des durées de pressions/relâchements. Ces durées sont représentées par une variable aléatoire suivant une loi dont les paramètres varient d'un utilisateur à l'autre. La dynamique de frappe au clavier est assumée ne pas évoluer au court du temps. La génération s'effectue donc par l'estimation de ces paramètres pour chaque utilisateurs, puis par la génération aléatoire de nouvelles valeurs à partir de ces lois.

Nous considérons trois métriques, l'estimation de l'EER des données réelles via les données synthétiques (EEE), la capacité d'usurpation des données synthétiques (ERS), et l'aire entre les courbes ROC des données synthétiques et réelles (ABS).

Comme le montre la Figure 7.13, les configurations testées montre un compromis entre l'EEE et l'ERS, bien que certaines configurations donnent des EEE et ERS satisfaisant.

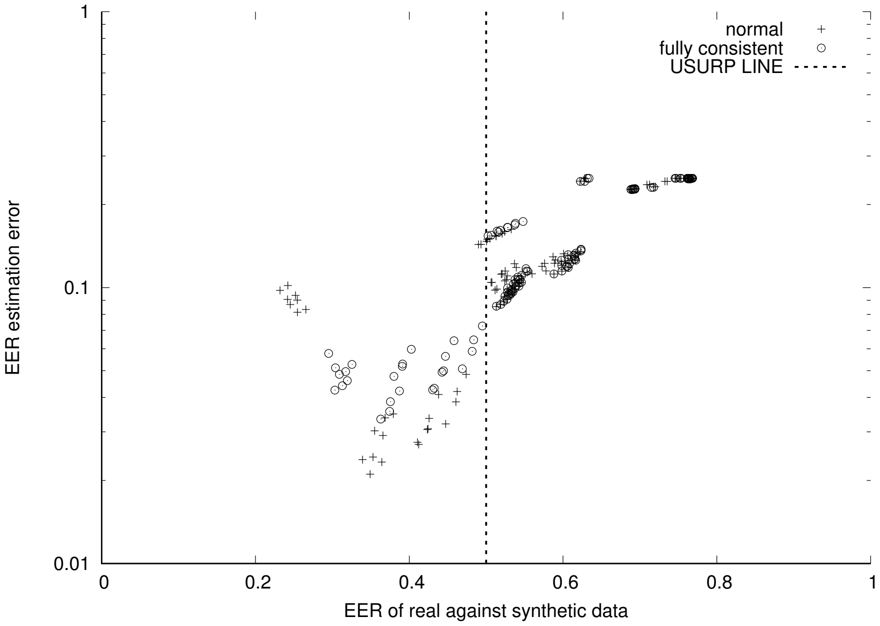


Figure 7.13: Performances des configurations de génération synthétiques de dynamique de frappe au clavier.

7.7.5 Quelques exemples d'applications avec preuves de concept

Preuve de parternité

Nous proposons l'utilisation du PICRP comme preuve de paternité d'un texte écrit. L'utilisateur est authentifié de manière continue lors de l'écriture collaborative d'un document. L'authentification continue prévient la répudiation du travail fournit par un auteur ainsi que l'usurpation d'identité non-consentie. Cependant cela ne protège pas contre les faux-comptes, ou les auteurs fantômes.

Preuve sociale d'identité

Nous proposons aussi une preuve sociale d'identité permettant à un tiers de vérifier l'identité d'une personne sur un réseau social. L'identité d'une personne est garantie par des membres de ce même réseau, leur identité étant elle même garantie par d'autres membres du réseau. PICRP est alors utilisé comme moyen d'authentification continue et transparente afin de garantir que le compte demeure dans les mains du même utilisateur.

La certification de l'identité associée à un compte est représentée sous forme d'un certificat cryptographique. Pour que l'identité soit confirmée, le compte doit recevoir t certificats provenant de comptes dont l'identité a été confirmée. Des entités externes, les *Root Node Certifiers* (e.g. un État), peuvent aussi vérifier l'identité de comptes de manière plus exhaustive (e.g. via une carte nationale d'identité), dès lors une seule certification est requise. Ces comptes sont appelés *Root Nodes* (RN). Une représentation est donnée dans la Figure 7.14.

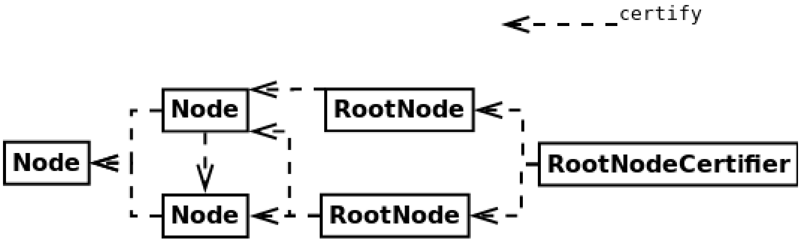
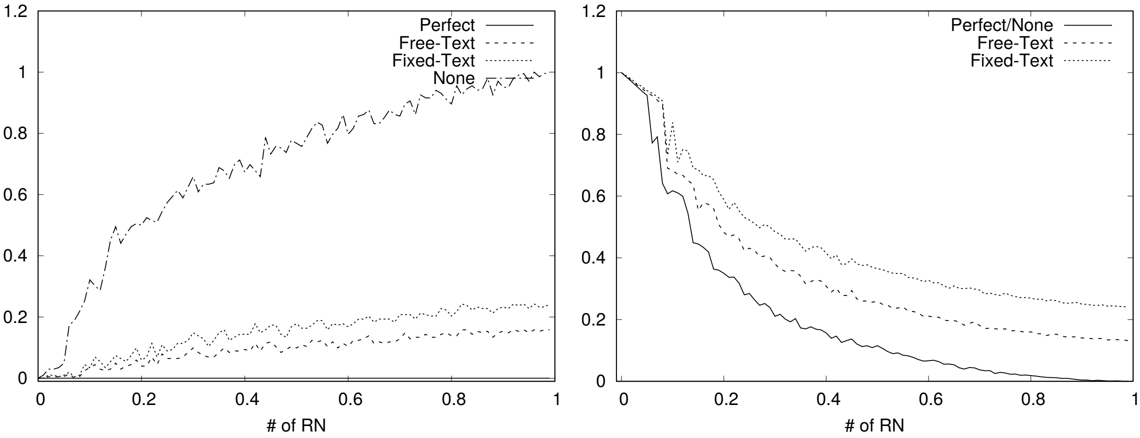


Figure 7.14: Certification avec $t = 2$

Nous visons aussi à engager la responsabilité des nœuds en cas de comportements inappropriés, tout en respectant la vie privée des utilisateurs, i.e. en minimisant la quantité d'informations que nous dévoilons. Nous ne dévoilons pas ces aspects dans cette synthèse.



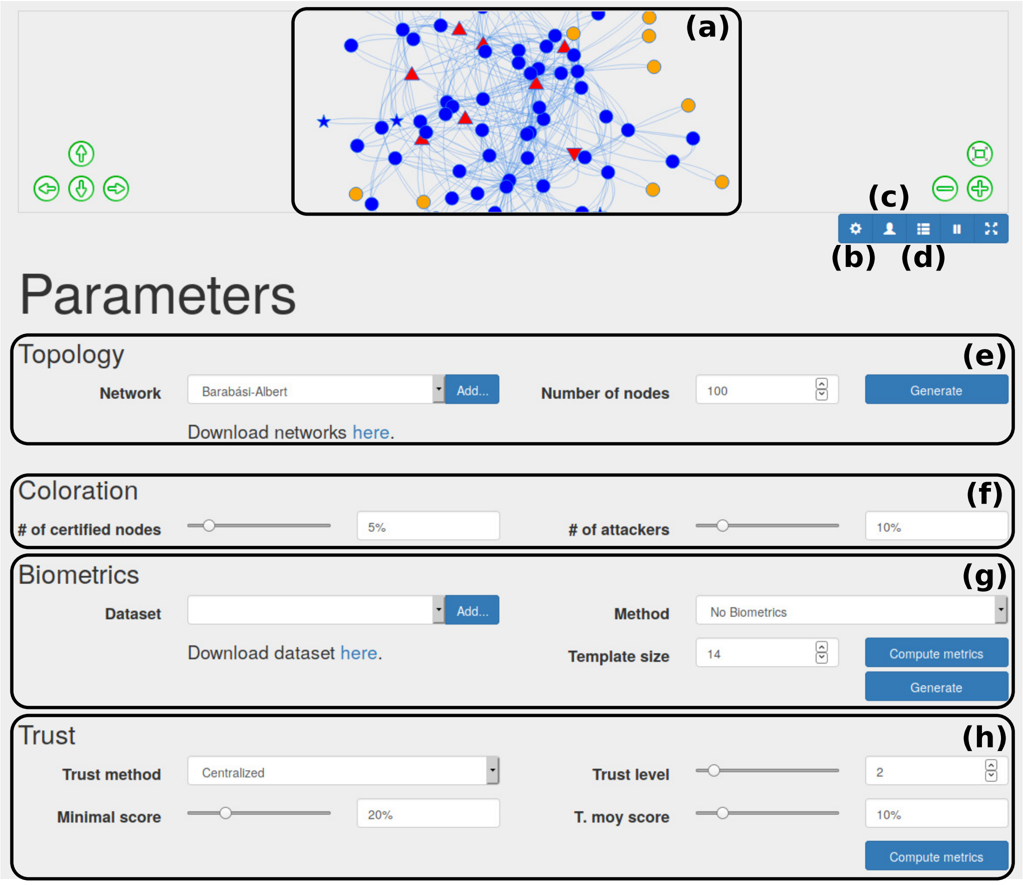
(a) FCR

(b) FNCR

Figure 7.15: FCR et FNCR en fonction du taux de RN, avec $t = 15$, et 5% de comptes compromis.

La performance de notre proposition est présentée dans la Figure 7.15. Nous comparons plusieurs scénario, l'un sans authentification (None), un théorique avec une authentification parfaite (Perfect), et deux avec une authentification basés respectivement sur du texte fixe (Fixed-Text) et du texte libre (Free-Text). Nous regardons le taux de fausse confirmation (FCR) ainsi que le taux de fausses non-confirmations (FNCR) correspondant aux taux d'attaquants faussement confirmé, et au taux d'utilisateurs faussement non-confirmés. Le FNCR est calculé sur l'ensemble des nœuds candidats, i.e. l'ensemble des RN ou des nœuds avec au moins t certificats. Dans cette expérience, nous avons utilisé $t = 15$, un nombre de certificats requis plus faible devrait faire baisser le FNCR, notamment lorsque le taux de RN est faible.

Une preuve de concept a été developpé, son interface graphique est présenté dans la Figure 7.16. La zone (a) donne une représentation visuelle du réseau configuré via la zone (e). La coloration du graphe (f) détermine le nombre d'attaquant et de Root Nodes. La zone (g) détermine la manière dont l'authentification est effectuée, et la zone (h) la manière dont les nœuds deviennent certifiés. Plusieurs métriques sont ensuite calculées.



Parameters

Topology

Network: Barabási-Albert

Number of nodes: 100

Download networks here.

Coloration

# of certified nodes: 5%

# of attackers: 10%

Biometrics

Dataset:

Method: No Biometrics

Download dataset here.

Template size: 14

Trust

Trust method: Centralized

Trust level: 2

Minimal score: 20%

T. moy score: 10%

Figure 7.16: Interface graphique de la preuve de concept de la preuve d'identité sociale.

Améliorer la sécurité via un matériel dédié

Afin de protéger les utilisateurs d'un client corrompu ou malicieux (e.g. ordinateur publique ou d'entreprise), nous proposons l'utilisation d'un matériel dédié afin d'anonymiser la dynamique de frappe au clavier et de calculer des PICRP sans exposer l'information biométrique sur le client.

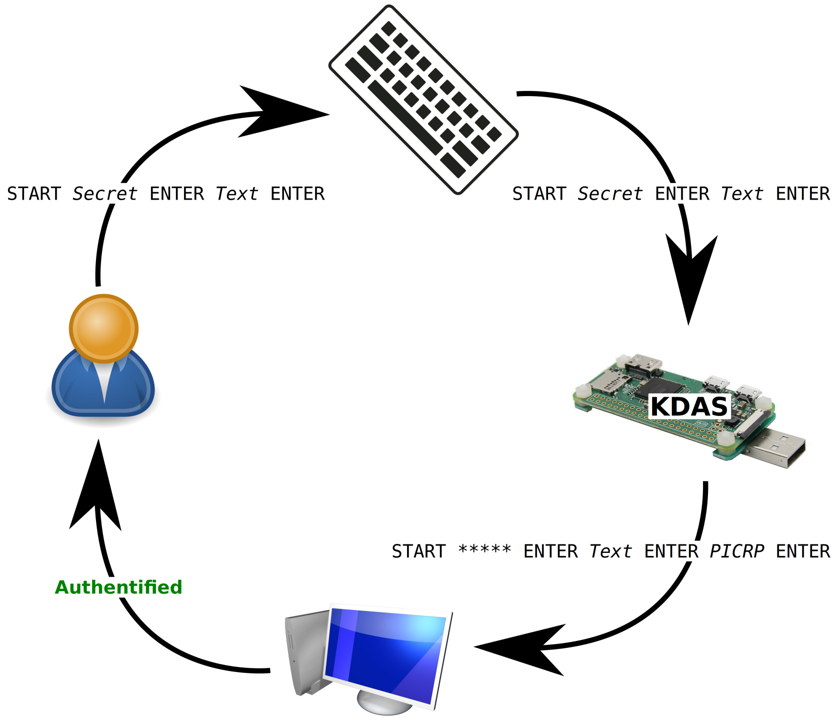

START *Secret* ENTER *Text* ENTER
START *Secret* ENTER *Text* ENTER
KDAS
START ***** ENTER *Text* ENTER *PICRP* ENTER
Authentified

Figure 7.17: PICRP sur Raspberry Pi.

Comme montré par la Figure 7.17, la communication est mono-directionnelle. Le matériel spécifique se place entre le clavier et le client se faisant passer pour un clavier et retardant les événements qu'il reçoit du vrai clavier.

Afin de pouvoir générer, à la demande de l'utilisateur, des PICRP, des combinaisons de touches spéciales (START et ENTER) sont nécessaires afin d'envoyer des commandes qui se distinguent de la simple saisie de texte. Le secret tapé par l'utilisateur n'est pas pas transmis au client, chaque caractère est remplacé par un astérisque. Ces commandes doivent aussi être interprétées par le client (e.g. via une WebExtension) afin de collecter le PICRP envoyé par le matériel.

Thesis Publications

International Journals

[1] Denis Migdal and Christophe Rosenberger. Statistical Modeling of Keystroke Dynamics Samples For the Generation of Synthetic Datasets. *Elsevier Journal on Future Generation Computer Systems, Special Issue on CyberSecurity & Biometrics for a better Cyberworld (Q1 - JCR)*, 2019.

International Conferences

[2] Denis Migdal and Christophe Rosenberger. My Behavior is my Privacy & Secure Password ! In *Cyberworlds (B - Core)*, Kyoto, Japan, October 2019.

[3] Denis Migdal and Christophe Rosenberger. Keystroke Dynamics Anonymization System. In *SeCrypt (B - Core)*, Prague, Czech Republic, July 2019.

[4] Abir Mhenni, Denis Migdal, Estelle Cherrier, Christophe Rosenberger, and Najoua Essoukri Ben Amara. Vulnerability of Adaptive Strategies of Keystroke Dynamics Based Authentication Against Different Attack Types. In *Cyberworlds*, Kyoto, Japan, October 2019.

[5] Denis Migdal and Christophe Rosenberger. Analysis of Keystroke Dynamics For the Generation of Synthetic Datasets. In *CyberWorlds (B - Core)*, Singapour, Singapore, October 2018.

[6] Denis Migdal and Christophe Rosenberger. Towards a Personal Identity Code Respecting Privacy. In *International Conference on Information Systems Security and Privacy (ICISSP)*, Madeira, Portugal, January 2018.

[7] Francesco Buccafurri, Gianluca Lax, Denis Migdal, Serena Nicolazzo, Antonino Nocera, and Christophe Rosenberger. Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *CyberWorlds (B - Core)*, Chester, United Kingdom, September 2017.

[8] Denis Migdal, Christian Johansen, and Audun Jøsang. Offline Trusted Device and Proxy Architecture based on a new TLS Switching technique. In *International*

Workshop on Secure Internet of Things SIOT 2017 (ESORICS Workshop - A - Core), Oslo, Norway, September 2017.

National Journals

[9] Denis Migdal and Christophe Rosenberger. Protection de données personnelles pour la sécurité sur Internet. *Technique et Science Informatiques*, 2019.

National Conferences

[10] Denis Migdal and Christophe Rosenberger. Schéma d'Anonymisation de Dynamique de Frappe au Clavier. In *APVP*, Cap Hornu, France, July 2019.

[11] D Migdal and C Rosenberger. Protection de données personnelles pour la sécurité sur Internet. In *Atelier sur la Protection de la Vie Privée*, Porquerolles, France, June 2018.

[12] Francesco Buccafurri, Gianluca Lax, Denis Migdal, Serena Nicolazzo, Antonino Nocera, and Christophe Rosenberger. Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *Italian Conference on CyberSecurity (ITASEC)*, Milan, Italy, February 2018.

[13] Denis Migdal and Christophe Rosenberger. Towards a Personal Identity Code Respecting Privacy. In *International Conference on Information Systems Security and Privacy (ICISSP)*, Madeira, Portugal, January 2018.

[14] Denis Migdal and Christophe Rosenberger. Vers un Code Personnel d'Identité Respectueux de la Vie Privée. In *CORESA*, Caen, France, November 2017.

[15] Denis Migdal and Audun Jøsang. OffPAD – Objet Personnel d'Authentification Hors-ligne appliqué aux hôpitaux et banques en ligne. In *RESSI 2017*, Grenoble, France, May 2017.

Posters & Demos

[16] Denis Migdal and Christophe Rosenberger. Don't listen to my Keystroke Dynamics! (Summer School). 16th Int.l Summer School on Biometrics and Forensics 2019, May 2019.

[17] Denis Migdal, Christian Johansen, and Audun Jøsang. DEMO: OffPAD-Offline Personal Authenticating Device with Applications in Hospitals and e-Banking. In *ACM CCS 2016 (A* - Core)*, Vienna, Austria, October 2016.

Bibliography

[Acar et al., 2013] Acar, G., Juarez, M., Nikiforakis, N., Diaz, C., Gürses, S., Piessens, F., and Preneel, B. (2013). Fpdetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1129–1140. ACM.

[al, 2014] al (2014). *Guide d'autodéfence numérique*. Éditions Tahin Party.

[Allen, 1996] Allen, A. L. (1996). Constitutional law and privacy. *A companion to philosophy of law and legal theory*, pages 139–155.

[Atighehchi et al., 2019] Atighehchi, K., Ghammam, L., Barbier, M., and Rosenberger, C. (2019). Greyc-hashing: Combining biometrics and secret for enhancing the security of protected templates. *Future Generation Computer Systems*, 101:819–830.

[Banerjee et al., 2014] Banerjee, R., Feng, S., Kang, J. S., and Choi, Y. (2014). Keystroke patterns as prosody in digital writings: A case study with deceptive reviews and essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1469–1473, Doha, Qatar. Association for Computational Linguistics.

[Barabási and Albert, 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439):509–512.

[Barbier and Rosenberger, 2014] Barbier, M. and Rosenberger, C. (2014). Tatouage d'images avec des données biométriques révocables pour la preuve de propriété. In *Conférence sur la Sécurité des Architectures Réseaux et des Systèmes d'Information (SAR SSI)*.

[Bjork et al., 2009] Bjork, B.-C., Roos, A., and Lauri, M. (2009). Scientific journal publishing: yearly volume and open access availability. *Information Research: An International Electronic Journal*, 14(1).

[Bleha et al., 1990] Bleha, S., Slivinsky, C., and Hussien, B. (1990). Computer-access security systems using keystroke dynamics. *IEEE Transactions on pattern analysis and machine intelligence*, 12(12):1217–1222.

[Boda et al., 2012] Boda, K., Földes, Á., Gulyás, G., and Imre, S. (2012). User tracking on the web via cross-browser fingerprinting. *Information Security Technology for Applications*, pages 31–46.

[Bours and Denzer, 2018] Bours, P. and Denzer, T. (2018). Cross-pocket gait recognition. In *2018 International Conference on Cyberworlds (CW)*, pages 331–338. IEEE.

[Boyd, 2014] Boyd, D. (2014). *It's complicated.* Yale university press.

[Buccafurri et al., 2013] Buccafurri, F., Foti, V. D., Lax, G., Nocera, A., and Ursino, D. (2013). Bridge analysis in a social internetworking scenario. *Information Sciences*, 224:1–18.

[Buccafurri et al., 2017a] Buccafurri, F., Lax, G., Migdal, D., Nicolazzo, S., Nocera, A., and Rosenberger, C. (2017a). Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *CyberWorlds (B - Core)*, Chester, United Kingdom.

[Buccafurri et al., 2017b] Buccafurri, F., Lax, G., Migdal, D., Nicolazzo, S., Nocera, A., and Rosenberger, C. (2017b). Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *CyberWorlds (B - Core)*, Chester, United Kingdom.

[Buccafurri et al., 2018] Buccafurri, F., Lax, G., Migdal, D., Nicolazzo, S., Nocera, A., and Rosenberger, C. (2018). Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement. In *Italian Conference on CyberSecurity (ITASEC)*, Milan, Italy.

[Cao and Wijmans, 2017] Cao, S. Y. and Wijmans, E. (2017). Browser fingerprinting via os and hardware level features. *Network & Distributed System Security Symposium, NDSS*, 17.

[Cappelli et al., 2004] Cappelli, R., Maio, D., and Maltoni, D. (2004). Sfinge: an approach to synthetic fingerprint generation. In *International Workshop on Biometric Technologies (BT2004)*, pages 147–154.

[Cha et al., 2010] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, K. P. (2010). Measuring user influence in twitter: the million follower fallacy. In *Fourth International AAAI Conference on Weblogs and Social Media (ICWSM 2010)*, pages 10–17. AAAI Press.

[Clarke and Furnell, 2007] Clarke, N. L. and Furnell, S. M. (2007). Authenticating mobile phone users using keystroke analysis. *International journal of information security*, 6(1):1–14.

[Cordier, 2015] Cordier, A. (2015). *Grandir connectés.* C&F éditions.

[DB-IP, 2019] DB-IP (2019). Ip geolocation api and database. `https://db-ip.com/`.

[Eckersley, 2010] Eckersley, P. (2010). How unique is your web browser? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–18. Springer.

[Epp, 2010] Epp, C. (2010). Identifying emotional states through keystroke dynamics. Master's thesis, University of Saskatchewan, Saskatoon, CANADA.

[Fox et al., 2000] Fox, S., Rainie, L., Horrigan, J., Lenhart, A., Spooner, T., and Carter, C. (2000). Trust and privacy online: Why americans want to rewrite the rules. *The Pew Internet & American Life Project*, pages 1–29.

[Gaines et al., 1980] Gaines, R., Lisowski, W., Press, S., and Shapiro, N. (1980). Authentication by keystroke timing: some preliminary results. Technical Report R-2567-NSF, Rand Corporation.

[Giot, 2012] Giot, R. (2012). *Contributions à la dynamique de frappe au clavier: multibiométrie, biométrie douce et mise à jour de la référence.* PhD thesis, S.I.M.E.M.

[Giot et al., 2012] Giot, R., Abed, M. E., and Rosenberger, C. (2012). Web-based benchmark for keystroke dynamics biometric systems: a statistical analysis. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*, pages 11–15. IEEE.

[Giot et al., 2015] Giot, R., Dorizzi, B., and Rosenberger, C. (2015). A review on the public benchmark databases for static keystroke dynamics. *Computers & Security*, 55:46–61.

[Giot et al., 2011] Giot, R., El-Abed, M., Hemery, B., and Rosenberger, C. (2011). Unconstrained keystroke dynamics authentication with shared secret. *Computers & Security*, 30(6-7):427–445.

[Giot et al., 2009] Giot, R., El-Abed, M., and Rosenberger, C. (2009). Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*, pages 1–6.

[Giot and Rosenberger, 2012] Giot, R. and Rosenberger, C. (2012). A new soft biometric approach for keystroke dynamics based on gender recognition. *International Journal of Information Technology and Management (IJITM). Special Issue on : "Advances and Trends in Biometrics by Dr Lidong Wang*, 11(1/2):35–49.

[Hocquet et al., 2007] Hocquet, S., Ramel, J.-Y., and Cardot, H. (2007). User classification for keystroke dynamics authentication. In *The Sixth International Conference on Biometrics (ICB2007)*, pages 531–539.

[Idrus et al., 2013] Idrus, S. Z. S., Cherrier, E., Rosenberger, C., and Bours, P. (2013). Soft biometrics for keystroke dynamics. In *Image Analysis and Recognition*, pages 11–18. Springer.

[Iorliam et al., 2015] Iorliam, A., Ho, A., Poh, N., Tirunagari, S., and Bours, P. (2015). Data forensic techniques using benford's law and zipf's law for keystroke dynamics. *3rd International Workshop on Biometrics and Forensics, IWBF 2015.*

[Jansen and Martin, 2015] Jansen, S. C. and Martin, B. (2015). The streisand effect and censorship backfire. *International Journal of Communication*, 9.

[Jorgensen and Yu, 2011] Jorgensen, Z. and Yu, T. (2011). On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 476–482. ACM.

[Killourhy and Maxion, 2008] Killourhy, K. and Maxion, R. (2008). The effect of clock resolution on keystroke dynamics. In *Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection*, pages 331–350. Springer.

[Killourhy and Maxion, 2009] Killourhy, K. S. and Maxion, R. A. (2009). Comparing anomaly detectors for keystroke dynamics. In *Proc. of the 39th Ann. Int. Conf. on Dependable Systems and Networks*, pages 125–134.

[Killourhy and Maxion, 2011] Killourhy, K. S. and Maxion, R. A. (2011). Should security researchers experiment more and draw more inferences? In *4th Workshop on Cyber Security Experimentation and Test (CSET'11)*, pages 1–8.

[Kim et al., 2018] Kim, J., Kim, H., and Kang, P. (2018). Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection. *Applied Soft Computing*, 62:1077–1087.

[KPFRS, 1901] KPFRS, L. (1901). On lines and planes of closest fit to systems of points in space. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (SIGMOD).*

[Kumar et al., 2010] Kumar, R., Novak, J., and Tomkins, A. (2010). Structure and evolution of online social networks. In *Link mining: models, algorithms, and applications*, pages 337–357. Springer.

[Lacharme et al., 2013] Lacharme, P., Cherrier, E., and Rosenberger, C. (2013). Preimage attack on biohashing. In *2013 International Conference on Security and Cryptography (SECRYPT)*, pages 1–8. IEEE.

[Lacharme and Plateaux, 2011] Lacharme, P. and Plateaux, A. (2011). Pin-based cancelable biometrics. *International Journal of Automated Identification Technology (IJAIT)*, 3(2):75–79.

[Laperdrix, 2017] Laperdrix, P. (2017). *Browser Fingerprinting: Exploring Device Diversity to Augment Authentification and Build Client-Side Countermeasures.* PhD thesis, Rennes, INSA.

[Laperdrix et al., 2016] Laperdrix, P., Rudametkin, W., and Baudry, B. (2016). Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. *Security and Privacy (SP)*, pages 878–894.

[Learned-Miller et al., 2016] Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H., and Hua, G. (2016). Labeled faces in the wild: A survey. In *Advances in face detection and facial image analysis*, pages 189–248. Springer.

[Lee and Cho, 2007] Lee, H. and Cho, S. (2007). Retraining a keystroke dynamics-based authenticator with impostor patterns. *Computers & Security*, 26(4):300–310.

[Mhenni et al., 2019] Mhenni, A., Cherrier, E., Rosenberger, C., and Essoukri Ben Amara, N. (2019). Analysis of Doddington Zoo Classification for User Dependent Template Update: Application to Keystroke Dynamics Recognition. *Future Generation Computer Systems*.

[Migdal et al., 2017] Migdal, D., Johansen, C., and Jøsang, A. (2017). Offline Trusted Device and Proxy Architecture based on a new TLS Switching technique. In *International Workshop on Secure Internet of Things SIOT 2017 (ESORICS Workshop - A - Core)*, Oslo, Norway.

[Migdal and Jøsang, 2017] Migdal, D. and Jøsang, A. (2017). OffPAD – Objet Personnel d'Authentification Hors-ligne appliqué aux hôpitaux et banques en ligne. In *RESSI 2017*, Grenoble, France.

[Migdal and Rosenberger, 2017] Migdal, D. and Rosenberger, C. (2017). Vers un Code Personnel d'Identité Respectueux de la Vie Privée. In *CORESA*, Caen, France.

[Migdal and Rosenberger, 2018a] Migdal, D. and Rosenberger, C. (2018a). Analysis of Keystroke Dynamics For the Generation of Synthetic Datasets. In *CyberWorlds (B - Core)*, Singapour, Singapore.

[Migdal and Rosenberger, 2018b] Migdal, D. and Rosenberger, C. (2018b). Protection de données personnelles pour la sécurité sur Internet. In *Atelier sur la Protection de la Vie Privée*, Porquerolles, France.

[Migdal and Rosenberger, 2018c] Migdal, D. and Rosenberger, C. (2018c). Towards a Personal Identity Code Respecting Privacy. In *International Conference on Information Systems Security and Privacy (ICISSP)*, Madeira, Portugal.

[Migdal and Rosenberger, 2019a] Migdal, D. and Rosenberger, C. (2019a). Don't listen to my Keystroke Dynamics! (Summer School). 16th Int.l Summer School on Biometrics and Forensics 2019.

[Migdal and Rosenberger, 2019b] Migdal, D. and Rosenberger, C. (2019b). Keystroke Dynamics Anonymization System. In *SeCrypt (B - Core)*, Prague, Czech Republic.

[Migdal and Rosenberger, 2019c] Migdal, D. and Rosenberger, C. (2019c). Keystroke Dynamics Anonymization System. In *SeCrypt*, Prague, Czech Republic.

[Migdal and Rosenberger, 2019d] Migdal, D. and Rosenberger, C. (2019d). My Behavior is my Privacy & Secure Password ! In *Cyberworlds (B - Core)*, Kyoto, Japan.

[Migdal and Rosenberger, 2019e] Migdal, D. and Rosenberger, C. (2019e). Schéma d'Anonymisation de Dynamique de Frappe au Clavier. In *APVP*, Cap Hornu, France.

[Migdal and Rosenberger, 2019f] Migdal, D. and Rosenberger, C. (2019f). Statistical Modeling of Keystroke Dynamics Samples For the Generation of Synthetic Datasets. *Elsevier Journal on Future Generation Computer Systems, Special Issue on CyberSecurity & Biometrics for a better Cyberworld (Q1 - JCR).*

[Monaco, 2018] Monaco, V. (2018). Public keystroke dynamics datasets.

[Monrose and Rubin, 1997] Monrose, F. and Rubin, A. (1997). Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56. ACM.

[Monrose and Rubin, 2000] Monrose, F. and Rubin, A. (2000). Keystroke dynamics as a biometric for authentication. *Future Generation Computer Syststems*, 16(4):351–359.

[Moore and Thorsheim, 2016] Moore, P. and Thorsheim, P. (2016). Keyboard privacy plugin.

[Nikiforakis et al., 2015] Nikiforakis, N., Joosen, W., and Livshits, B. (2015). Privaricator: Deceiving fingerprinters with little white lies. *Proceedings of the 24th International Conference on World Wide Web*, pages 820–830.

[Ninassi et al., 2017] Ninassi, A., Vernois, S., and Rosenberger, C. (2017). Authentification multi-biométrique sur mobile respectueuse de la vie privée. In *CORESA*.

[Nitot, 2016] Nitot, T. (2016). *surveillance://*. C&F éditions.

[Plateaux et al., 2014] Plateaux, A., Lacharme, P., Rosenberger, C., and Josang, A. (2014). Biométrie à usage unique pour la monétique. In *Conférence sur la Sécurité des Architectures Réseaux et des Systèmes d'Information (SAR SSI).*

[Revett et al., 2007a] Revett, K., de Magalhaes, S., and Santos, H. (2007a). On the use of rough sets for user authentication via keystroke dynamics. In *EPIA Workshops*, pages 145–159.

[Revett et al., 2007b] Revett, K., Gorunescu, F., Gorunescu, M., Ene, M., Tenreiro, S. d. M., and Santos, H. M. D. (2007b). A machine learning approach to keystroke dynamics based user authentication. *International Journal of Electronic Security and Digital Forensics*, 1:55–70.

[Saini and Singh, 2018] Saini, P. and Singh, A. K. (2018). Biometric-based authentication in cloud computing. *Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives*, page 147.

[Schneier, 2001] Schneier, B. (2001). Protecting privacy and liberty. *Nature*, 413(6858):773.

[Shen et al., 2013] Shen, C., Cai, Z., Guan, X., Du, Y., and Maxion, R. A. (2013). User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security*, 8(1):16–30.

[Simon, 2005] Simon, B. (2005). The return of panopticism: Supervision, subjection and the new surveillance. *Surveillance & Society*, 3(1).

[Spillane, 1975] Spillane, R. (1975). Keyboard apparatus for personal identification. IBM Technical Disclosure Bulletin.

[Stefan et al., 2012] Stefan, D., Shu, X., and Yao, D. D. (2012). Robustness of keystroke-dynamics based biometrics against synthetic forgeries. *computers & security*, 31(1):109–121.

[Stefan and Yao, 2010] Stefan, D. and Yao, D. (2010). Keystroke-dynamics authentication against synthetic forgeries. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on*, pages 1–8. IEEE.

[Sugden et al., 2000] Sugden, R., Smith, T., and Jones, R. (2000). Cochran's rule for simple random sampling. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):787–793.

[Syed Idrus et al., 2013] Syed Idrus, S., Cherrier, E., Rosenberger, C., and Bours, P. (2013). Soft biometrics database: A benchmark for keystroke dynamics biometric systems. In *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*, pages 1–8.

[Teoh et al., 2004] Teoh, A., Ngo, D., and Goh, A. (2004). Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition*, 40.

[Teoh et al., 2008] Teoh, A. B., Kuan, Y. W., and Lee, S. (2008). Cancellable biometrics and annotations on biohash. *Pattern Recognition*, 41:2034–2044.

[Teoh and Ngo, 2006] Teoh, A. B. and Ngo, D. C. (2006). Biophasor: Token supplemented cancellable biometrics. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–5. IEEE.

[Tian et al., 2018] Tian, Y., Li, Y., Liu, X., Deng, R. H., and Sengupta, B. (2018). Pribioauth: Privacy-preserving biometric-based remote user authentication. In *2018 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–8. IEEE.

[Umphress and Williams, 1985] Umphress, D. and Williams, G. (1985). Identity verification through keyboard characteristics. *Internat. J. Man Machine Studies*, 23:263–273.

[Vastel, 2019] Vastel, A. (2019). *Tracking Versus Security: Investigating the Two Facets of Browser Fingerprinting.* PhD thesis, Université de Lille.

[Weinberg et al., 2011] Weinberg, Z., Chen, E. Y., Jayaraman, P. R., and Jackson, C. (2011). I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks. *Security and Privacy (SP).*

List of Figures

List of Tables

CONTRIBUTIONS TO KEYSTROKE DYNAMICS FOR PRIVACY AND SECURITY ON THE INTERNET

Interactions on the Internet require trust between each involved party. Internet entities assume, at the same time, several roles, each having their own interests and motivations; leading to conflicts that must be addressed to enable security and trust. In this thesis, we use, and focus on, Keystroke Dynamics (the way a user type on its keyboard) in an attempt to solve some of these conflicts.

Keystroke Dynamics is a a costless and transparent biometric modality as it does not require neither additional sensors nor additional actions from the user. Unfortunately, Keystroke Dynamics also enables users profiling (s.a. identification, gender, age), against their knowledge and consent.

In order to protect users privacy, we propose to anonymize Keystroke Dynamics. Still, such information can be legitimately needed by services in order to straighten user authentication. We then propose a Personal Identity Code Respecting Privacy, enabling biometric users authentication without threatening users privacy.

We also propose a Social Proof of Identity enabling to verify claimed identities while respecting user privacy, as well as ensuring users past behaviors through a system of accountability. Generation of synthetic Keystroke Dynamics is also considered to augment existent Keystroke Dynamics datasets, and, in the end, enabling sharing of Keystroke Dynamics datasets without exposing biometric information of real users.

Les interactions requièrent une confiance mutuelle des parties impliquées. Les entités d'Internet endossent plusieurs rôles, chacun ayant ses propres intérêts et motivations; conduisant à des conflits qui doivent être adressés afin de permettre confiance et sécurité. Dans cette thèse nous nous concentrons sur la dynamique de frappe au clavier afin de résoudre quelques de ces conflits.

https://www.overleaf.com/project/5b961d6379823d6602532919La manière de taper au clavier est une modalité biométrique sans coûts et transparente, elle ne requiert ni capteurs ni actions additionnels. Malheureusement, elle permet aussi le profilage des utilisateurs (s.a. identification, âge, sexe), contre leur consentement et connaissance.

Afin de protéger la vie privée des utilisateurs, nous proposons d'anonymiser la dynamique de frappe au clavier. Cependant, cette information peut être légitimement requise afin de renforcer l'authentification des utilisateurs. Nous proposons ainsi un Code Personnel d'Identité Respectueux de la Vie Privée, permettant l'authentification biométrique des utilisateurs, sans menacer leur vie privée.

Nous proposons aussi une preuve sociale d'identité permettant de vérifier des déclarations d'identités ainsi que la génération synthétique de dynamique de frappe au clavier.

Keywords: BIOMETRICS, KEYSTROKE, SECURITY, PRIVACY

ENSICAEN - UNICAEN - CNRS - GREYC UMR 6072, F-14050 Caen, France