NTNU
Norwegian University of
Science and Technology

Trust Management for a Smart Card Based Private eID Manager

Shi Chen

Master of Telematics - Communication Networks and Networked Services (2
Submission date: June 2016
Supervisor: Colin Alexander Boyd, ITEM
Co-supervisor: Bian Yang, Norwegian Information Security Lab, NTNU i Gjøvik

Norwegian University of Science and Technology
Department of Telematics

NTNU – Trondheim
Norwegian University of
Science and Technology

Trust Management for a Smart Card Based Private eID Manager

Shi Chen

Submission date: June 2016
Responsible professor: Colin Alexander Boyd, ITEM
Supervisor: Bian Yang, NISLab

Norwegian University of Science and Technology
Department of Telematics

Abstract

Humans are smart when facing solely technical attacks; they invented technical strategies to defend against technical attacks. However, when facing a social engineering attack, a socio-technical attack, humans become the weakest link of security defense. By exploiting vulnerabilities of trust relationships, social engineers physically and psychologically manipulate victims to gain confidential information and proprietary assets. In spite of the severity and universality of social engineering, unfortunately, there is no better solution but training and educating at present.

When dealing with identity verification in face-to-face interactions, threats from social engineering are particularly serious. Verifying human identity and limits of their authority rely on experience and intuition which is far from accurate. After investigation, vulnerabilities of current identity management solutions are discovered.

By referring to the protocols used in European ePassport, as well as the growing popularity and security properties of smart devices and biometrics, we decide to use smart card, fingerprint, and Near Field Communication (NFC)-enabled smart phone as main technologies of the mechanism. Due to lack of ideal fingerprint smart card, we use fingerprint sensor enabled smart phone – Nexus 5X and programmable Java card for implementation. The tests and evaluation present the availability and possibility to prevent face-to-face social engineering attacks. Future improvements and expectations of the mechanism are also mentioned in the thesis.

Keywords: Social Engineering, Electronic Identity (eID), Smart Card, Biometrics

Preface

This thesis has been written in spring of 2016 in TTM4905 master's thesis project at Norwegian University of Science and Technology (NTNU) under the supervision of professor Colin Alexander Boyd. The project was given by the Norwegian Information Security laboratory (NISLab) at NTNU i Gjøvik and was performed under the supervision of Bian Yang and Qingbao Guo. As a part of research project IDforU[1], this thesis focuses on solving identity verification against social engineering attacks.

I would like to thank my responsible professor Colin Alexander Boyd for much valuable guidance and great support during the work. I really appreciate his vast knowledge and assistance in writing report.

I would also like to thank my supervisor Bian Yang for this great chance to join IDforU team, and guiding me to research social engineering. This is a very interesting research direction, and challenges of this research direction motivated me a lot.

I would also like to thank Qingbao Guo, who is the project manager of IDforU, for his patience and assistance during the design and implementation procedures.

Lastly I would also like to thank all our respondents for committing their time to participate in our questionnaire, as well as volunteers for assisting in tests.

My extended thanks to the Department of Telematics at NTNU and the NISLab at NTNU i Gjøvik for providing me the opportunity to pursue this thesis.

Shi Chen
Gjøvik, Norway

[1] Check Appendix A for more information about IDforU.

Contents

List of Figures

List of Tables

List of Acronyms

AA Active Authentication.

AES Advanced Encryption Standard.

APDU Application Protocol Data Unit.

API Application Program Interface.

APT Advanced Persistent Threat.

APWG Anti-Phishing Working Group.

BAC Basic Access Control.

CA Chip Authentication.

CAN Card Access Number.

CIA confidentiality, integrity, and availability.

CPU Central Processing Unit.

CVCA Country Verifying Certificate Authority.

CZDS Centralized Zone Data System.

DC Document Signer.

DES Data Encryption Standard.

DV Document Verifier.

EAC Extended Access Control.

EAL Evaluation Assurance Level.

EEPROM Electrically Erasable Programmable Read Only Memory.

eID Electronic Identity.

FIDO Fast IDentity Online.

FIPS Federal Information Processing Standard.

G&D Giesecke & Devrient.

GAC Governmental Advisory Committee.

GUI Graphical User Interface.

HiG Høgskolen i Gjøvik.

IC Integrated Circuit.

ICANN Internet Corporation for Assigned Names and Numbers.

ICAO International Civil Aviation Organization.

IDE Integrated Development Environment.

IVR Interactive Voice Response.

IWR Intelligent Word Recognition.

JCDE Java Card Development Environment.

JCDK Java Card Development Kit.

JSON JavaScript Object Notation.

LDS Logical Data Structure.

MFA Multi-factor authentication.

MITM Man-in-the-middle.

MONA Mobile Usage of the New German Identity Card.

MRZ Machine Readable Zone.

NFC Near Field Communication.

NISLab Norwegian Information Security laboratory.

NTNU Norwegian University of Science and Technology.

OCR Optical Character Recognition.

OS Operating System.

PA Passive Authentication.

PACE Password Authenticated Connection Establishment.

PII Personally Identification Information.

PIN Personal Identification Number.

PKI Public Key Infrastructure.

RAM Random Access Memory.

RFID Radio-Frequency Identification.

ROM Read Only Memory.

SAC Supplemental Access Control.

SE Social Engineering.

SFA Single-Factor Authentication.

SHA Secure Hash Algorithm.

SMS Short Messaging Service.

TA Terminal Authentication.

TEE Trusted Execution Environment.

TFA Two-factor authentication.

TLS Transport Layer Security.

Triple DES Triple Data Encryption Standard.

U2F Universal Authentication Framework.

UAF Universal 2nd Factor.

URL Uniform Resource Locator.

VM Virtual Machine.

VoIP Voice over IP.

XSS Cross-site Scripting.

List of Code Listings

Chapter 1
Introduction

1.1 Motivation

We have all watched a magic show, which seems magical and incredible, but it is actually the magician's tricks. Similarly, we have all met superb lies and scams made up by dishonest people, they obtain your trust, then manipulate you to do things you should not do, and the final consequences are normally gaining profits for themselves. They behave as an unscrupulous magician who has you watching his left hand while with his right hand steals your secret [71]. In fact, this is one form of attacks in context of information security, which is Social Engineering (SE), and these dishonest people are named as social engineers.

Obviously, most of us are not experts in debunking scams from micro expressions or behavioral patterns. In addition, the ability of judgment is limited if only depending on knowledge and experience, identifying the true identity and purpose of a stranger is tough enough. However, a research [31] carried out in a close-knit network of people showed that even close friends are sometimes unable to identify each other and even themselves through telephone. Therefore it is difficult for people to detect and prevent social engineering attacks.

In the context of information security, just like technical attacks, the basic goals of social engineering attacks are attempting to obtain sensitive information or unauthorized access. Social engineers manipulate victims for malicious intentions, such as identity theft, property theft, network intrusion, industrial espionage, or system disruption. Social engineering has a very long history and various techniques, a lot of incidents and statistics[1] show that consequences of social engineering attacks are extremely serious and troublesome.

The triad of confidentiality, integrity, and availability (CIA) is at the heart of

[1] More information about social engineering incidents and statistics in Section 2.1 and Section 2.3 respectively.

Figure 1.1: Beauty attack from [24]

information security, and the key to maintain CIA of information of an organization or system is controlling who accesses what information [107]. However, the biggest difference between social engineering attacks and other attacks is the target of social engineering attack is human, which is truly the weakest link of security. Figure 1.1 is a very common form of social engineering attack. It may recall similar experiences which happened to you or people around you.

While scams and tricks are hardly new, the speed and reach of them has been magnified enormously with the increasing dependence on the Internet, email and social media [89]. Unfortunately, there are no better ways to prevent social engineering apart from training and educating at present. This reality motivates us to find and design a solution with popular technologies to fight against social engineering attacks.

Moreover, countermeasures[2] against Internet technology-related social engineering attacks, phishing[3] for instance, have been researched and can be prevented with certain technologies. However, ID checks in face-to-face interactions are still challenging. How to maximize the advantages of advanced technologies in resolving face-to-face social engineering risks is also the motivation of this project.

1.2 Scope and Objectives

As just mentioned, social engineering attacks are happening at various scenes of life and work. Face-to-face social engineering attacks is especially serious due to lack of research and effort. Hence the aim of our project is minimizing the threats of social engineering attacks in face-to-face interactions. Taking advantage of smart devices (smart cards, smart phones, etc.) and biometrics (fingerprint, voice, face, etc), a

[2]More information about countermeasures against social engineering attacks in Section 2.5.
[3]More information about phishing in Section 2.4.1.

mechanism which can minimize face-to-face social engineering risks is designed and implemented to solve this problem to the greatest extent.

There are several forms of social engineering attacks. There are also several scenarios of face-to-face interactions, and the performances of the designed mechanism are various. However, this thesis only covers the following two scenarios:

1. Face-to-face authentication between potential victim (Party A) and a stranger (Party B) who is probably designated to the same transaction (transaction here is a upcoming or in-progress affair such as home repairs services) with the potential victim.

2. Face-to-face authentication between potential victim (Party A) and a stranger (Party B), and there is no transaction between them.

Social engineering attacks target at humans, and unreliable identity verification leads trust relationships existing between victims and social engineers. Thus, identity verification in human authentication becomes the focus of our research. eID is a commonly used method for identity verification like national eID cards, driving licenses, and ePassports[4]. Based on current eID systems, especially the third generation specification of European ePassport, we proposed a specific mechanism for migrating social engineering risks in the two scenarios we mentioned above. To simply verify our design, we developed a prototype in lab environment by using technologies like smart cards, fingerprints, smart phones, etc.

The full implementation of the mechanism needs support from national and government organizations, so it is impossible to realize in the project duration. Therefore the implementation of this mechanism only focuses on the face-to-face authentication procedure.

This project is one research direction of project IDforU[5]. IDforU is a research project in the NISLab at NTNU i Gjøvik, it is funded by *Regional Research Funds in Norway* [30]. Due to the increasing volume of eID usage, human factors leads eIDs become one of the weakest links in security management. Mismanagement of eIDs can have serious consequences. In order to fulfill end users' expectations, improve efficiency and trustworthiness of eID management, and integrate features like user definability and trust management. Hence, the main purpose of IDforU is analyzing user demands and proposing a future eID management architecture for Norwegian end users.

[4]More information about Electronic Identity (eID) at Section 3.2.
[5]Check Appendix A for more information about IDforU.

1.3 Ethical Considerations

The personal information (basic information and biometric data) of volunteers who joined the questionnaire and implementation tests are confidential and only used in this project.

1.4 Outline

This report is divided into ten chapters.

In Chapter 1, we introduce the motivation of this project, the scope and objectives, as well as the ethical considerations.

In Chapter 2, we list some famous and representative social engineering attacks in history. By means of introducing social engineering techniques as well as statistical data, we reveal the mystery of social engineering. The main countermeasures against social engineering attacks are normally educational training.

In Chapter 3, we introduce trust management by recalling social engineering incident introduced in the Chapter 2, revealing the trust relationship existing between victims and social engineers. To make this term easier to understand, we introduce three commonly used eID systems to let readers know how the trust is managed at present. Through the development of national eID card, the advantages of smart card technology are shown. By introducing the evolution of three generations of European ePassports, the main security mechanisms give us a lot of inspiration for our design.

In Chapter 4, we propose a mechanism which can minimize social engineering risks. By assuming five common scenarios in our daily life where social engineering attacks may occur, we list out the requirements for the design. In order to meet the requirements, we decide to exploit technologies smart card (Java Card technology), biometrics (match-on-card fingerprint verification), and smart phones. We design a biometric eID card infrastructure includes issue, update, and revocation of biometric eID cards. For the two main face-to-face authentication scenarios, we design different procedure under both conditions, when there is a pre-assigned transaction between both parties (Party A and Party B). The description of the transaction will be displayed after the authentication procedure, otherwise, a new transaction will be created and stored. A critical step before the authentication is to establish a secure communication channel between a eID card and a terminal, as well as to verify the authenticity of both the card and the terminal. We propose a new protocol by referring to Diffie–Hellman key exchange, Password Authenticated Connection Establishment (PACE), Terminal Authentication (TA), Chip Authentication (CA). In the end, we recall the scenarios

again to check whether the design can minimize social engineering risks in those scenes or not.

In Chapter 5, we introduce three previous projects, the basis of our project – *Smart Identity Card*, the fingerprint smart card – Zwipe Access, and client of German eID card – Mona. We have learn a lot experience from these projects, and find some inspiration as well as shortcomings. Even though these previous work have made great achievement, our project has overcome some limitations of these previous work.

In Chapter 6, because of the unavailability of ideal fingerprint Java card, we use Nexus 5X, smart cards and a smart card reader to complete the implementation. The hardware and software used for implementation are introduced, as well as the implementation procedure from environment setup to final face-to-face demonstration. The performance is also introduced after several tests in the lab environment.

In Chapter 7, we analyze security properties of the mechanism, including security goals, security measures, as well as different potential threats to smart cards, biometrics, and channels.

In Chapter 8, we conclude the achievements and limitations of this project. In addition, we also present further development of this project.

Chapter 2
Social Engineering

In the vast field of information security, hackers are hunting all possibilities to achieve misconduct in every dark corner. After decades of fighting against hackers, people are no longer shocked when hearing or reading some common attacks happened. In addition, with the rapid development of information technology and security education, numerous solutions are emerging to detect and prevent cyber attacks effectively.

However, there is a unique attack in the context of information security, which is social engineering. To a certain extent, social engineering sounds like a political science rather than an attack. In fact, instead of using programming skills or hacker tools as weapons, social engineers using confidence trick tactics to attack the weakest link of information security – the human [71].

This chapter reveal the truth about social engineering. After reading through this chapter, readers may find resonance with victims in mentioned incidents, and sudden realize that social engineering are everywhere and difficult to prevent.

2.1 Incidents

In brief, social engineering[1] refers to psychological manipulation of people into divulging confidential information even performing actions they should not supposed to perform.

2.1.1 Trojan Horse

In retrospect, social engineers have been duping victims dating back to ancient times without network, even as early as the very beginning of human existence [37]. One classic and famous story is Trojan Horse as shown in Figure 2.1, which is the key of

[1] Wikipedia, "Social Engineering (Security)", https://en.wikipedia.org/wiki/Social_engineering_(security) [Online; Accessed 09 Mar 2016]

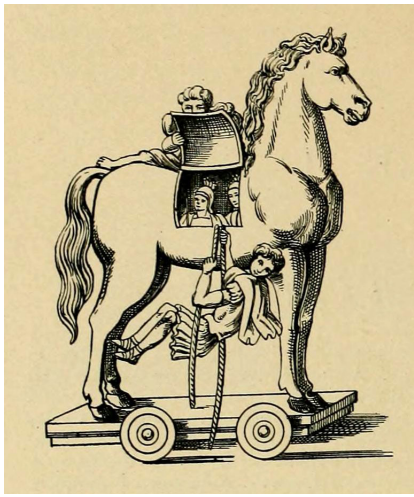


Figure 2.1: Trojan horse from Wikipedia

success in Trojan War [90, 37]. After an exhausting but unsuccessful 10-year siege of Troy, the leader of the Greek army, Ulysses, engineered the legendary Trojan Horse scheme – a "gift" with hidden Greek soldiers inside the horse, which lead to the fall of Troy.

The Trojan Horse was a just a tale until Heinrich Schliemann[2], a pioneer in the field of archaeology, excavated the historical Troy. Mythical or not, Trojan Horse is a representative example of social engineering attack.

As most people know, Trojan Horse or Trojan[3] is also used to name an class of malicious computer program. Just like Ulysses's tricks, Trojan misrepresents itself to appear tantamount to normal programs, it seems routine, useful or interesting in order to persuade victims to install them, which exploits exactly forms of social engineering attacks.

2.1.2 Eiffel Tower Scam

Victor Lustig[4] was well known as "the man who sold the Eiffel Tower twice" [37]. After the recovery period of France from World War I, Victor Lustig read from newspaper about the expansive cost of maintaining Eiffel Tower and saw the possibilities of

[2]Wikipedia, "Heinrich Schliemann", https://en.wikipedia.org/wiki/Heinrich_Schliemann [Online; Accessed 10 Mar 2016]

[3]Wikipedia, "Trojan Horse (Computing)", https://en.wikipedia.org/wiki/Trojan_horse_(computing) [Online; Accessed 10 Mar 2016]

[4]Wikipedia, "Victor Lustig", https://en.wikipedia.org/wiki/Victor_Lustig [Online; Accessed 10 Mar 2016]

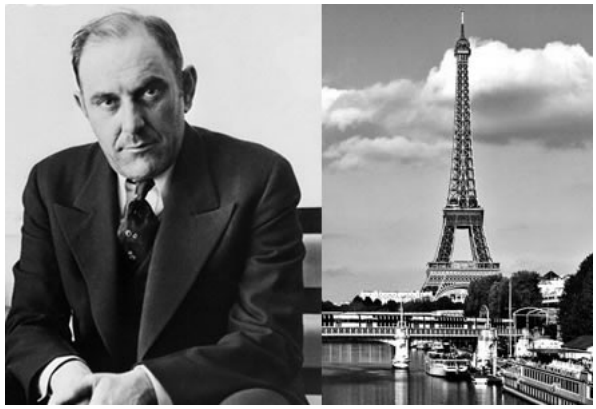


Figure 2.2: Eiffel tower scam from [62]

making a profit. He then disguised as the deputy director-general of the Ministry of Posts and Telegraphs using fake government stationery, he invited six scrap metal dealers to the "secret meeting" about selling Eiffel Tower for scrap. Through this scam, Victor Lustig received not only funds for the Eiffel Tower but also a large bribe. After tasted honey of selling the Eiffel Tower, before he got arrested, he repeated the scam just one month after the first scam and succeeded again.

Victor Lustig was a typical social engineer with seemingly authoritative fake identity and rhetorical lies. In history, there were some other similar famous social engineers, such as Ferdinand Waldo Demara. Jr[5], Frank Abagnale[6], Anna Anderson[7], and Lambert Simnel[8] [32].

2.1.3 Military Scams

Apart from social engineers like Victor Lustig launching social engineering attacks for personal wealth or reputation, just like [19] wrote, nothing is worse than the fake war heroes – social engineers who want all of the glory and cool stories of people who served without the actual "risking their lives" part.

It sounds ridiculous of faking war heroes, however, there were no lack of challengers

[5]Wikipedia, "Ferdinand Waldo Demara", https://en.wikipedia.org/wiki/Ferdinand_Waldo_Demara [Online; Accessed 16 Mar 2016]

[6]Wikipedia, "Frank Abagnale", https://en.wikipedia.org/wiki/Frank_Abagnale [Online; Accessed 16 Mar 2016]

[7]Wikipedia, "Anna Anderson", https://en.wikipedia.org/wiki/Anna_Anderson [Online; Accessed 16 Mar 2016]

[8]Wikipedia, "Lambert Simnel", https://en.wikipedia.org/wiki/Lambert_Simnel [Online; Accessed 16 Mar 2016]

Figure 2.3: Wilhelm Voigt sculpture from Wikipedia

who attempted and succeed. In the beginning of 19th century, Wilhelm Voigt[9] masqueraded as a Prussian military officer, he not only "confiscated" more than 4,000 marks[10] from the entire town but also rounded up a number of soldiers under his "command". Than he just disappeared with all the cash. The more ridiculous follow-up story about Wilhelm Voigt was that he was pardoned for his crime, and there is a sculpture of him as the Captain of Köpenick at Köpenick city hall, see Figure 2.3.

Stanley Clifford Weyman[11] impersonated various military officers for free food and great treatment, in 1921, he even met USA president in the White House. Douglas R. Stringfellow faked paralysis and got elected to be a one-term congressman in the United States House of Representatives[12], Figure 2.4 shows Douglas R. Stringfellow (the man on the right side in the phone) was talking in wheelchair with Hugh Richens, a real paralyzed soilder from World War II, in that time he was already elected as a congressman.

In addition, there are more examples of military scams, readers can read more about fake private army commander – David Deng in [19], fake counter-terrorism

[9]Wikipedia, "Wilhelm Voigt", https://en.wikipedia.org/wiki/Wilhelm_Voigt [Online; Accessed 16 Mar 2016]

[10]Mark was the official currency of Germany, but was already replaced by the Euro in 1999.

[11]Wikipedia, "Stanley Clifford Weyman", https://en.wikipedia.org/wiki/Stanley_Clifford_Weyman [Online; Accessed 16 Mar 2016]

[12]Wikipedia, "Douglas R. Stringfellow", https://en.wikipedia.org/wiki/Douglas_R._Stringfellow [Online; Accessed 16 Mar 2016]

Figure 2.4: Hugh Richens and Douglas R. Stringfellow from [64]

expert William "Bill" Hillar in [19], and fake “Green Beret” in I-40 bridge disaster[13] – William James Clark in [19].

As bystanders rather than victims, it is hard to believe how can these social engineers succeeded. The truth is a fancy-looking military uniform can trumps any degree, job title, or letter of recommendation you could possibly get [19].

2.1.4 RSA SecurID Breach

Unlike other incidents introduced in this section, RSA SecurID Breach was not a completely non-technical attack. Reasons made this incident notable is RSA[14] is an American computer and network security company, and RSA SecurID is a two-factor authentication token produced by RSA for a user to a network resource.

On 18 March 2011, Art Coviello, the Executive Chairman of RSA wrote an open letter to RSA customers [18] and revealed that RSA SecurID was targeted by “an extremely sophisticated cyber attack”, the internal RSA staff phished successfully, which leading to the master keys for all RSA SecurID as shown in Figure 2.5 being stolen, then subsequently used to break into US defense suppliers [108]. Art Coviello claimed this attack was in the category of an Advanced Persistent Threat (APT), which is a set of stealthy and continuous computer hacking processes. The first phase of an APT attack, or the main method attackers used to gain foothold on the system more precisely, is social engineering attack [21, 95]. Owing to the use of

[13]Wikipedia, “I-40 Bridge Disaster”, https://en.wikipedia.org/wiki/I-40_bridge_disaster [Online; Accessed 16 Mar 2016]

[14]RSA Security LLC,[5] formerly RSA Security, Inc. and doing business as RSA. Official website: https://www.rsa.com [Online; Accessed 08 Apr 2016]

Figure 2.5: RSA SecurID from Wikipedia

social engineering techniques combined with multiple techniques such as zero-day exploits[15], APT is much harder to defend against [105].

The breach cost EMC[16], the parent company of RSA, $66.3 million. This incident was one of the first high-profile attacks against a security company, it impacted not only the security of RSA SecurID but also thousands of other organizations [90].

2.2 Social Engineering Around Us

In addition to those famous incidents and scams we mentioned above, think carefully, social engineering attacks are not just stories you read from news or only happened to some unlucky victims. Social engineers are around us, a variety of social engineering attacks are happening everyday. Have you ever been stopped in the street by strangers who introduce their program and hope you can fill a form with your personal information to support them?

Have you ever received phone calls from strange callers but they claimed to be bank consultants, property consultants, telecom staff, or any sounds credible job titles? Have you ever received emails from seems credible people with super attractive content and attached file or link? Have you ever happen with strange salesman who knocked the door and hoped to get in to introduce their products or services? Have you ever been lead to unknown e-commerce website or video-sharing website because of clicking attractive advertising when visiting other website?

[15]Zero-day (also known as zero-hour, 0-day) exploits means that hackers exploit undisclosed vulnerabilities to attack network or system.

[16]An American multinational corporation. Official website: http://www.emc.com [Online; Accessed 08 Apr 2016]

Social engineering attacks are quite different from other technical attacks, social engineering attacks are related to all aspects of our daily lives. By using influence, persuasion and strong social skills, social engineers can manipulate people without causing their attention.

However, there are some repentant social engineers who decided use their knowledge and experience of social engineering to teach and help people. Kevin Mitnick[17], the author of book "The Art of Deception: Controlling the Human Element of Security" [71], was a famous social engineer and did various computer and communications-related crimes. Christopher Hadnagy[18] is a professional social engineer, he wrote books "Social Engineering: The Art of Human Hacking" [43], "Unmasking the Social Engineer: The Human Element of Security" [44] and "Phishing Dark Waters: The Offensive and Defensive Sides of Malicious Emails" [45]. Besides, Christopher Hadnagy is the Chief Human Hacker of Social-Engineer, Inc[19].

Maybe you are so lucky that you never be trapped by social engineering attacks, but social engineering is also popular in films. Here we list out some which may give readers clearer impressions about social engineering around us.

1. In the American biographical crime film *Catch Me If You Can*[20], a social engineer Frank Abagnale got money by confidence scams at his teenage years. When he grow up, he even impersonated an airline pilot and succeeded in stealing over $2.8 million by forging Pan Am payroll checks. Despite of fake identity of an airline pilot, he also impersonated a teaching assistant, a doctor, a lawyer and so on.

2. In the American comedy film *Dirty Rotten Scoundrels*[21], a British social engineer Lawrence Jamieson steals money from wealthy and morally suspect women by seducing them, while an American social engineer Freddy Benson impersonates a wounded soldier in a wheelchair and swindles money from female victims.

3. In the American crime film *The Thomas Crown Affair*[22], a social engineer only has amusement purposes, Thomas Crown, exploits social engineering tactics

[17]Wikipedia, "Kevin Mitnick", https://en.wikipedia.org/wiki/Kevin_Mitnick [Online; Accessed 22 Mar 2016]

[18]Wikipedia, "Christopher Hadnagy", https://en.wikipedia.org/wiki/Christopher_Hadnagy [Online; Accessed 22 Mar 2016]

[19]Official website: https://www.social-engineer.com [Online; Accessed 22 Mar 2016]

[20]Wikipedia, "Catch Me If You Can", https://en.wikipedia.org/wiki/Catch_Me_If_You_Can [Online; Accessed 06 May 2016]

[21]Wikipedia, "Dirty Rotten Scoundrels (film)", https://en.wikipedia.org/wiki/Dirty_Rotten_Scoundrels_(film) [Online; Accessed 06 May 2016]

[22]Wikipedia, "The Thomas Crown Affair (1999 film)", https://en.wikipedia.org/wiki/The_Thomas_Crown_Affair_(1999_film) [Online; Accessed 06 May 2016]

with various distractions to steal the painting "San Giorgio Maggiore at Dusk" by Monet, which worth $100 million.

4. In the American comedy film *Six Degrees of Separation*[23], a skillful social engineer Paul claims to be a close college friend of Kittredges' children, by using social engineering tactics Pual succeed in winning their trust and conned money from Kittredges.

5. In the American caper film *Sneakers*[24], a social engineer Martin Bishop leads a team of security specialists, in one scene, one of Martin's teammate impersonated a pizza delivery person to distract a security guard and let Martin get pass.

6. In the American crime film *Matchstick Men*[25], a social engineer Roy Waller operates a fake lottery with his friend Frank Mercer, they also sell overpriced water filtration systems to unsuspecting customers, they even target an arrogant businessman Chuck Frechette with a long-term scam.

7. In the American crime comedy film *Identity Thief*[26], a social engineer Diana used pretexting to get personal information of accountant Sandy Patterson over phone then stole his identity.

8. In the American caper film *The StingMatchstick Men*[27], a social engineer Henry Gondorff helps another social engineer Johnny Hooker and recruits many other social engineers, the group faked FBI agent, and a phony off-track betting parlor and succeeded in swindling $500,000 from vicious crime boss Doyle Lonnegan.

9. In the American crime techno-thriller film *Hackers*[28], a hacker and social engineer Dade Murphy impersonates an accountant, an important executive, a delivery worker and so on in order to steal information or bypass security.

10. In the American comedy film *Paper Moon*[29], a social engineer Moses Pray impersonates a Bible salesman and targets recently widowed women, he swindles

[23]Wikipedia, "Six Degrees of Separation (film)", https://en.wikipedia.org/wiki/Six_Degrees_of_Separation_(film) [Online; Accessed 06 May 2016]

[24]Wikipedia, "Sneakers (1992 film)", https://en.wikipedia.org/wiki/Sneakers_(1992_film) [Online; Accessed 06 May 2016]

[25]Wikipedia, "Matchstick Men", https://en.wikipedia.org/wiki/Matchstick_Men [Online; Accessed 06 May 2016]

[26]Wikipedia, "Identity Thief", https://en.wikipedia.org/wiki/Identity_Thief [Online; Accessed 14 Apr 2016]

[27]Wikipedia, "The Sting", https://en.wikipedia.org/wiki/The_Sting [Online; Accessed 06 May 2016]

[28]Wikipedia, "Hackers (film)", https://en.wikipedia.org/wiki/Hackers_(film) [Online; Accessed 06 May 2016]

[29]Wikipedia, "Paper Moon (film)", https://en.wikipedia.org/wiki/Paper_Moon_(film) [Online; Accessed 06 May 2016]

money from them by deceiving them that their deceased husband had recently purchased an expensive, personalized Bible from him.

2.3 Statistics

Most of us are familiar with technology-based security attacks, and no matter in organizations or daily lives, there are various tools and processes in place to help protect sensitive data from technology-based security attacks.

When analyzing an attack, it is always interesting and necessary to figure out what exactly resulting in the start point of an incident, even a technology-based security attack. In other words, how did attackers gain the initial access to the target environment? According to [67], attackers typically gain initial access with a blend of social engineering and unpatched (or unknown) vulnerabilities. However, in some cases, how the attackers were able to access and steal data is unclear from evidence if they use social engineering knowledge, which implies that social engineering attacks are more challenging to manage.

According to a survey [94] of 853 IT professionals conducted by Checkpoint in the United States, United Kingdom, Canada, Australia, New Zealand, and Germany during July and August 2011, there are some findings we want to list:

1. Social engineering attacks cost of over $25,000 to over $100,000 per incident in organizations, check detailed statistics in Figure 2.6.
2. 48% of large companies and 32% of companies of all sizes have experienced 25 or more social engineering attacks in the past two years.
3. 51% of social engineering attacks are motivated by financial gain.
4. Most organizations lack of proactive training to prevent social engineering attacks, only 26% of respondents do ongoing training.

Another investigation conducted on 44 respondents/organizations represent over 100,000 employees in Norway from a master thesis of Høgskolen i Gjøvik (HiG) [47] shows that successful social engineering attacks result is losses of several millions Norwegian Krone[30]. In addition, most organizations do not treat social engineering as a high risky attack, but most organizations have taken precautions such as awareness training as well as technical measures.

In regarding with respective detailed statistics of different social engineering techniques, there are more statistics in the following section.

[30] Norwegian Krone is the currency of Norway and its dependent territories.

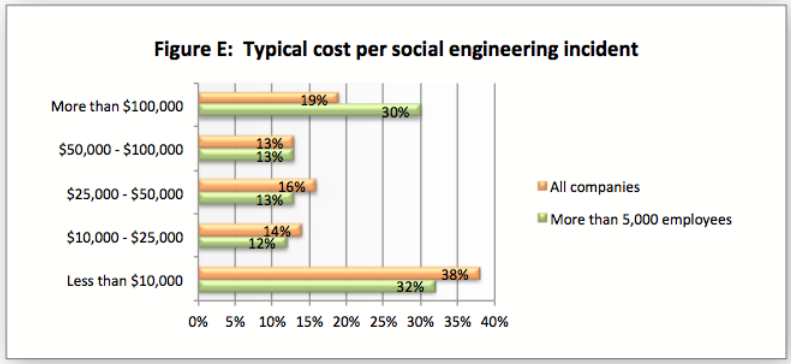


Figure 2.6: Typical cost per social engineering incident from [94]

2.4 Social Engineering Techniques

When it comes to the basic goals of social engineering, we alreay know from Section 2.3 that financial gain is the principal goal. Apart from this, just the same as hacking in general, the basic goals of social engineering attacks is to gain unauthorized access to systems or information for malicious purposes such as commit fraud, network intrusion, industrial espionage, identity theft, or simply to disrupt the system or network [39].

How social engineering attacks work? What exactly skills or tricks are social engineers playing? These are still interesting to find out. Generally, social engineering attacks take place on both physical level and psychological level. In this section, we introduces several popular social engineering techniques.

2.4.1 Phishing

Phishing[31], a neologism created as a homophone of fishing. Fishing means using a bait in attempt to catch a fish while phishing attempts to catch a victim with a "bait" – masquerading as a trustworthy entity in an electronic communication [55]. Figure 2.7 shows an interesting definition of phishing. Through phishing, social engineers can fraudulently acquire sensitive information from a victim. According to [94] , phishing is most common source of social engineering attacks (%47).

[31] Wikipedia, "Phishing", https://en.wikipedia.org/wiki/Phishing [Online; Accessed 06 Apr 2016]

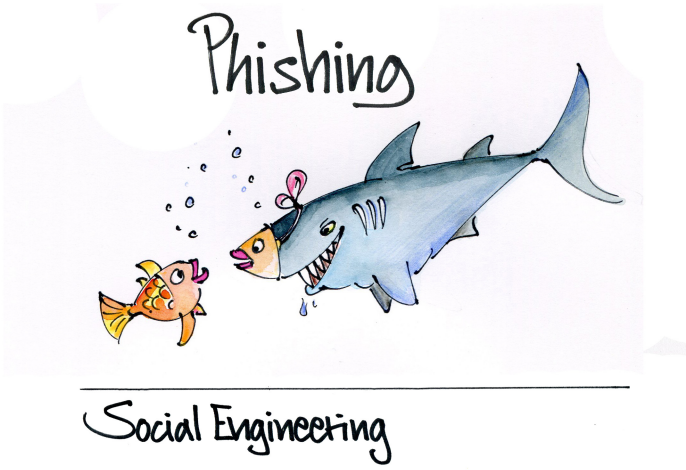


Figure 2.7: Phishing comic from [61]

2.4.1.1 Phishing Statistics

In Section 2.1.4, we introduced the notable social engineering incident – RSA SecurID Breach, a successful email phishing attack is the fuse of the great loss. Apart from this, in November 2013, 110 million customer and credit card records of Target, the second-largest discount retailer of United States, were stolen through a phished subcontractor account [79], the CEO and IT security staff of Target got fired because of this attack. In September 2014, customer and credit card data of over 100 million shoppers of all 2200 Home Depot, an retailer of home improvement and construction products and services in United States, posted for sale on hacking web sites, similarly, it was caused by a phished vendor account [113]. In November 2014, Internet Corporation for Assigned Names and Numbers (ICANN)[32], a non-profit public-benefit organization dedicated to ensuring stable and secure of network, was targeted by spear phishing attack and the administrative access to the Centralized Zone Data System (CZDS)[33] was gained, which lead to zone files as well as credentials and real information of users were stolen. In addition, the access to ICANN Governmental Advisory Committee (GAC) wiki[34], ICANN Blog[35], and ICANN WHOIS[36] were also gained by attackers [51].

[32]Official website: https://www.icann.org [Online; Accessed 08 Apr 2016]
[33]Official website: https://czds.icann.org [Online; Accessed 08 Apr 2016]
[34]Official website: https://gacweb.icann.org [Online; Accessed 08 Apr 2016]
[35]Official website: https://blog.icann.org [Online; Accessed 08 Apr 2016]
[36]Official website: https://whois.icann.org [Online; Accessed 08 Apr 2016]

Table 2.1 shows the total number of unique phishing reports (campaigns) received by Anti-Phishing Working Group (APWG) from customers in the past eleven years [7]. It implies that the amount of phishing attacks are increasing rapidly in the past eleven years.

Table 2.1: Total number of unique phishing reports (campaigns) received

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Sum
2005	12845	13468	12883	14411	14987	15050	14135	13776	13562	15820	16882	15244	**173063**
2006	17877	17163	18480	17490	20109	28571	23670	26150	22136	26877	25816	23787	**268126**
2007	29930	23610	24853	23656	23415	28888	23917	25624	38514	31650	28074	25683	**327814**
2008	29284	30716	25630	24924	23762	28151	24007	33928	33261	34758	24357	23187	**335965**
2009	34588	31298	30125	35287	37165	35918	34683	40621	40066	33254	30490	28897	**412392**
2010	29499	26909	30577	24664	26781	33617	26353	25273	22188	23619	23017	21020	**313517**
2011	23535	25018	26402	20908	22195	22273	24129	23327	18388	19606	25685	32979	**284445**
2012	25444	30237	29762	25850	33464	24811	30955	21751	21684	23365	24563	28195	**320081**
2013	28850	25385	19892	20086	18297	38100	61453	61792	56767	55241	53047	52489	**491399**
2014	53984	56883	60925	57733	60809	53259	55282	54390	53661	68270	66217	62765	**704178**
2015	49608	55795	115808	142099	149616	125757	142155	146439	106421	**194499**	105233	80548	**1413978**

It is clear to see from this table that the total number of yearly reported phishing attacks kept increasing except year 2010 to year 2012. In addition, Figure 2.8 shows that the phishing reports quantity in year 2015 (**1413978**) was nearly ten times of this number back in year 2005 (**157819**), and in Figure 2.9 the quantity doubled in year 2015 (**1413978**) than year 2014 (**704178**).

With the rapid development of Internet technologies as well as information security technologies, the risks of phishing attacks also increase rapidly.

2.4.1.2 Phishing Types

There are several types of phishing:

1. **Spear Phishing** is a type of phishing attack which directed at specific individuals or organizations. Different from traditional phishing attacks, spear phishing is much more complex, time-consuming but offers significantly more financial gain [89]. Social engineers may gather personal information about their target to increase their probability of success. Spear Phishing is the most successful social engineering attack on the internet today, accounting for 91% of cyber attacks, there are more detailed statistics in [103].

2. **Clone Phishing** is a type of phishing attack which cloned a legitimate and previously delivered email but containing malicious attachment or link. Due to social engineers use an spoofed email address to appear as the original sender, it may claim to be a resend of the original email or an updated version of the original email. Clone Phishing is also high risky, it can be used to gain a foothold on a machine through a previously infected machine.

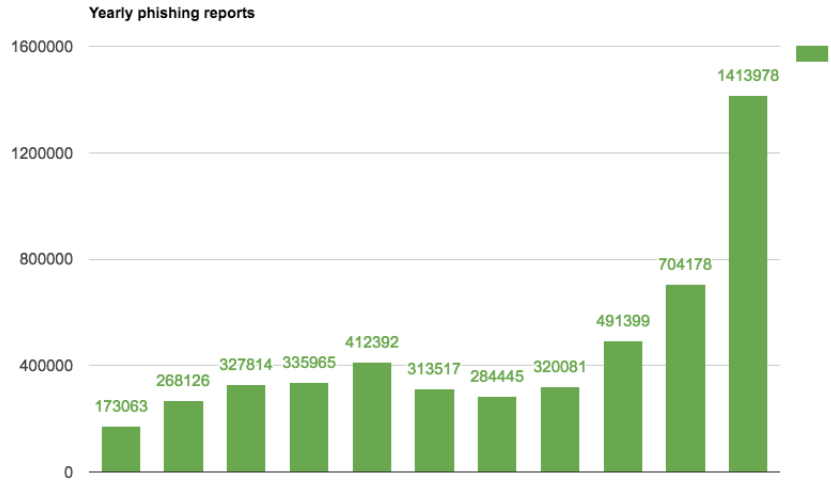


Figure 2.8: Phishing attacks reported between year 2005 to 2015

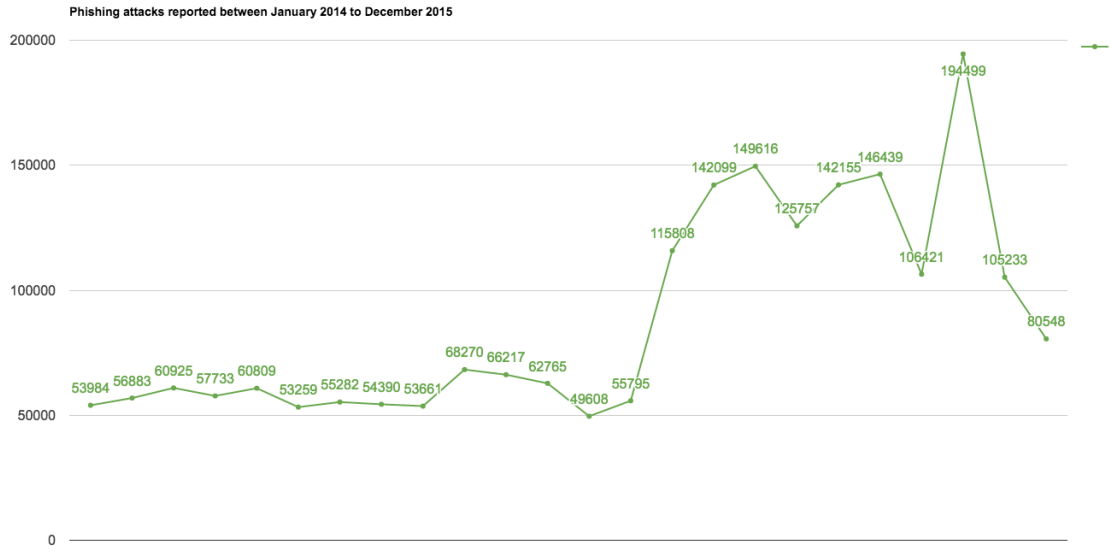


Figure 2.9: Phishing attacks reported between January 2014 to December 2015

3. **Whaling** is a type of phishing attack which targets specifically at victims who have high positions or profile within organizations. Social engineers masquerade whaling emails as critical business emails sent from a legitimate business authority. The content of whaling emails normally take a more serious form such as customer complaints or executive issues. Whaling targets at the upper management of organizations and lead to a lot of concern.

2.4.1.3 Phishing Techniques

Obviously, phishing is the most common technique of social engineering attacks and cause serious consequences. Why phishing works? Lack of knowledge, visual deception, and bounded attention are reasons discovered in [22]. How phishing works? In this section, we will introduce popular techniques used in phishing attacks.

Link Manipulation

Most of phishing emails contain malicious attachment or link, link manipulation is the trick which social engineers playing here with the malicious link. This link appears as usual Uniform Resource Locator (URL) but it will lead victims to spoofed websites belong to the spoofed organizations, social engineers achieve their goals by common tricks such as misspelled URLs, the use of sub-domains, or make the displayed link suggest a reliable destination while it actually goes to the malicious site (such as the code below does).

```
1 <a href="http://malicious.com">http://legitimate.com</a>
```

Filter Evasion

To detect commonly used text in phishing emails, there are some anti-phishing filters doing this task. Anti-phishing filters use techniques such as Optical Character Recognition (OCR) and Intelligent Word Recognition (IWR) to optically scan text used in the emails. However, social engineers started using multimedia such as images with hidden text instead of pure text to make it harder to detect.

Website Forgery

Making victims click the malicious link and visit the phishing website is not the end of a phishing attack. Some phishing scams use JavaScript[37] commands in order to alter the address bar by placing a picture of a legitimate URL, or by closing the original bar and opening up a new one with the legitimate URL [72].

Cross-site Scripting (XSS) is a type of attack which social engineers use to steal victims' credentials, XSS exploits flows in trusted websites' own scripts, victims

[37] JavaScript is a front-end programming language.

are much easier to be attacked because everything in the website appears correct including website address and security certificates. In 2006, Paypal[38] has been attacked because of such security flaws and lead to identity theft [75].

In 2007, a universal Man-in-the-middle (MITM) phishing kit [48] which provides a simple-to-use interface to convincingly reproduce websites and capture log-in credentials that victims entered at the fake site. Tools like this allows social engineers to attack victims by phishing much more easily.

Covert Redirect

Covert redirect is a technique of phishing attacks which makes links appear legitimate but actually redirect victims to a malicious website. Different from other phishing techniques such as link manipulation we introduced above, in which malicious URL is usually different from the legitimate URL, covert redirect is usually masqueraded under a log-in pop-up based on an affected site's domain, so it is difficult to spot. Covert redirect is a notable security flaw and a threat to the Internet. In 2014, covert redirect vulnerability were found both in log-in tools OAuth[39] and OpenID[40] [15, 106], which raised serious debate and attention on covert redirect. After the founder and interim CEO at WhiteHat Security[41], Jeremiah Grossman, looking at the covert redirect vulnerability in OAuth and OpenID, he said following words which can reveals the serious consequence of covert redirect flaws.

> *"While I can't be 100 percent certain, I could have sworn I've seen a report of a very similar if not identical vulnerability in OAuth. It would appear this issue is essentially a known WONTFIX[42]. This is to say, it's not easy to fix, and any effective remedies would negatively impact the user experience. Just another example that Web security is fundamentally broken and the powers that be have little incentive to address the inherent flaws."*

2.4.2 Vishing

As phishing becomes more prevalent, users are trained to not click the obfuscated link, however, social engineering attacks has moved seamlessly back and forth between email to one of our most trusted utilities, the telephone system [40]. Vishing is the social engineering attack happened in the telephone system.

[38] Wikipedia, "PayPal", https://en.wikipedia.org/wiki/PayPal [Online; Accessed 21 Apr 2016]
[39] Wikipedia, "OAuth", https://en.wikipedia.org/wiki/OAuth [Online; Accessed 21 Apr 2016]
[40] Wikipedia, "OpenID", https://en.wikipedia.org/wiki/OpenID [Online; Accessed 21 Apr 2016]
[41] WhiteHat Security is a company responsible for securing web applications. Official website: https://www.whitehatsec.com/ [Online; Accessed 24 May 2016]
[42] WONTFIX refers to "won't fix", it means such issue can be fixed, but it's not worth spending development time on it.

Vishing is also known as voice phishing or phone phishing, this word is a combination of voice and phishing, which is a social engineering attack over the telephone system to gain sensitive information from victims. Vishing takes advantage of the public's trust in telephone services and utilizes convenience of Voice over IP (VoIP) such as caller ID spoofing and Interactive Voice Response (IVR), which enables social engineers to dislay any number on recipient's phone and win victim's trust easily through telephone services. Social engineers usually masquerade phone calls as from bank or similar organizations, the purpose of vishing are normally identity theft and financial fraud.

In addition, vishing is difficult to monitor or trace because of the feature of VoIP, which also makes vishing a serious and somewhat successful social engineering attack[43].

2.4.3 Smishing

Smishing[44] is a combination of Short Messaging Service (SMS) and phishing. Simliar to vishing, smishing also exploits public's trust in telephone services to acquire sensitive information by masquerading as a trustworthy entity. Comparing to phishing and vishing, there are less smishing attacks [111]. On March 9, 2012, Walmart[45] issued a fraud alert regarding a large number of scam SMSs that offered a nonexistent $1000 gift card as bait, this was a typical smishing attack.

2.4.4 Pretexting

In breif, pretexting is a trick to pretend as someone else. By using an elaborate lie or an invented scenario, social engineers increase the chance of success in social engineering attacks. Pretexting is never a easy technique to dress up and make up to perform another person's character [81]. To establish legitimacy in the mind of the targeted victim, a social engineer needs to do some prior research about the background, personal information, and character of the victim. In some situations, social engineers can even pretend to be a new identity which the victim may not familiar with.

Social engineers can impersonate bank staff, tax officers, co-workers, policemen, insurance investigators — or any other individual who seems reliably and have right to know sensitive information from victim. Victims may reveal personal information and some other sensitive information when attacked by pretexting.

[43] Wikipedia, "Voice Phishing", https://en.wikipedia.org/wiki/Voice_phishing [Online; Accessed 14 Apr 2016]

[44] Wikipedia, "SMS Phishing", https://en.wikipedia.org/wiki/SMS_phishing [Online; Accessed 14 Apr 2016]

[45] Walmart is an American multinational retail corporation that operates a chain of hypermarkets, discount department stores and grocery stores.

2.4.5 Baiting

Book [43] states that baiting as "an in-person attack where access is gained to the target's building or other property by some method, and USBs or DVDs are dropped that contain malicious files on them embedded with malicious code". It means that baiting uses physical media, which is like a real-world Trojan Horse, and relies on the curiosity or greed of the victim [53]. USB flash drive, CD-ROM, or Floppy disk are common forms of the physical media which social engineers used as a bait, these physical media are normally infected by malware and left at obvious location (e.g. parking lot, bathroom, elevator, sidewalk). When curious victims pick them up and insert them into computers to see contents, the malware is installed on the computers quietly.

2.4.6 Dumpster Diving

Most of us throw many papers into trash bin and seldom treat this behaviour seriously, sometime some papers with personal information are throw unconsciously. The personal information on the paper can help social engineers a lot if they get the papers. In the context of information security, searching confidential information from a dumpster is called dumpster diving [52]. Dumpster diving can offer a quick way for social engineers to find all the useful information they want. Figure 2.2 shows the what information social engineers can collect through dumpster diving, as well as what they could do with victims' information.

Table 2.2: Information can be collected through dumpster diving, and the consequences it can cause. Based on a table in [112].

Information can be collected	**Consequences**
• Pre-approved credit card offers • Street address • Personal number • Telephone number • Email address • Bank account information • Employment history • Other personal information	• Identity theft • Employment-related fraud • Loan fraud/payday loan fraud • Bank fraud • Benefits fraud • Tax fraud • Other identity fraud

2.5 Countermeasures

In a information age, whether social engineering attacks will attack victims is no longer a puzzled question, the answer is affirmative. The question really troubles

victims is when social engineering attacks will happen? Finding effective countermeasures has become a urgent and important mission. Currently, there exist several countermeasures against social engineering attacks, but they have the common characteristics of precautionary technologies [3].

Educational training is the most common countermeasure against social engineering attacks. For organizations, which are the places that social engineering attacks happen most frequently, getting to know which motives social engineers is very important. Normally, organizations reduce social engineering risks by tactics such as establishing security systems, security policies, etc. But by training employees, it can help organizations to prevent and identify the social engineering attempts, and let people no longer become the weakest link of the security chain.

Check more countermeasures and more specific information at Section 3 of book "Hacking the human: social engineering techniques and security countermeasures" [68].

Chapter 3
Trust Management

This chapter presents the trust management issues in social engineering attacks. Unreliable identity verification causes trust relationship existing between victims and social engineers. Thus, this chapter introduces the evolution from identity papers to eID cards like national eID cards, driving licenses, and ePassports. The security mechanisms used in European ePassports are described in detail, the advantages of those security mechanisms support our design in Chapter 4.

3.1 Trusting the Social Engineer

Thinking about the incidents we introduced in Section 2.1, Troy trusted the Trojan Horse "gift", then Troy fall. Scrap metal dealers trusted Victor Lustig's fake identity and tricks, then the Eiffel tower was sold twice. People trusted military social engineers' fancy-looking uniforms and lies, then Wilhelm Voigt deprived money from innocent people, Stanley Clifford Weyman swaggered in the White House, Douglas R. Stringfellow got elected to be a congressman. RSA staff trusted emails from trustworthy sources, then the master keys of RSA SecurID was stolen, leading to great loss and users' concern.

From these incidents, it is obvious to realize that trust is the serious issue in social engineering attacks, a social engineer's primary goal is to develop the trust to enable them to carry out their attack [68]. Considering yourself in victim's position. When victims were deceived by social engineers, a trust relationship existed between victims and social engineers, victims trust social engineers but they just didn't know the trust was established based on scams.

Security is all about trust. While the human is the weakest link of security, therefore trust management is very important in the context of social engineering. It is easy to say "I trust him" or "I don't trust him on this case", but trust is more or less instinct from experience or trust relastionship to human. Somehow trust management is challenging and vulnerable to be attacked by social engineers.

Trust is an abstractive and wide-ranging concept which can be used in various disciplines and circumstances. From a sociological viewpoint, trust can create social capital [34, 93, 17] –

> *"the ability of people to work together for common purposes in groups and organizations."*

Trust is like a compass for guiding us safely through a world of uncertainty, risk, and moral hazards [57].

As the formal statement from Merriam-Webster's Collegiate Dictionary[1], trust is the

> *"assured reliance on the character, ability, strength, or truth of someone or something."*

Therefore trust is one of the fundamental elements of relationships between people as well as between people and technology [16].

Kevin Mitnick claimed in his book "The Art of Deception: Controlling the Human Element of Security" [71] that trust is the key of deception, and the reason makes social engineering attacks so successful

> *"isn't because people are stupid or lack common sense. But we, as human beings are all vulnerable to being deceived because people can misplace their trust if manipulated in certain ways."*

Can trust be managed? The term trust management implies the affirmative answer. Building trust, assessing trustworthiness, and making decisions based on trust belong to trust management. In general, trust management refers to the management of the trustworthiness of relationships among entities [65], but in different contexts this term is being used with very different meanings. During the social engineering process, building and managing a relationship of trust is important [87], from incidents we introduced in Chapter 2, it is apparent that social engineers exploit trust relationships to manipulate victims for malicious purposes.

According to [11], trust management is

[1] Merriam-Webster's Collegiate Dictionary is an American dictionary of the English language

> *"a unified approach to specifying and interpreting security policies, credentials, relationships which allow direct authorization of security-critical actions."*

Trust management has received a lot of attention as it is an important component in decision making for a lot scenarios, including face-to-face interactions, thus appropriate methods are needed to effectively specify and manage trust relationships [38]. Identity verification is an important step when building and managing trust, one widely used method assists people to identify another person is checking identity documents, in next section, we'll introduce the development as well as the pros and cons of identity documents.

3.2 Eletronic Identity (eID)

Valentin Groebner[2] claimed that

> *"the close association of citizenship and identity papers that we take for granted today was not enforced until the early twentieth century."*

Identity documents[3] are the commonly used method to verify a person's identity in face-to-face interactions. Before World War I, most people did not have or need an identity document. However, with the development of civilization, identity document is being adopted gradually. in 1876, photograph started appear on identity document [46]. In 1985, the shape and size of identity cards were standardized by ISO/IEC 7810[4], which is the most common form of eID. Biometric data, such as fingerprints, face, voice, and iris etc, is now being embedded in identity documents.

With the rapid development and spread of identity documents, identity verification by using identity documents has become one important and necessary measure in daily life. Checking identity documents does not just happen when you cross borders or deal issues at law enforcement and government agencies. It happens everywhere, when you take public transit, buy tickets, check in hotels, enter schools or companies, or even a stranger knocks your door and implies to enter.

[2]Wikipedia, "Valentin Groebner", https://en.wikipedia.org/wiki/Valentin_Groebner [Online; Accessed 11 May 2016]

[3]Wikipedia, "Identity Document", https://en.wikipedia.org/wiki/Identity_document [Online; Accessed 11 May 2016]

[4]Wikipedia, "ISO/IEC 7810", https://en.wikipedia.org/wiki/ISO/IEC_7810 [Online; Accessed 11 May 2016]

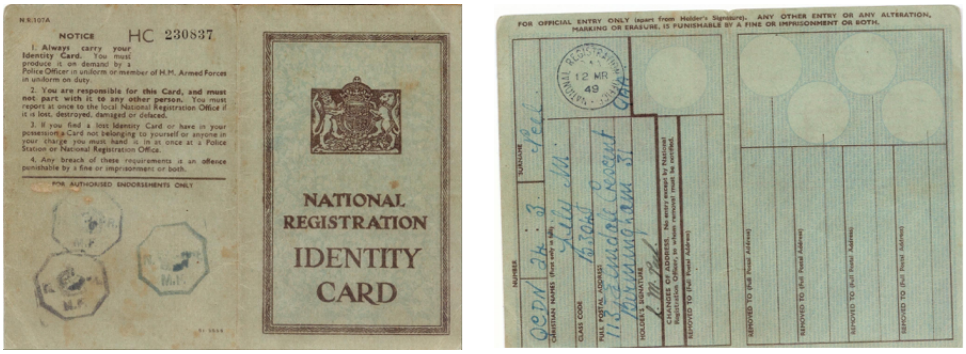


Figure 3.1: Cover and inside page of a mid-20th century ID card from [33]

In recently years, after the evolution of simple identity papers, eID[5] are emerging everywhere rapidly. Actually, both national eID cards and driving licenses we will introduce belong to eID card. eID cards can be used for both online and offline personal identification or authentication. One common point of these eID system is to let the user to be fully identifiable by using eID in online and offline transactions [10].

3.2.1 National eID Card

eID cards such as national eID cards can provide a universal, nationwide mechanism for user authentication. Governments trust eID they issued, thus most European countries have started deploying eID for government and private-sector applications [91]. And so do organizations trust eID they issue to their employees or customers. eID has become more and more universal.

3.2.1.1 National ID Evolution

Figure 3.1 show a mid-20th century national ID card, which is a very simple paper document with handwritten text and holder's signature. Because there is no photo or any biometric data inside this ancient ID card, it can be forged very easily, thus it is difficult to identify a fake ID card or a fake person who holds this ID card.

Nowadays, national ID cards have a huge improvement compares with the mid-20th ID card, Figure 3.2 is a sample of a national ID card (German)[6], this is a standard ISO/IEC 7810 chip card, detailed information of the card holder (including unique

[5]Wikipedia, "Electronic Identification", https://en.wikipedia.org/wiki/Electronic_identification [Online; Accessed 26 May 2016]

[6]Wikipedia, "German Identity Card", https://en.wikipedia.org/wiki/German_identity_card [Online; Accessed 25 May 2016]

Figure 3.2: German national ID card. Based on a photograph from Wikipedia.

document number, name, nationality, date and place of birth, photograph, signature, residence, access number for Radio-Frequency Identification (RFID) chip, color of eyes, height, data of issue and expire, issuing authority, and machine-readable zone) are printed on the card. There is also a biometric chip holds fingerprint record. The RFID chip is tamper-proof and has processing capability for cryptographic computations [99]. Smart card technology is widely used for eID cards includes national eID cards, Section 3.2.1.2 introduces the basic knowledge about smart cards.

Figure 3.3 is a sample of a resident permit card (German)[7], it has similar hardware construction of the national ID card, it is normally issued to people who lives in one country but is not a citizen of this country. On both front and back sides of the resident permit card, information including unique document number, name, nationality, date of birth, photograph, signature, residence, sex, height, color of eyes, type of document, access number for RFID chip, first day of validity, place of issue, data of expire, issuing authority, entitlements or restrictions, and machine-readable zone are printed on. There is also a biometric logo which implies that this card has

[7]Wikipedia, "German Residence Permit", https://en.wikipedia.org/wiki/German_residence_permit [Online; Accessed 25 May 2016]

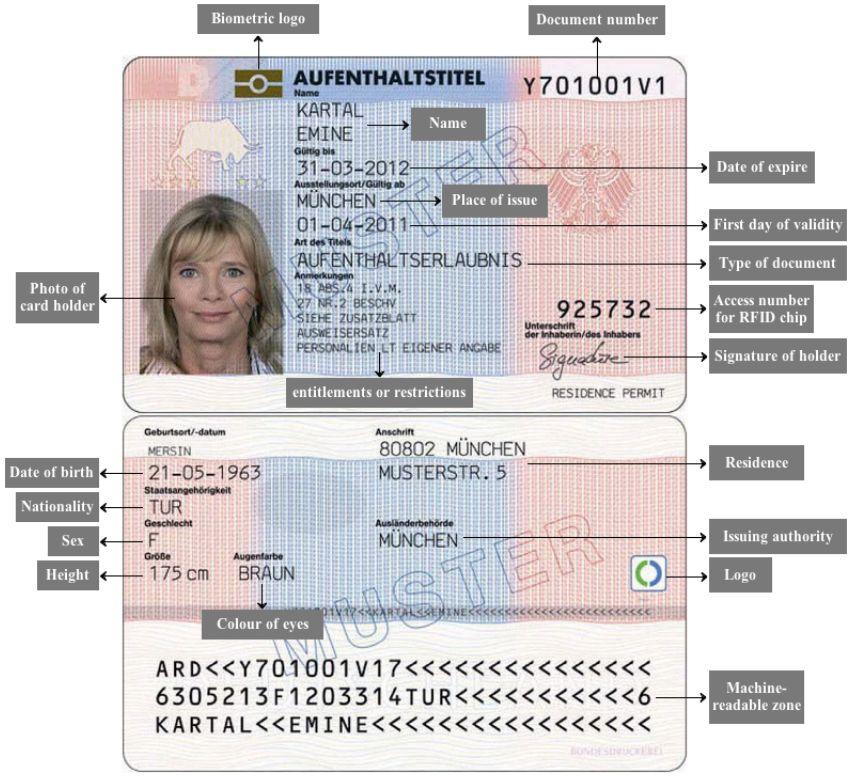


Figure 3.3: German resident permit card. Based on a photograph from Wikipedia.

a biometric chip which stores the card holder's biometric data.

3.2.1.2 Smart Card

Keith Mayes claimed in his book "Smart cards, tokens, security and applications" [69] that

> *"If a system is much more convenient because a particular card is being used then that is pretty smart thing to do, even if by technical the card is quite stupid."*

Figure 3.4 shows the construction of smart card, this kind of small plastic cards are familiar to us. Credit cards, bus cards, access cards are emerging everywhere in our daily life. We use smart cards to store and process information, but indeed its construction is much more complex than we thought. Smart cards can receive and

Figure 3.4: Smart card construction from [9]

transmit data to or from card readers. A smart card has embedded Integrated Circuit (IC) and chip. All contents are stored in the chip, this is the magic of "smart".

Typically, a smart card chip contains a Central Processing Unit (CPU) and three types of memory (Read Only Memory (ROM), Random Access Memory (RAM) and Electrically Erasable Programmable Read Only Memory (EEPROM)). ROM content (normally contains the operating system, well tested common functions and constant data) is written to when the card be produced, and the content can only be read in normal operations. RAM is used for dynamic data storage, such as programs. EEPROM is programmable and used for data that requires modification during operational lifetime [9].

The smart card chip is tamper resistance, meaning that it can resist known and anticipated attacks. This ability makes smart card such popular for identity verification. It encrypts bus and memory with cryptographic algorithms like Triple Data Encryption Standard (Triple DES) and RSA. The encryption of information ensures that only authorized readers have access to the sensitive data inside the chip, which also increase the security and popularity of smart card. More security features about smart cards at book [69].

Table 3.1: Information on the front side of Norwegian driving license

Number	Information
1	Surname
2	Given name(s)
3	Date and place of birth
4a	Date of issue of the license
4b	Date of expiry of the license. If the holder has several categories with different expiry dates, the longest expiry date is shown
4c	Name of the issuing authority
4d	Reference number (personal ID number)
5	Number of the license
6	Photograph of the holder
7	Signature of the holder
9	Category of vehicle(s) the holder is entitled to drive. Only the heaviest sub-categories are shown in each category

Table 3.2: Information on the back side of Norwegian driving license

Number	Information
9	Category of vehicle(s) the holder is entitled to drive
10	Date of issue of each category
11	Date of expiry of each category
12	Additional information/restriction(s)
13	Field for administrative information from the host country if the holder takes up residency in another EU/EEA-state

3.2.2 Driving License

Figure 3.5[8] shows a sample of driving license in Norway[9]. According to [1], the information on the Norwegian driving license are listed in Table 3.1 and Table 3.2.

Nowadays, driving licenses can also be used to verify identity in some countries. The main reason is personal information (name, birthday, photograph, and signature) of the holder are printed on the license card just like the national ID card. So in some cases, driving licenses have same functionality as national ID card. But biometric

[8]Wikipedia, "Driving Licence in Norway", https://en.wikipedia.org/wiki/Driving_licence_in_Norway [Online; Accessed 19 May 2016]

[9]Wikipedia, "Driving Licence in Norway", https://en.wikipedia.org/wiki/Driving_licence_in_Norway [Online; Accessed 26 May 2016]

Figure 3.5: Norwegian driving license from [1]

data like fingerprints are not stored in the driving license, thus driving licenses can provide function of identity verification in some situations, but can't replace national ID cards, resident permit cards, or passports.

3.2.3 ePassport

Passports are another identity documents used for identity verification, especially when passport holders cross borders to another country.

Figure 3.6 is the front cover of a Norwegian passport, the biometric logo on the cover implies this is an ePassport. Nowadays, there are more than 100 countries with ePassport systems in use, European countries have implemented ePassports over 10 years. Even though an ePassport is not a ID card-sized plastic card, with the increase in the quantity of ePassports the need arises to better understand their security and privacy implications [13]. ePassports use contactless smart card technology, including the RFID chip and antenna (for both power to the chip and communication) embedded[10]. Public Key Infrastructure (PKI) technology is also used to reduce fraud and enhance security [63] in ePassport systems.

Since the first generation specification of European ePassport implemented in 2004, European ePassport has experienced three generation specifications. Figure 3.7 shows the evolution of ePassport security mechanisms.

[10]Wikipedia, "Biometric Passport", https://en.wikipedia.org/wiki/Biometric_passport [Online; Accessed 26 May 2016]

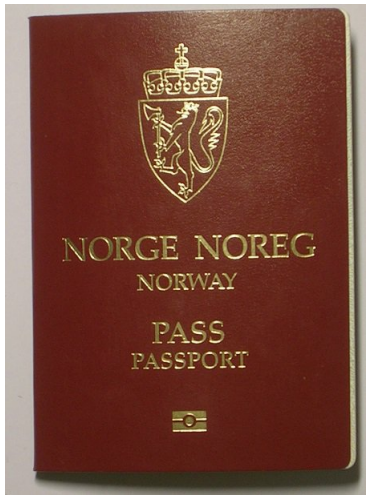


Figure 3.6: Norwegian passport from [23]

In the first generation (from 2004), Passive Authentication (PA) is the only mandatory protocol ensures authenticity and authenticates all relevant data stored on the RFID chip. Basic Access Control (BAC) is an optional protocol for establishing a secure communication channel between RFID chip and the reader by encrypting transmitted information. Active Authentication (AA) is another optional protocol to prevent cloning attack on RFID chip.

In the second generation (from 2006), Extended Access Control (EAC) is used to reach mutual authentication with CA and TA. CA ensures the authenticity of the chip while TA ensures the authenticity of the reader. As introduced in [78], in this generation, less-sensitive data such as the facial image is protected by BAC, but highly sensitive data like fingerprints is protected by EAC.

In the second generation (from 2014), Supplemental Access Control (SAC), a new mechanism based on PACE, is used to replace BAC by generating keys for secure messaging. In addition, the TA and CA protocols were also updated to version 2, TA need to be executed before CA in this version. Read more about the updated version of TA and CA at [86].

3.2.3.1 Passive Authentication (PA)

PA is used to allow a reader to verify that the data stored in the ePassport is authentic. However, it can only verify the data inside the RFID chip, it can't verify the chip itself, thus cloning attack can't be detected if we only use PA. As stated in

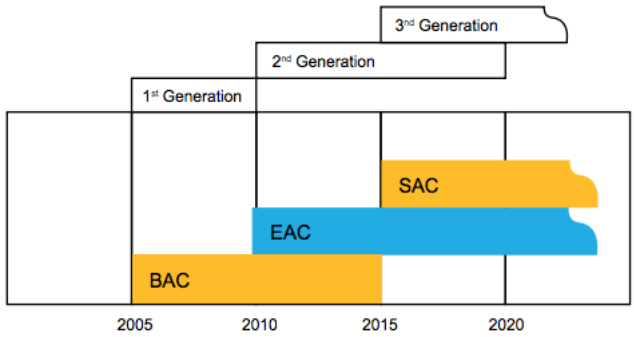


Figure 3.7: Evolution of ePassport security mechanisms from [73]

[85] by International Civil Aviation Organization (ICAO), the procedure of PA is described as below.

1. The reader reads the Document Security Object SO_D from the chip, then reads Document Signer (DC) from SO_D.
2. The back-end inspection system of the reader then verifies the digital signature with Document Signer Public Key KPu_{DS}. The Document Signer Certificate C_{DS} for KPu_{DS} is stored both at the inspection system as well as the RFID chip of ePassport. C_{DS} ensures the authenticity of SO_D, then the content of SO_D can be trusted.
3. Then the inspection system reads data from Logical Data Structure (LDS) of the chip.
4. Hash the data read from LDS and compare the result with the corresponding hash value in the SO_D, if match, the data inside the RFID chip is authentic and unchanged.

3.2.3.2 Active Authentication (AA)

AA is used to detect whether the RFID chip of ePassport is clone. If AA is supported, the chip also stores Active Authentication Key pair (KPu_{AA} and KPr_{AA}). The procedure of AA [85, 78, 41] is described as below.

1. The reader read the public Active Authentication key KPu_{AA} from the data group in the chip where is protected by PA, thus KPu_{AA} can be considered as authentic and unchanged.
2. The reader generates a random 64 bit string M_1 and sends it to the chip.

Table 3.3: Active Authentication (AA) procedure

Reader		**ePassport**
	$\xleftarrow{KPu_{AA}}$	
$M_1 \in_R \{0,1\}^{64}$		
	$\xrightarrow{M_1}$	
		M_2
		$h = H(M_1, M_2)$
		$C = E_{KPr_{AA}}(M_1 \| M_2 \| h)$
	$\xleftarrow{C}$	
$P = D_{KPu_{AA}}(C)$		
obtain M_2 and h		
$h' = H(M_1, M_2)$		
$h' = h$?		

3. The chip generates a random string M_2, usually the string length is 106 bytes.

4. The chip computes a hash value h of M_1 and M_2, then encrypts h, M_1, and M_2 using the private Active Authentication key KPr_{AA}:

$$C = E_{KPr_{AA}}(M_1||M_2||h) \tag{3.1}$$

   and sends the result to the reader.

5. The reader decrypts the cipher C with KPu_{AA} to obtain M_2 and h.

6. The reader computes a new hash value h' of M_1 and M_2, then compares it with the obtained h. If match, the chip can be considered as genuine.

3.2.3.3 Basic Access Control (BAC)

BAC is used to make sure that only authorized readers can read data from ePassports. When a reader attempts to read an ePassport, the ePassport needs the reader to prove its authority by extracting a pair of basic access keys (K_{ENC}, K_{MAC}) from Machine Readable Zone (MRZ) of the ePassport, K_{ENC} is the encryption key and K_{MAC} is the MAC key. Information includes the passport number (PNo), date of birth of holder (DoB), date of expire (DoE), and 3 check digits (C) are provided by MRZ. According to [58, 78], the process of BAC [85, 78] is described as below.

Table 3.4: Basic Access Control (BAC) procedure

Reader		**ePassport**
	$\xleftarrow{PNo,DoB,DoE,C}$	
$K_{seed} = SHA-1(PNo\|\|DoB\|\|DoE\|\|C)$		
$K_{ENC} = SHA-1(K_{seed}\|\|1)$		
$K_{MAC} = SHA-1(K_{seed}\|\|2)$		
	$\xrightarrow{challenge}$	
		$r_C \in_R \{0,1\}^{64}$
	$\xleftarrow{r_C}$	
$r_R, k_R \in_R \{0,1\}^{64}$		
$C_R = E_{K_{ENC}}(r_R\|\|r_C\|\|k_R)$		
$M_R = M_{K_{MAC}}(C_R)$		
	$\xrightarrow{C_R\|\|M_R}$	
		check M_R
		decrypt C_R to retrieve r'_C
		$r_C = r'_C$?
		$k_C \in_R \{0,1\}^{64}$
		$C_C = E_{K_{ENC}}(r_C\|\|r_R\|\|k_C)$
		$M_C = MAC_{K_{MAC}}(C_C)$
	$\xleftarrow{C_C\|\|M_T}$	
check M_C		
decrypt C_C to retrieve r'_R		
$r_R = r'_R$?		
$k_{seed} = k_R \bigoplus k_C$		$k_{seed} = k_R \bigoplus k_C$
$K_E = SHA-1(k_{seed}\|\|1)$		$K_E = SHA-1(k_{seed}\|\|1)$
$K_M = SHA-1(k_{seed}\|\|2)$		$K_M = SHA-1(k_{seed}\|\|2)$

1. When the reader reads the information from MRZ, a access key seed used for secure messaging is obtained by:

$$K_{seed} = SHA-1(PNo||DoB||DoE||C) \tag{3.2}$$

Access key K_{ENC} is derived by:

$$K_{ENC} = SHA-1(K_{seed}||1) \tag{3.3}$$

And another access key K_{MAC} is derived by:

$$K_{MAC} = SHA-1(K_{seed}||2) \tag{3.4}$$

2. The RFID chip generates a random 64 bit string r_C and sends it to the reader.

3. The reader receives r_C and then generates two random 64 bit strings r_R and k_R, then the cipher is encrypted the random strings by the algorithm below, here E is two-key 3-DES in CBC mode with an all-0 IV:

$$C_R = E_{K_{ENC}}(r_R||r_C||k_R) \tag{3.5}$$

4. The reader then computes the MAC, here M is the ANSI retail MAC [92].

5. The reader sends both the cipher and the MAC to the chip.

6. The chip checks the MAC and decrypts the cipher after received them to retrieve r'_C. Then verifies whether the retrieved r'_C matches the original r_C. If either check fails, the chip aborts. Then the reader extract k_R.

7. The chip generates another 64 bit random string k_C .

8. The chip computes a cipher, here E is also two-key 3-DES in CBC mode with an all-0 IV:

$$C_C = E_{K_{ENC}}(r_C||r_R||k_C) \tag{3.6}$$

9. The chip then computes the MAC of the cipher, here M is also the ANSI retail MAC [92].

10. The chip sends the cipher and the MAC to the reader.

11. The reader checks the MAC and decrypts the cipher after received them to retrieve r'_R. Then verifies whether the retrieved r'_R matches the original r_R. If either check fails, the chip aborts. Then the reader extract k_C.

12. Both the reader and the chip compute the session key seed (k_{seed}) by

$$k_{seed} = k_R \bigoplus k_C \tag{3.7}$$

Table 3.5: Chip Authentication (CA) procedure

Reader		**ePassport**
	$\xleftarrow{CKPu_{CA},D}$	
verify $CKPu_{CA}$ using Passive Authention generate $(RKPu_{CA}, RKPr_{CA})$ using D		
	$\xrightarrow{RKPu_{CA}}$	
$K_{seed} = K_a(CKPu_{CA}, RKPr_{CA}, D)$ $K_E = SHA-1(K_{seed}\|\|1)$ $K_M = SHA-1(K_{seed}\|\|2)$		$K_{seed} = K_b(RKPu_{CA}, CKPr_{CA}, D)$ $K_E = SHA-1(K_{seed}\|\|1)$ $K_M = SHA-1(K_{seed}\|\|2)$

13. Both the reader and the chip generate a new session encryption key K_E

$$K_E = SHA-1(k_{seed}||1) \tag{3.8}$$

and generate a new session MAC key K_M

$$K_M = SHA-1(k_{seed}||2) \tag{3.9}$$

From this point on all communication between the reader and the ePassport is secured using encryption key K_E and MAC key K_M.

3.2.3.4 Extend Access Control (EAC)

EAC contains two authentication approaches, i.e. CA and TA.

Chip Authentication (CA)

CA is used to replace AA in the first generation European ePassport. As this is a mandatory protocol, Chip Authentication key pair ($CKPu_{CA}$ and $CKPr_{CA}$) is stored in the RFID chip. The process of CA [80, 78] is described as below.

1. The chip sends the public Chip Authentication key $CKPu_{CA}$ as well as the Diffie-Hellman key agreement parameters D to the reader.
2. The reader verifies $CKPu_{CA}$ by PA, and generates its own ephemeral key pair ($RKPu_{CA}$ and $RKPr_{CA}$) using D.
3. The reader sends $RKPu_{CA}$ to the chip.

Table 3.6: Terminal Authentication (TA) procedure

Reader		**ePassport**
	$\xrightarrow{certificates}$	
		verify certificates
		extract the public key $RKPu_{TA}$
		$r \in_R \{E\}$
	$\xleftarrow{r,ID}$	
$S = Sign_{RKPr_{TA}}(ID\|\|r\|\|H(RKPu_{TA}))$		
	$\xrightarrow{S}$	
		$Verify_{RKPu_{TA}}(ID\|\|r\|\|H(RKPu_{TA}))$

4. The chip computes a new seed key using the shared information by

$$K_{seed} = K_b(CKPu_{CA}, PKPr_{CA}, D) \tag{3.10}$$

While the reader also computes a new seed key by

$$K_{seed} = K_a(RKPu_{CA}, CKPr_{CA}, D) \tag{3.11}$$

5. Then both parties generate a new encryption key K_E and a new MAX key K_M, these two keys can replace sessions keys derived by BAC to enable secure messaging. Read more about CA version 2 used in the third generation European ePassport at [86].

Terminal Authentication (TA)

TA is executed only if more sensitive data such as biometrics is required to be read. It allows the RFID chip to validate the reader, the reader needs to prove its authenticity and authorization by using digital certificates, therefore TA requires support from PKI. The process of TA [80, 78] is described as below.

1. The reader sends a certificate chain to the chip, including a certificate of back-end inspection system that received from local Document Verifier (DV), as well as a certificate of the DV that received from Country Verifying Certificate Authority (CVCA).
2. The chip verifies the certificates and extracts public key $RKPu_{TA}$ of the reader.
3. The chip generates a random string r and sends it to the reader as a challenge.

4. The reader signs the message with its private key

$$S = Sign_{RKPr_{TA}}(ID||r||H(KPu_{TA})) \tag{3.12}$$

ID refers to document number printed on MRZ of ePassport.

5. The reader sends the signature to the chip.

6. The chip verifies the signature by

$$Verify_{RKPu_{TA}}(ID||r||H(RKPu_{TA})) \tag{3.13}$$

ID refers to document number printed on MRZ of ePassport. If the verification is passed, the chip then allows the reader access to sensitive biometric data. Read more about TA version 2 used in the third generation European ePassport at [86].

3.2.3.5 Password Authenticated Connection Establishment (PACE)

In the third generation specification of European ePassport, PACE is used to replace BAC to establish secure communication channel between the ePassport and the reader. When PACE is used, the reader and the RFID chip share a common password π, which is read from MRZ or Card Access Number (CAN). The process of PACE [86, 78] is described as below.

1. The chip generates a random string r and encrypts it by

$$c = E_{K_\pi}(r) \tag{3.14}$$

here $K_\pi = SHA-1(\pi||3)$.

2. The chip sends the cipher c and the Diffie-Hellman key agreement parameters D to the reader.

3. The reader uses the share password π to decrypt the cipher c and obtains r.

4. Both parties compute the Diffie-Hellman ephemeral key agreement parameters D' using D and r.

5. The reader generates the key pair (RPu_{PACE}, RPr_{PACE}) and sends the public key RPu_{PACE} to the chip.

6. The chip generated the key pair (CPu_{PACE}, CPr_{PACE}) and sends the public key CPu_{PACE} to the reader.

Table 3.7: Password Authenticated Connection Establishment (PACE) procedure

Reader		**ePassport**
	$\xleftarrow{\pi}$	
		$r \in_R \{E\}$
		$c = E_{K_\pi}(r)$
	$\xleftarrow{c,D}$	
$r = D_{K_\pi}(c)$		
$D' = Map[D, r]$		$D' = Map[D, r]$
(RPu_{PACE}, RPr_{PACE})		
	$\xrightarrow{RPu_{PACE}}$	
		(CPu_{PACE}, CPr_{PACE})
	$\xleftarrow{CPu_{PACE}}$	
$K_{seed} = K_a(CPu_{PACE}, RPr_{PACE}, D)$		$K_{seed} = K_b(RPu_{PACE}, CPr_{PACE}, D)$
$K_E = SHA-1(K_{seed}\|\|1)$		$K_E = SHA-1(K_{seed}\|\|1)$
$K_M = SHA-1(K_{seed}\|\|2)$		$K_M = SHA-1(K_{seed}\|\|2)$
$T_R = MAC_{K_M}(CPu_{PACE}, D')$		
	$\xrightarrow{T_R}$	
		$T_C = MAC_{K_M}(RPu_{PACE}, D')$
	$\xleftarrow{T_C}$	

7. The chip computes a new seed key using the shared information by

$$K_{seed} = K_a(RPu_{PACE}, CPr_{PACE}, D) \tag{3.15}$$

While the reader also computes a new seed key by

$$K_{seed} = K_a(CPu_{PACE}, RPr_{PACE}, D) \tag{3.16}$$

8. Both parties generate a new session encryption key K_E and generate a new session MAC key K_M.

9. The reader computes an authentication token by:

$$T_C = MAC_{K_M}(RPu_{PACE}, D') \tag{3.17}$$

and sends T_R to the chip for verification.

10. The chip computes an authentication token by:

$$T_R = MAC_{K_M}(CPu_{PACE}, D') \tag{3.18}$$

and sends T_C to the reader for verification.

3.3 eID Challenges

Because of the widely adoption of eID, personal interaction within a community is also replaced widely by eID. This situation makes security become more essential than ever.

eID offers strong cryptographic identification or qualified digital signatures [10]. Even though eID cards have various architecture and specific designs, the basic usage are similar. When holding a eID card in a transaction, a card reader is always the communication media between the card and the back-end service.

However, in spite of the popularity of eID cards, identity frauds are still rampant. Indeed eID card has a lot advantages and other security measures, in past few years, identity fraud has receiving more and more attention. This is also frequently happened attacks launched by social engineers as we introduced in Chapter 2. As stated in [35], reducing the frequency of identity fraud is one of the prime challenges facing eID card systems, which is also the main purpose of the face-to-face authentication mechanism we will propose in our project.

In order to design a mechanism to solve social engineering attack in face-to-face authentication, we need to achieve more data support from our living environment, the following chapter introduces the questionnaire we made as well as the result analysis.

3.4 eID Questionnaire

In order to investigate the usage of eID as well as the prospect of smart card based private eID manager, we designed a questionnaire to collect data to analysis, the complete questionnaire is attached in Appendix B. We divided the questionnaire into five parts and chose Google Forms[11] as the collection platform. Until the end of May, we have received results from 46 respondents[12] and a lot valuable suggestions.

3.4.1 Purpose

The purposes of this questionnaire includes:

1. Investigate the usage of various eID, including the use frequency and user experience.

[11] Google Forms is free tool to create and analyze surveys.

[12] The deadline of questionnaire is July, we can receive more results until then. But considering of the thesis deadline, we only count results we received before June.

2. Investigate the usage of smart card and smart phone, in order to figure out the advantages and disadvantages of smart devices, and discover the probability and usability of our new mechanism.

3.4.2 Methodology

We took the following steps to complete this questionnaire.

3.4.2.1 Design Questionnaire

According to the purposes of this questionnaire, we divided the questionnaire into five parts.

1. Questions about respondent's profile. This part helps us to quickly determine which user group the respondent belongs to.

2. Questions about the usage of eID. In this part, we investigate situations of using frequency, obtaining methods, trust degree of different types of eID. We divided eID types by the usability, for instance the combination of card and PIN code.

3. Questions about the usage of smart cards. In this part, we investigate situations of cards amount, using frequency, lost experience, and willing to use multiple-eID cards.

4. Questions about the usage of passwords. In this part, we investigate situations of passwords amount, using frequency, forgot experience, and willing to use multiple-password cards.

5. Questions about the usage of smart phones, NFC function and fingerprints.This part helps us to have clearer ideas about the design of this project.

3.4.2.2 Test Questionnaire

Before we formally start collecting data from respondents, we test and improved the questionnaire for several times. We invite two master student and one engineer with technology-related background to do the questionnaire. They gave great suggestions about unreasonable questions or choices, unclear descriptions and logics. We improved the questionnaire and got the final version, which is also the one spread to all the recipients, please check the complete questionnaire in Appendix B.

3.4.2.3 Choose Platform for Questionnaire

In order to collect and analyze real-time data precisely and effectively, after comparing platforms such as Google Forms and Google Consumer Surveys[13], we decided to use Google Forms as our platform, as it has advantages like well-known, popular to use, willing to accept by recipients, great user experience, etc. The results of collected data can be displayed graphically is also a key reason made us choose it.

3.4.2.4 Define User Groups

According to the using frequency and technology-related background, We divided the recipients of the questionnaire into following user groups.

1. Professionals. Individuals have technology-related background and are highly educated, such as professors and PHD students.
2. Administration staff. Individuals have more or less technology-related background, but are responsible for all the staff in the organization.
3. Youth. Individuals have somehow technology-related background, such as bachelor and master students.
4. Elderly. Individuals have few technology-related background and not familiar with smart devices.
5. Challenged. Individuals have physical or mental issues which may affect their usage of smart devices.

3.4.2.5 Find Appropriate Recipients

Pointing on features of different user groups, we decided to find appropriate recipients from following organizations.

1. Professionals. Professors and PHD students at NTNU i Gjøvik.
2. Administration staff. Administration staff at NTNU i Gjøvik.
3. Youth. Bachelor and master students at NTNU i Gjøvik.
4. Elderly. We will visit a few elderly volunteers in next phase of project IDforU.
5. Challenged. We will visit a few challenged volunteers in next phase of project IDforU.

[13]Google Consumer Surveys is a business tool to create and analyze surveys.

3.4.2.6 Deliver Questionnaire

We delivered the questionnaire mostly by email. In order to deliver questionnaire to all students at NTNU i Gjøvik, we asked university for approve first then sent emails to all students. Questionnaires delivered to university staff were sent through Qingbao Guo's staff email.

3.4.3 Result Analysis

By the end of May, we have received responses from 46 respondents in total. From the summary of all responses by Google Form, we can sum up the following statistics.

3.4.3.1 Respondents

From the first part of the questionnaire, we can sum that among 46 respondents, there are 4 professionals, 7 administration staff, 34 youth, and 1 elderly[14].

3.4.3.2 eID Usage

We divided eID into the following types:

- **Type 1:** Nickname/pseudonym + password
- **Type 2:** Real name/ID(student/personal ID) + password
- **Type 3:** Supported by a token (generates temporary random numbers for one-time usage), e.g. BankID
- **Type 4:** Card + PIN code (employee card, bank card, etc)
- **Type 5:** Supported by a SIM card + mobile device (SMS message one-time passcode with manual input), e.g. MinID
- **Type 6:** Supported by a SIM card + mobile device (SMS message one-time passcode with visual comparison), e.g. BankID on Mobile
- **Type 7:** Supported by biometric information (fingerprint, iris scan, etc), e.g TouchID on iPhone

From the summary of this question at Table 3.8, we can get to know that the most commonly used eID form is still traditional username and password (Type 1 and 2). However, we all know that the security offered by simple combination of username

[14]Whether the respondent is challenged person or not can't be reflected from the questionnaire, we planned to visit a few challenged volunteers in next phase of project IDforU. But we don't have enough time to finish this in the thesis duration.

Table 3.8: Summary of usage of different types of eID, the number refers to the amount of respondents out of 46 (the total respondents)

	Daily	Weekly	Monthly	Yearly	Never used	Do not have
Type 1	40	4	2	0	0	0
Type 2	19	24	3	0	0	0
Type 3	5	23	15	1	1	1
Type 4	21	14	5	0	1	5
Type 5	2	11	18	9	4	2
Type 6	6	6	9	4	5	14
Type 7	11	1	1	0	7	25

Figure 3.8: How many cards do you have (including credit cards, bus cards, access cards, membership cards, etc.)?

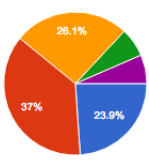
and password is weak and vulnerable. Thus, smart cards become a popular solution (Type 4), such as national eID cards, driving licenses we introduced before. It is obvious from this summary that the usage frequency is high. A interesting data about biometric solution (Type 7) is that the daily usage of biometric solutions is second only to the smart card solutions, but it is not widespread enough so over half of respondents (25) do not have any biometric eID.

3.4.3.3 Smart Card Usage

From the following two pie charts and corresponding questions, we can conclude that most people have around 1 to 10 smart cards. Obviously, it causes troubles of managing smart cards as well as the credential related to these cards. Figure 3.9 implies strong willings from most respondents of using biometric recognition to replace passwords when using smart cards. In the meantime, the security concerns also impede some respondents' willings of using such type of eID.

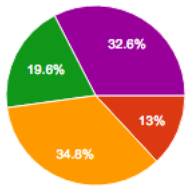
3.4.3.4 Password Usage

From the responses about the usage of different types of eID, we already knew that password is the most common used eID solution. However, from all responses about



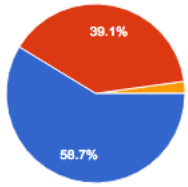
Yes, Absolutely!	11	23.9%
Yes, but with some worry about card's lost and security/privacy of the card's content, etc.	17	37%
No, I am worry about about card's lost and security/privacy of the card's content, etc.	12	26.1%
I don't know	3	6.5%
Other	3	6.5%

Figure 3.9: If there has a smart card that saves all of your IDs in your daily life which is also combining fingerprint recognition instead of using passwords, are you willing to use it?



0	0	0%
1 - 5	6	13%
6 - 10	16	34.8%
11 - 20	9	19.6%
21 ++	15	32.6%

Figure 3.10: How many passwords do you have (Including PIN code, any password you are using)?

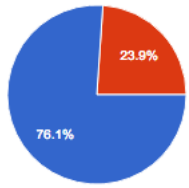


Yes	27	58.7%
No	18	39.1%
I don't know	1	2.2%

Figure 3.11: Do you feel any inconvenience when you do have to use multiple passwords in your daily life?

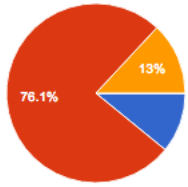
passwords, we can find out some issues about it. From Figure 3.10, it shows that everyone uses passwords, over 80% of respondents have more than 6 passwords, over 50% of respondents have more than 10 passwords, and over 30% of respondents have more than 20 passwords. This statistic implies that using passwords is quite common.

In traditional cryptosystems, passwords is quite common used. However, passwords can be forgotten, lost, or stolen and, thus, cannot provide non-repudiation [110]. The inconvenience and security risks of using passwords can be revealed from these statistics and charts. Hence, it is easy to understand why replacing passwords becomes a tendency in recent years.



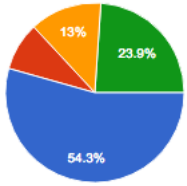
Yes	35	76.1%
No	11	23.9%
I don't know	0	0%

Figure 3.12: Have you used the same password for multiple applications or/and websites?



Yes, once	5	10.9%
Yes, several times	35	76.1%
No	6	13%
I don't know	0	0%

Figure 3.13: Have you forgot any of you passwords?



Yes	25	54.3%
Some of my smartphones have	4	8.7%
No	6	13%
I don't know	11	23.9%

Figure 3.14: Do your smart phones have NFC?

3.4.3.5 NFC Usage

Contactless smart card technology gives us a lot chances to exploit this advantages in various scenarios. Moreover, the NFC function of smart phones shows more possibilities of using smart cards and smart phones in our daily life. With the rapid development of modern smart phones, embedding NFC function in smart phones gets more and more popular. From Figure 3.14, we can get to know that over 50% respondents' smart phones have NFC function. This is a good sign that using NFC function on smart phones have large possibility.

Chapter 4
Design

Humans make mistakes, and technology must respect that [56]. In the world of social engineering, human is the most serious security weakness while using technologies properly is difficult.

Social engineering is a collection of various attacks target at human trust, including technical attacks such as phishing, or non-technical attacks such as Trojan Horse story we introduce in Section 2.1.1. In our project, we aim at using technical methods to minimize social engineering threats in face-to-face authentication scenarios.

Thinking about the intended victim has no substantial background or knowledge of social engineering, he or she might be a friend or family member, and have nobody to ask for help when being attacked. This is the average human in real life. However, thinking again about the face-to-face scenarios which average human may encounter in daily lives, bank staff deal with account issues, salesmen come to promote products, repairmen come from home repairs services, receptionists at hotels or other organization, strangers ask for help, etc. Social engineers using fake identity and deceitful tricks to win trust and manipulate victims for malicious purposes, we already know relevant incidents and statistics from Chapter 2. How to let a potential victim identify the person in front of him or her has the real and correct identity. If there is any case related to each parties, the potential can also figure out limits of authority of this person.

We propose a mechanism for face-to-face authentication in this section, including the basic infrastructure of the whole system, the procedure of face-to-face authentication as well as the used technologies.

4.1 Scenarios

In the beginning of the design procedure, based on the purposes of this project and the problems we desire to solve, we assume several scenes would happen in

daily life for designing a comprehensive and effective mechanism for face-to-face authentication.

4.1.1 Scenario One

Considering a police officer who wears fancy-looking uniform knocking out your door, he wants to enter house to discuss a theft case happened nearby, but you never receive any notice in advance or hear about this case. Holding doubt in the heart, but can hardly say "no" to a police officer.

4.1.2 Scenario Two

Considering you have booked a home repair service online. When the repairman arrives and shows his identity then enters your house, because you lack knowledge of repair issues, you have no idea about which tools he can use, which room he can enter, or how long it normally takes. These problems can lead to your doubt and contradiction about home security or repair spending.

4.1.3 Scenario Three

Considering you meet a stranger on the way who wants your help to take a ride with you. You have no idea whether the stranger has malicious intentions or not, but your conscience drives you to help the stranger. Then you are caught in a contradiction, refusing to help the stranger in order to prevent potential dangers, or follow your intuition to invite the stranger to get on your car.

4.1.4 Scenario Four

Considering you are assigned to meet a staff from a partner company, but the person who showed up at the meeting place is another person you don't know. This person claims to be assigned to replace the original one, but you never heard this information. The person shows his business card to you and tells you some reliable information about the partner company and business. Thus you suppose to have no excuse to doubt his identity, and it is not convenient to call the partner company and ask about the replacement in front of him. However, the business you will discuss is very sensitive and valuable secrets which can never be disclosed. These doubts and hesitation leads you in a dilemma.

4.1.5 Scenario Five

Considering an individual accessing a wine shop which restricts access to teenagers only. Hence, checking eID card to confirm age is a necessary procedure. Even though current eID cards can easily offer age information, but other irrelevant information is

also displayed to the clerk or other customers standing around, for instance, personal numbers, name, address, date of birth, etc. These sensitive information may lead to crimes like identity theft if is disclosed to malicious social engineers.

4.2 Human Authentication

Because of the major differences in calculation capabilities and memory of people versus computers, computers have large memories and can do large calculations quickly and correctly, but human don't. Therefore authenticating people is significantly different from authenticating machines.

4.2.1 Authentication Approaches

According to [97], the approaches for human authentication rely on at least one of the following:

- **Something you know.** This is the most common method for human authentication. Passwords are the most widely used example of "something you know". In ancient times, sentries only allowed strangers know the same password or watchword to pass. Nowadays, passwords are used everyday to access smart devices and online services. Unfortunately, something you know can be easily forgot or disclosed, which causes risks and threats when authenticating people.
- **Something you have.** To resolve the disadvantages of "something you know", "something you have" is another method for human authentication, for instance a smart card or token. However, these devices could be lost or stolen, which also cause risks.
- **Something you are.** Unlike the above two methods, "something you are" is much harder to be lost or stolen, for example, fingerprints, face, voice, hand geometry, etc. But biometric sensors are fairly expensive and (at present) not accurate enough.

4.2.2 Multi-factor Authentication (MFA)

From the introduction of human authentication methods, it is obvious that any method has flaws. In such situation, Multi-factor authentication (MFA) is a common used solution for human authentication by utilizing a combination of multiple different components. For example, a combination of a password and a smart card is a method of Two-factor authentication (TFA) used for bank system. When it comes to authentication, Single-Factor Authentication (SFA) can be easily circumvented by criminals, thus using TFA is ever-increasing necessity in present human authentication [102].

Considering of the common forms of eID checks in face-to-face interactions, generally, you need to show "something you have" – national eID cards, driving licenses, or ePassports. Apart from seldom authority agencies which have access to all sensitive data (biometrics included) by reading your eID cards. Moreover, not every organization owns an authorized eID card reader or has rights to read your eID card and collect your biometric data. Under such conditions, identity verification is always done by checking the authenticity of "something you have" and comparing the photograph and your face, sometimes verifying information on the eID by asking "something you know" is an additional action.

However, because of the weaknesses of "something you know" and "something you have", identity frauds is not a fresh news. Using "some you have" can make an eID solution with low risk of identity fraud. For the purpose of minimizing risks of identity fraud in face-to-face interactions, a combination of "something you have" and "something you are" is the general idea of our project.

4.3 Design Goals

In order to specify the design goals of our project, we sum up possible doubts from the five scenarios we assumed in Section 4.1. From Table 4.1, we can make a summary of our design goals, which are

1. How to verify the identity of the stranger?
2. How to get to know the limit of authorities of the stranger?
3. How to get to know the motivation and specific information about the stranger's visit or request?
4. How to limit the sensitive information to be read by strangers?
5. How to automatically record the authentication action?

The doubt *"Does he has malicious purposes?"* is conjectures of the stranger's mental activity, it can't be solved in our design.

4.4 Design Decision

To meet the design goals, we made the decision of the final design. Combining smart cards, biometrics, and smart phones, we design a biometric eID card infrastructure for face-to-face authentication. Just like the infrastructure of national eID cards, we design a similar infrastructure of biometric eID cards.

Table 4.1: Summary of possible doubts in the five scenarios

Scenario	Doubts	Summary
1	• Is he a real policeman? • was a real theft case happened? • Does he has rights to enter in?	
2	• Is he a real repairman? • Is he the assigned repairman by the service company? • Does he has rights to enter in bedrooms? • Will he work slow on purpose for more payment?	Verify identity Check limit of authorities
3	• Is he the person he claims? • Does he has malicious purposes? • Will anybody know who he is if bad things happened?	Check motivation and case description Control information access Record automatically
4	• Is he the person he claims? • Is he from the partner company? • Is he assigned to the business meeting? • How many business secrets he has rights to know?	
5	• Does the clerk record my personal information on the card? • Does anybody nearby peep my ID card?	

The general idea of the design is every citizen holds a biometric eID card instead of normal national eID card, the biometric eID card is issued by the authority agency. By exploiting the combination of "something you have" and "something you are" as a TFA method in human authentication, when authenticating card holder's identity, biometric verification is an additional but necessary process done on the eID cards. Moreover, authorized terminals (e.g. authorized card reader, authorized online application, authorized application on computers and smart phones) would be used as the trusted relying-party while authentication. If there is pre-defined transaction between both parties, the description of motivation and limits of authorities will be displayed, otherwise a new transaction will be created and recorded automatically. Section 4.5 introduces specific design about the biometric eID card infrastructure.

In current systems, a citizen's personal information is printed on the eID card. However, in order to protect the privacy of citizen (refer to the scenario of shopping in wine shop in Section 4.1.5), the personal information need to be only stored inside the eID card but not printed outside. In addition, the terminal should be anonymous, which means even though it is used to access the most sensitive information but it will not keep any sensitive information in the terminal. All those information are stored in authorized server and database, citizens' basic personal information is also stored in their eID cards.

Considering the risks of the system, for instance, a real eID card could be used at an untrusted or malicious terminal (no matter it is a own device or a third-party device), similarly, a fake or untrusted eID card could also be used at an real terminal. Therefore, the system should be able to protect the security and privacy of both eID card and terminal autonomously. The eID in our design can be used to replace national eID cards, which requires the eID can be trusted nationwide. As we mentioned in Section 3.2 that governments trust eID they issued, thus the eID should be issued by an authority agency. In addition, the terminal used to read eID should also be a trusted device.

Limits of authority restricts the usage of eID for face-to-face authentication, it is generally impossible for ordinary people to execute identity verification accurately with authorized reader or system in daily life. However, the assumptions we proposed in Section 4.1 are not rare around us, these scenes are vulnerable to social engineering attacks. Hence the requirement of multi-scene support is important and useful, our design can provide huge convenience by reach this requirement.

In this design, we consider the conditions and performances of entire procedure is ideal, which means we do not includes situations like inaccurate biometric sensors, expensive devices, or programming difficulties.

4.5 Biometric eID Card Infrastructure

This section introduces each link of the biometric eID card infrastructure, including issuing a biometric eID card to a citizen, a citizen updates his biometric eID card with authorized terminal, revoking a lost or stolen biometric eID card, and face-to-face authentication between two parties in two scenarios.

4.5.1 Issue

To ensure the infrastructure always keep available, and the biometric eID card which a citizen holds should be able to use over the citizen's entire lifetime. Just like issuing

national eID cards and ePassports, the biometric eID cards will be issued as shown in Figure 4.1. The procedure of issuing a biometric eID card is as following:

1. A citizen requests a biometric eID card from the authority agency (normally the police office or immigration office).
2. The authority agency collects personal information and biometric data from the citizen.
3. The authority agency stores the collected data into the official database through the official web server.
4. After confirming the collected data, the authority agency will personalize a specific biometric eID card which stores personal information of the citizen.
5. The authority agency issues the biometric eID card to the citizen.

To avoid information disclosure caused by the printed information on the eID card, which can be read and memorized by social engineers. We decided to leave both sides of biometric eID cards blank, thus all information retrieve need to be done with authorized terminals.

To reach the goal of verifying biometric data on the card, the biometric eID card needs to equip a biometric sensor. Moreover, in order to prove the authenticity of the biometric eID card, a certificate issued by the authority agency should be stored in the card.

In addition, for the purpose of authenticating the terminal, the ROOT certificate of the authority agency is also stored in the card. Read more information about the authentication procedure between the card and the terminal in Section 4.7.2.

4.5.1.1 Java Card

To realize a portability and security biometric eID cards, Java Card technology is the main and basic technology used for the implementation of the biometric eID cards.

Java Card[1] is a software technology which provides a secure standard environment for Java-based applications (Java applet) that run on smart cards and similar trusted devices [83]. Java Card allows users to program smart cards with Java programming language, which can make smart cards with specific features. The portability and security of Java Card are the main design goals of the Java Card, also the main reasons of our choice.

[1]Wikipedia, “Java Card”, https://en.wikipedia.org/wiki/Java_Card [Online; Accessed 24 May 2016]

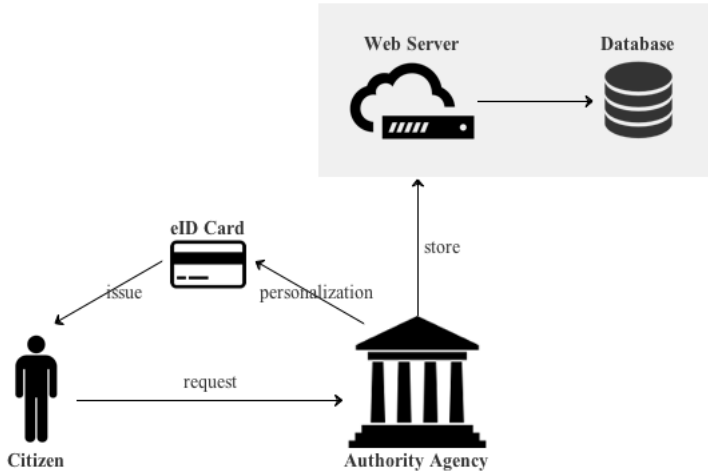


Figure 4.1: Issue a biometric eID card

As we said, using Java applets to program smart cards gives us portability. Figure 4.2 illustrates the architecture of a Java Card application, it is not standalone, but rather includes card-side, reader-side, and back-end elements. The security of Java Card is reflected in the following points:

1. **Data encapsulation** All the data is stored in the Java applet it belongs to, and each Java applet executes in the Java Card Virtual Machine (VM), which is an isolated environment separates from the underlying Operating System (OS) and hardware.

2. **Applet Firewall** Multiple Java applets can run on a same Java Card VM, and each applet is separated from each other by a firewall, which can avoid data access from other Java applets. Which also shows the probability of running multiple IDs in the same card.

3. **Cryptography** Java card supports symmetric key algorithms like Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), asymmetric key algorithms like RSA, and other cryptography like elliptic curve cryptography and so on.

4. **Java Applet** Java applet only processes and responds when there is incoming requests.

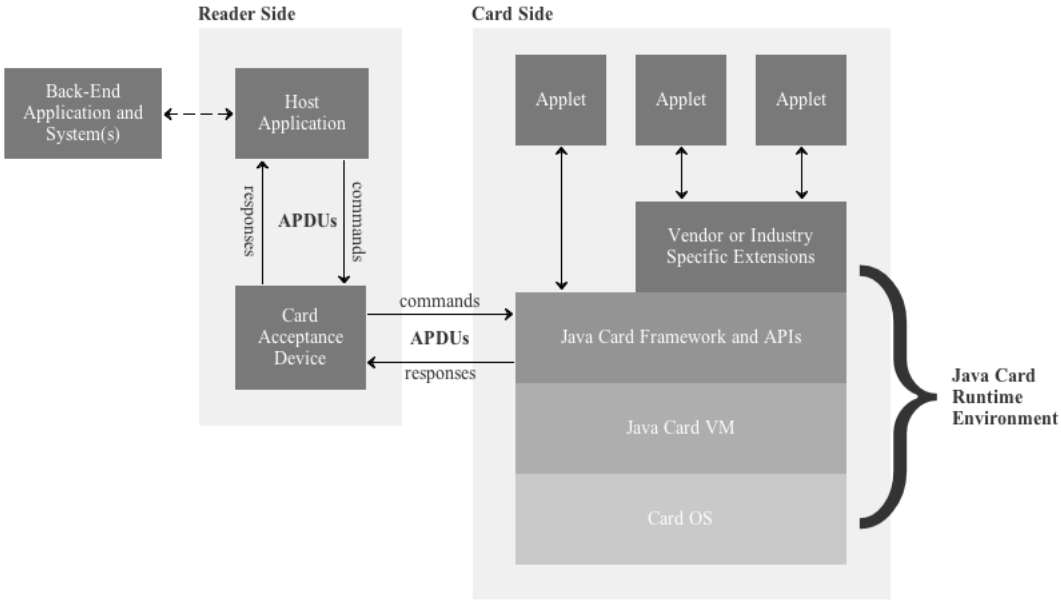


Figure 4.2: Architecture of a Java Card application. Based on a figure in [88]

4.5.1.2 Match-on-Card Fingerprint Verification

Because biometrics are extremely sensitive data which can't be disclosed. Thus, we decided to stored the biometric data only in the authority agency and the eID card that the citizen holds. Additionally, the verification process will happen on the eID card, this is another action we decided to take in order to protect sensitive biometric data.

Figure 4.3 shows the general operators of a biometric system. As biometrics is not a research focus of our project, here we only introduce it briefly. According to [66], normally there are three general operators of a biometric system:

- **Enrollment** Capture biometric characteristic with a biometric sensor and produce a sample first, then extract feature from the sample to produce a feature set, then use the feature set to produce an enrollment template, and save the enrollment template with other personal information into system database together in the end. There is also a Personal Identification Number (PIN) or a username can be used as the unique identifier of the enrollment template to the system database. From this point on, an individual is registered in the

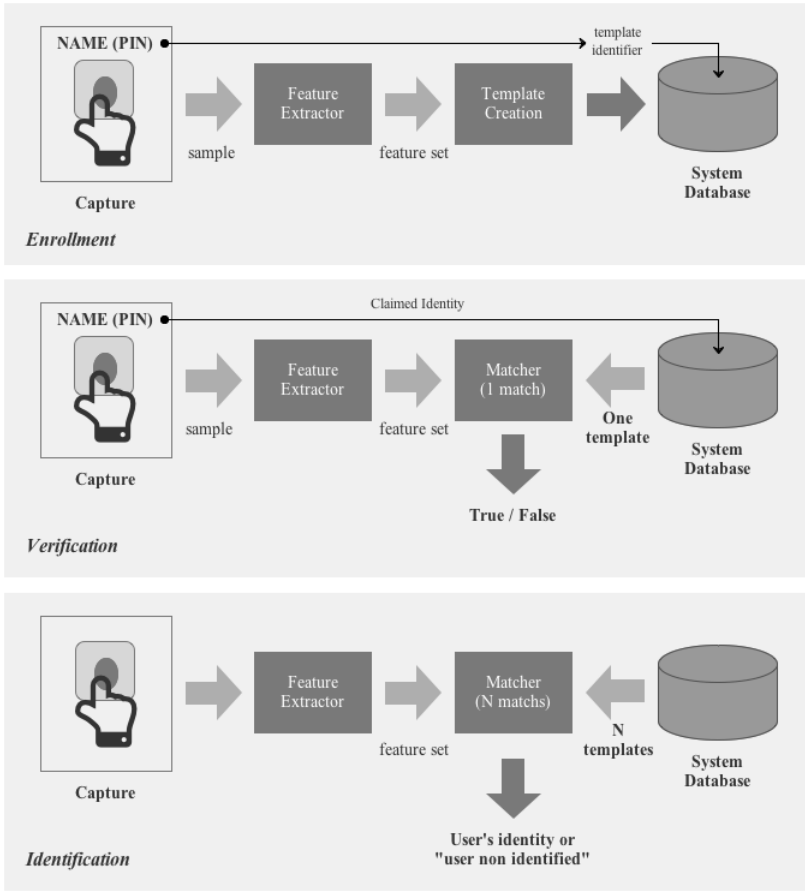


Figure 4.3: General operators of a biometric system. Based on a figure in [66]

biometric system. In our case, during the issue procedure, the authority agency will take the responsibility of enrollment.

- **Verification** This process is used to confirm the claim of the identity of a citizen. Firstly, capture biometric characteristic and produce a sample, then extract a feature set for recognition. In the recognition phase, the resulting feature set needs to be compared with the unique retrieved enrollment template (retrieved by the username or PIN provided by the citizen), then a match/non-match decision will be made by the matcher. In our case, the biometric eID card will do the verification job.
- **Identification** This process is also used to confirm identity, but there is no explicitly claim of the identity. Thus, in the recognition phase, the resulting feature set needs to be compared with several enrollment templates in the

system database, and final result will either be a confirmed identity or a non-identified decision.

Since smart cards can provide a secure and private environment to store information and to execute matching algorithms [96, 70]. We decided to execute the biometric verification process on the eID card. Fingerprint is the most common used characteristic of biometrics. With the rapid development of fingerprint recognition research and the improvement of fingerprint sensors, fingerprint recognition systems have been widely deployed in various areas, ranging from forensics to smart devices [66]. Thus, we choose fingerprint as the characteristic of biometrics for face-to-face authentication in our project. Combining with the security and privacy of smart cards, a citizen's fingerprint enrollment template and the matching algorithm are stored in the smart card. To capture and match fingerprint on the card, the biometric eID card needs to embed a fingerprint sensor.

4.5.1.3 Near Field Communication (NFC)

NFC enables contactless communication between two electronic devices. NFC is based on RFID technology and is standardized in ISO/IEC 18092 [54]. NFC makes the communication between two devices easier and more convenient with a touch, and the distance is limited up to 10cm [20]. Because of the read and write ability of NFC, it is adopted in various fields apart from smart cards. NFC-enabled smart phones are emerging in recent years, this feature of modern smart phones can be used to read data from biometric eID cards, as well as to write data into biometric eID cards during the update procedure. Thus, NFC-enabled smart phone is utilized in the implementation in Chapter 6.

4.5.2 Update

When some personal information of a citizen is changed (e.g. marital status, job status), although these information has been updated in the official database of the authority agency, the stored information of the biometric card which the citizen holds is not updated. In order to update the biometric card reliably and conveniently, the citizen can download the official application on his NFC-enabled smart phone or use other authorized terminals, then update the biometric card with his biometric data as shown in Figure 4.4, the update procedure is as following:

1. A citizen selects the update function on the authorize terminal.

2. The citizen uses his biometric eID card and the terminal to verify his biometric data.

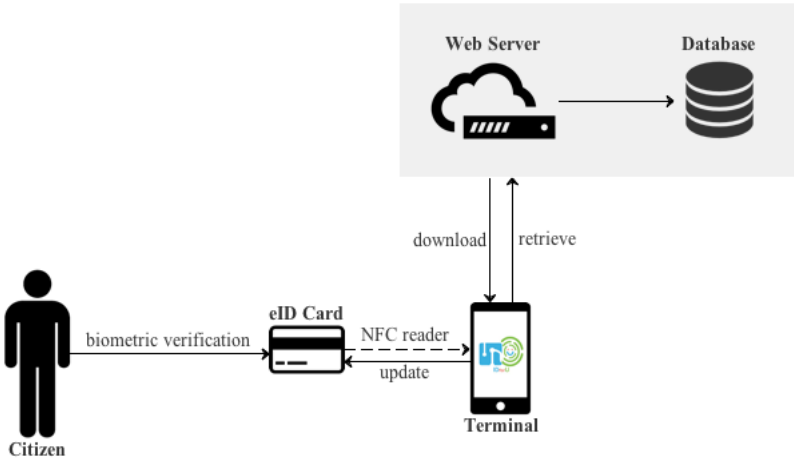


Figure 4.4: Update a biometric eID card

3. The citizen uses the terminal to read data from card and check whether the information stored in the card is updated or not.

4. If the card is not updated, the terminal then retrieves data from the official database through the official web server.

5. Rewrite the retrieved data into the biometric eID card.

4.5.3 Revoke

As stated in [10] that revocation is of central importance for eID systems. When the eID card was lost or stolen, the owner should be able to revoke the card through the authority agency.

When a citizen's biometric eID card is lost or stolen, even though biometrics verification process can protect the sensitive information in the card from social engineers, the revocation action is still necessary. In order to revoke the biometric eID card, the citizen needs to go to the authority agency and requests to revoke his biometric eID card as shown in Figure 4.5, the procedure is as following:

1. A citizen requests to revoke his biometric eID card.

2. The authority agency collects the biometric data and compares with the stored biometric data in the official database.

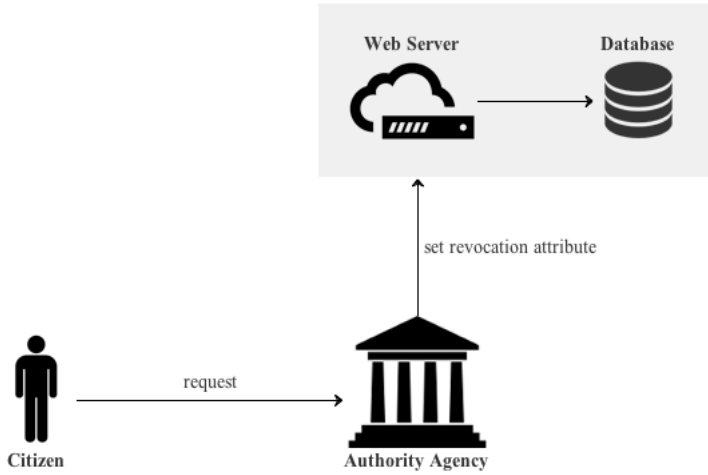


Figure 4.5: Revoke a biometric eID card

3. If the biometric data are matched, which can prove the citizen is the holder of the card need to be revoked, then set the revocation attribute for this card in the database.

After the revocation procedure, even a social engineer holds the citizen's biometric eID card and succeed in passing the biometric verification, the official application will not read data from the card after checking the revocation attribute from the official database through the official server.

4.6 Terminal Specification

The authentication terminal could be authorized card readers, authorized online applications, authorized applications on computers and smart phones. The application can provide several basic functions such as:

1. Read information from biometric eID cards.
2. Update information in biometric eID cards.
3. Face-to-face authentication with another biometric eID card holder.

The authentication terminal connects to the official server and official database, which ensures the confidentiality of the authentication procedure.

4.6.1 Terminal Certificate

To prove the authenticity of a terminal device, a terminal certificate issued by the authority agency (same authority agency as the agency issues biometric eID cards) needs to be stored in the terminal. Additionally, for the purpose of chip authentication, the ROOT certificate of the authority agency should also be stored in the terminal, we will introduce the authentication process between the terminal and the biometric eID card in Section 4.7.2.

As we assumed in Section 4.1.5, in a variety of daily scenarios, considering of risks of information disclosure, not every information of a citizen can be displayed or read. The terminal certificate supposes to limit the access rights to sensitive data, the limit is pre-defined when issuing the terminal certificate (e.g. terminal of wine shop can only read age information). For general face-to-face authentication procedure, accessible data is determined by the transaction between two parties.

4.6.2 Official Server and Database

The official server and database should be created and maintained by the authority agency, it contains information of all citizens, all biometric eID cards, as well as all transactions. The authority agency can open interfaces of the official server and database to authorize organizations. Therefore, no matter how the transaction been created (e.g. citizens booked home repair service through online application, and this application has access to the official server and database, when the transaction is created in this online application, the transaction information is also saved to the official server and database), the authentication terminal can retrieve the transaction information.

4.7 Face-to-Face Authentication

As we claimed in Section 1.2, we only focus on two face-to-face scenarios in our project. In these two scenarios, Party A is always the person who wants to verify the identity of Party B.

4.7.1 Authentication Procedure

Figure 4.6 shows general procedure when there is existing transactions between two parties, while Figure 4.7 shows the procedure when there is no transaction between two parties.

4.7.1.1 When Transaction Exists

The procedure when there is transaction between authentication parties is as following:

1. Party A verifies his biometric data on his biometric eID card.
2. The information stored in the Person A's biometric eID card is read by the terminal. Read Section 4.7.2 about the protocol specification between the biometric eID card and the terminal.
3. Terminal retrieves Party A's transaction list from the official database through the official server.
4. Party B verifies his biometric data on his biometric eID card.
5. The information stored in the Person B's biometric eID card is read by the terminal.
6. Terminal finds transaction between Party A and Party B, then displays the transaction information.

4.7.1.2 When No Transaction Exists

The procedure when there is no transaction between authentication parties is as following:

1. Party A verifies his biometric data on his biometric eID card.
2. The information stored in the Person A's biometric eID card is read by the terminal. Read Section 4.7.2 about the protocol specification between the biometric eID card and the terminal.
3. Terminal retrieves Party A's transaction list from the official database through the official server.
4. Party B verifies his biometric data on his biometric eID card.
5. The information stored in the Person B's biometric eID card is read by the terminal.
6. Terminal creates a new transaction between Party A and Party B, then displays the transaction information.

4.7.2 Protocol Specification

In the third generation specification of European ePassport, PACE (introduced in Section 3.2.3.5) is used to generate keys for secure messaging, subsequent procedures are TA (introduced in Section 3.2.3.4) and CA (introduced in Section 3.2.3.4) for mutual authentication. The security of ePassport system is significantly increased

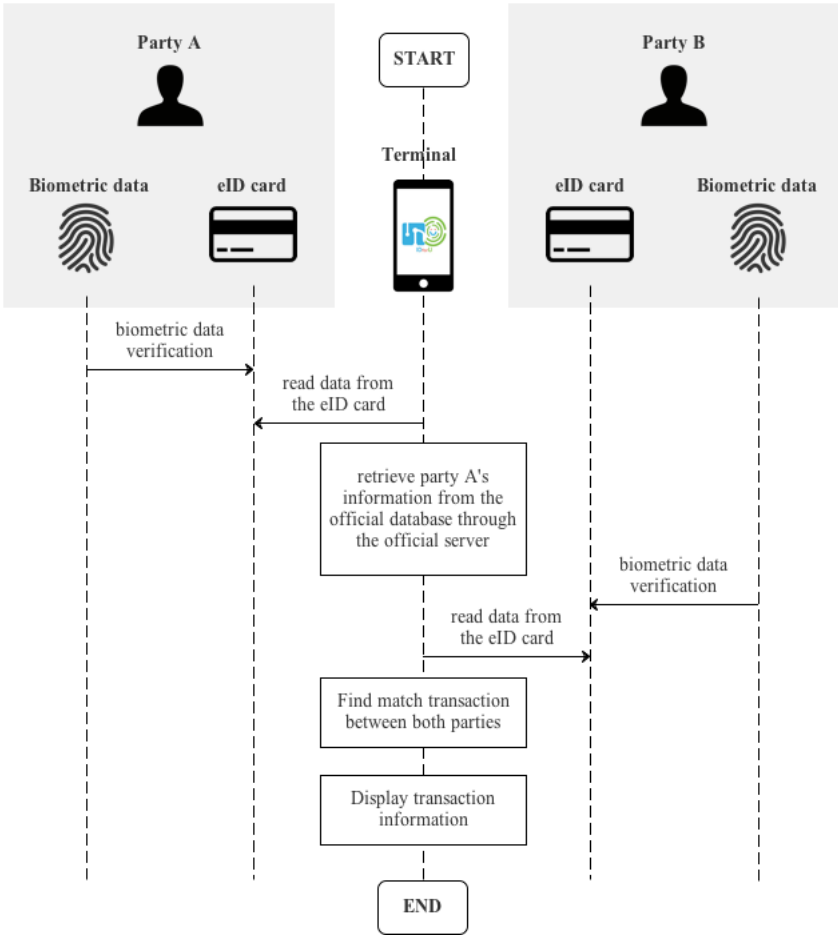


Figure 4.6: Transaction exist

through this method. Hence, based on the standard protocol used for the ePassport system, we propose the following protocol specification for the biometric eID card system.

After the match-on-card fingerprint verification passed, in order to establish a secure communication channel between an authorized eID card and an authorized terminal, keys for secure messaging (K_E and K_M) need to be generated by both parties. As we introduced, to generate keys in PACE, a common password π is read from MRZ or CAN by the reader, and π is also stored in the chip of the ePassport. However, a significant difference of biometric eID cards from ePassports is – there is no MRZ or CAN on biometric eID cards, and read a pre-store common password from the eID card before establishment of secure channel is risky. Hence, we refer to Diffie-Hellman

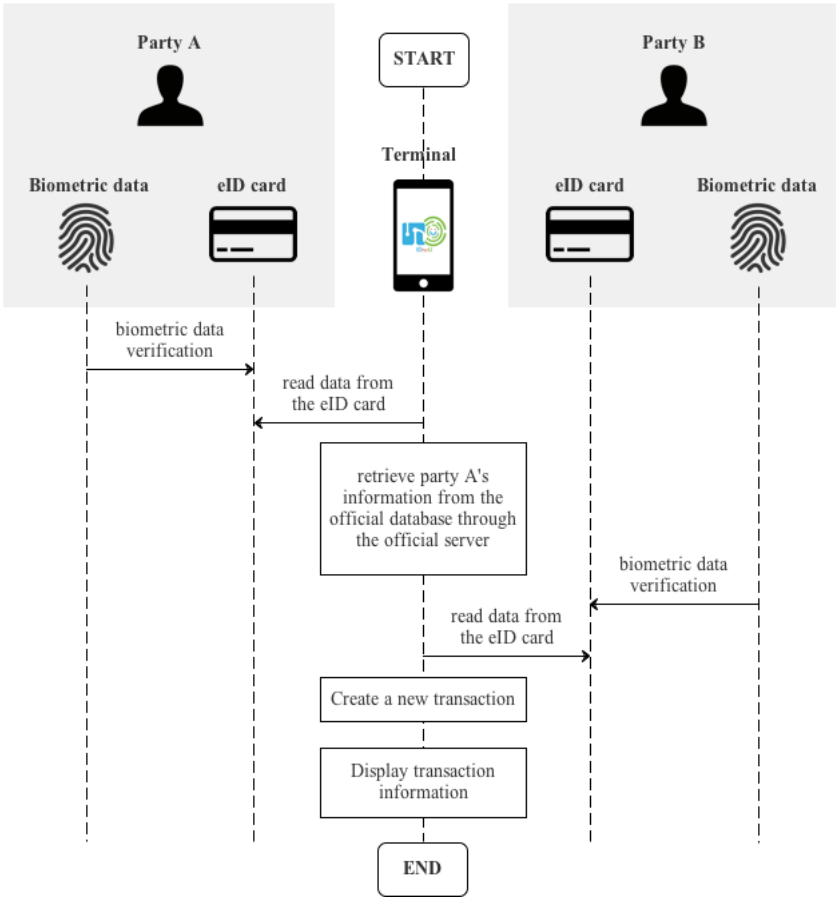


Figure 4.7: No transaction exist

key exchange[2] process to establish a shared password between two parties. Because of the discrete logarithm problem [2], the share secret can keep secret even an attacker obtains Diffie-Hellman key agreement parameters (p and g). When K_E and K_M are computed, the messages transmitted are protected those keys. Then we use the same method of ePassport system to authenticate the terminal (TA) and the eID card (CA). Table 4.2 shows each step of the protocol. The process is described as below.

We use a combination of security mechanisms for our system, and we believe it is appropriate to establish a secure communication channel under the conditions. However, we lack time and resources for formal security tests and analysis which is

[2]Wikipedia, "Diffie–Hellman Key Exchange", https://en.wikipedia.org/wiki/Diffie\T1\textendashHellman_key_exchange [Online; Accessed 07 Jun 2016]

a time-consuming specialist task. Formal security tests and analysis are desirable before adopting this protocol, so we keep this task for future work.

	Terminal		**Card**
Diffie-Hellman Key Exchange		$\xrightarrow{challenge}$	
			p, g
		$\xleftarrow{p,g}$	
			choose $CKPr \in_R [1, p-1]$
			$CKPu = g^{CKPr} \bmod p$
		$\xleftarrow{CKPu}$	
	choose $TKPr \in_R [1, p-1]$		
	$TKPu = g^{TKPr} \bmod p$		
		$\xrightarrow{TKPu}$	
	$K_{seed} = CKPu^{TKPr} \bmod p$		$K_{seed} = TKPu^{CKPr} \bmod p$
	$K_E = SHA-1(K_{seed}\|\|1)$		$K_E = SHA-1(K_{seed}\|\|1)$
	$K_M = SHA-1(K_{seed}\|\|2)$		$K_M = SHA-1(K_{seed}\|\|2)$
TA		$\xrightarrow{certificate_T}$	
			verify with ROOT certificate
			retrieve $TKPu_C$
			$r_C \in_R E$
		$\xleftarrow{r_C}$	
	$h_T = H(TKPu_C)$		
	$S_T = Sign_{TKPr_C}(h_T\|\|r_C)$		
		$\xrightarrow{S_T}$	
			$verify_{TKPu_C}(S_T)$
		$\xleftarrow{authentic}$	
		$\xrightarrow{challenge}$	
		$\xleftarrow{certificate_C}$	
	verify with ROOT certificate		
	retrieve $CKPu_C$		

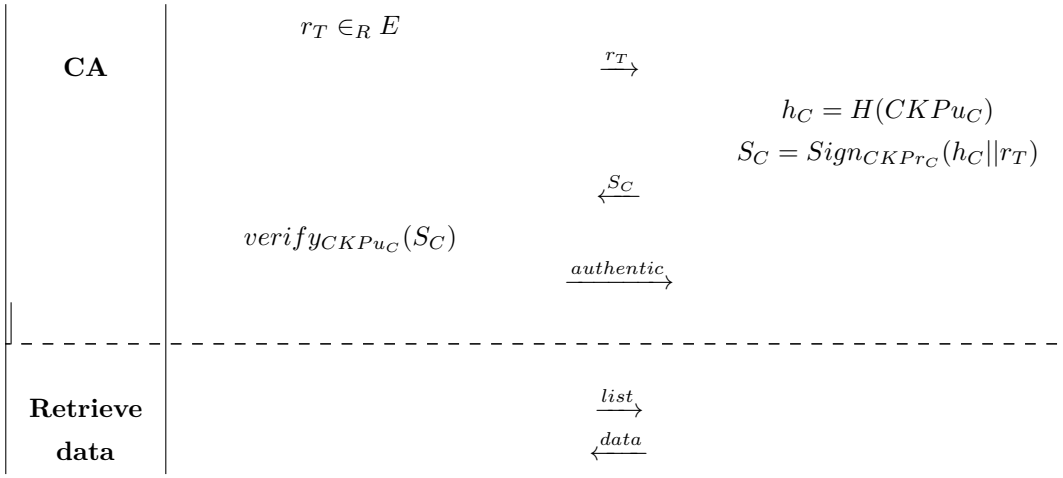


Table 4.2: Protocol specification between a eID card and a terminal after success fingerprint verification on the card

1. After the success fingerprint verification process on a citizen's eID card, he then put his eID card on the terminal.
2. First of all, the terminal sends a challenge to the card.
3. The card then generates Diffie-Hellman key agreement parameters p and g, p is prime, and g is a primitive root modulo p, then send these parameters to the terminal.
4. The card takes a random value $CKPr$ as its private key, here $1 \leq CKPr \leq p$.
5. The card computes its public key by

$$CKPu = g^{CKPr} \bmod p \tag{4.1}$$

   and sends $CKPu$ to the terminal.
6. The terminal also takes a random value $TKPr$ as its private key, here $1 \leq TKPr \leq p$.
7. The terminal then computes its public key by

$$TKPu = g^{TKPr} \bmod p \tag{4.2}$$

   and sends $TKPu$ to the card.
8. Both parties compute the common password and obtain $g^{TKPr \times CKPr} \bmod p$, which can be used as the key seed.

9. Both parties compute the encryption key K_E and the MAC key K_M. From this point on, the secure communication channel is established, all messages between the terminal and the eID card are protected by K_E and K_M.

10. Before granting the terminal access to sensitive data, the terminal needs to be authenticated first. The terminal sends its certificate to the card.

11. The card verifies the certificate with ROOT certificate stored in the card, and extracts the public key $TKPu_C$ of terminal certificate from it.

12. The card generates a random string r_C and sends it to the terminal.

13. The terminal then signs $(h_T||r_C)$ with its certificate's public key $TKPu_C$, here h_T refers the fingerprint of public key $TKPu_C$, and calculated by $h_T = H(TKPu_C)$.

14. The terminal sends the signature S_T to the card, then the card verifies the signature with $TKPu_C$. If verification succeed, the terminal can be proved authentic.

15. Similarly, the card also needs to the authenticated, in fact, the chip in the card is the part needs to be authenticated. The terminal sends a challenge to the card for card's certificate.

16. The card sends its certificate to the terminal, then the terminal verifies the certificate with ROOT certificate stored in the terminal, and extracts the public key $CKPu_C$ of card certificate from it.

17. The terminal generates a random string r_T and sends it to the card.

18. The card then signs $(h_C||r_T)$ with its certificate's public key $CKPu_C$, here h_C refers the fingerprint of public key $CKPu_C$, and calculated by $h_C = H(CKPu_C)$.

19. The card sends the signature S_C to the terminal, then the terminal verifies the signature with $CKPu_C$. If verification succeed, the card (the chip inside the card) can be proved authentic.

20. From this point on, the terminal can read data from the eID card as well as from the authorized database. For security considerations, different terminals should have different limits of authority, for example, wine shops only suppose to read age information from data, but authority agencies (police office, immigration office, government, etc.) can read all information from the card, and in daily face-to-face authentication, the accessed information is determined by the transaction. Thus, the terminal sends a list of data to the card when attempts to read the card, then the card returns data to the terminal.

4.8 Recall the Scenarios

Let's recall the five assumptions we proposed in the beginning of this chapter, whether our design effectively solve these problems in a reliable and secure way? Our design is helpless if a social engineer refuses to verify his identity or uses violent means to carry out attacks, however, as long as our design can minimize social engineering risks to a certain degree, it represents a significant improvement of the long-term fight against social engineers.

4.8.1 Scenario One

When the police officer is standing outside the door, you can requests him to show his identity by reading his biometric eID card with the official application on your NFC-enabled smart phone. Then his personal information includes his job title is shown on the screen, if he is a real policeman, you can create a new transaction between you before invite him enter the house. If his identity verification can't be passed or his job title is wrong, just refuse his entering request directly.

4.8.2 Scenario Two

When the repairman arrives, you can also requests him to prove his identity in the same method. Allow him to enter only when his identity is successfully verified. Then according to the description of the transaction (information like access limit, time spent, allowed tools, and cost), you can easily understand and control the process of the repair service.

4.8.3 Scenario Three

When the stranger wants to get on your car, you can also requests him to prove his identity in the same method. If his biometric eID card can pass the verification and the identity information is the same as he claims, create a new transaction and let him get in. Otherwise, refuse him anyway.

4.8.4 Scenario Four

When the staff wants to discuss business secrets with you, you can request him to prove his identity in the same method. Confirm his job information firstly when his identity verification is passed, then check if there are transaction between you and him (assigned by his company before the meeting). If there is no transaction about this business meeting, just end the meeting even though he is from the partner company. Otherwise, start the meeting with him according to the description of the transaction.

4.8.5 Scenario Five

When a individual shows his biometric eID card to the clerk in the wine shop, because his personal information is not printed on the card, information disclosure by peeping and memorizing is not possible. In addition, the terminal for reading biometric eID cards is issued by the authority agency, the terminal certificate in the terminal restricts that only age information can be read, other personal information will not be disclosed in this way.

Chapter 5
Previous Work

Before we started the implementation of our project, we did some research on the previous work in order to learn some experience as well as find out weaknesses from them.

5.1 Smart Identity Card

Smart Identity Card [41] is a master project done by Qingbao Guo in HiG in 2014. Qingbao Guo is also the project manager of project *IDforU*. In this project, existing ePassport authentication protocols and data protection methods were transplant into a generic smart card which can store multiple identities for multiple services. The main purpose of this project is to investigate the potential of using such ePassport

Figure 5.1: Smart Identity Card

Figure 5.2: Zwipe Access

authentication protocols in online person identity authentication with a smart identity card. Figure 5.1 is the smart cards and card reader of this project, and we will also use these devices in our project.

Comparing with project *Smart Identity Card*, The breakthrough of our project is to conclude biometrics for authentication, this action substantial increase the performance of identity authentication no matter in online or offline services, but it also bring more challenges at the same time.

5.2 Zwipe Access

Zwipe [114] is a fingerprint authentication company headquartered in Norway which develops and produce fingerprint cards for identity verification. Zwipe is one of industrial partners of project *IDforU*, their responsibility in this project is to develop a programmable fingerprint Java card. However, the ideal card wasn't being developed in time, which triggered our implementation plan.

Figure 5.2 is Zwipe Access, there are more details about Zwipe Access at [115]. Unfortunately, this card is non-programmable and the operating system is not Java but Mifare, so it is not ideal for our project.

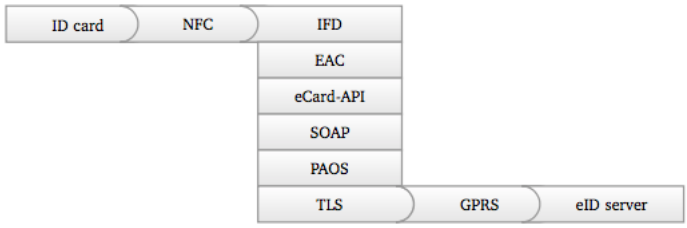


Figure 5.3: Architecture of MONA from [49]

5.3 MONA eID Client

Mobile Usage of the New German Identity Card (MONA) [49] is a Android application on NFC-enabled smartphones for using German eID cards as credentials to login web services. Figure 5.3 shows the architecture of MONA. To verify the authentic of a German eID card when a user attempts to access a web application, the procedure is described below:

1. The user opens the web application which supports German eID cards for login, then MONA will be opened when this web application performs a login process
2. MONA then establishes a Transport Layer Security (TLS) connection to the German eID server, and retrieves terminal certificate and a list of requested information which includes all data need to be read from the eID card.
3. Then the detail of the terminal certificate is displayed on the screen, the user can restrict which data groups will be sent to the eID server, and these data is agreed to be submitted when the user entering the correct PIN.
4. Before transmitting data, PACE is used to establish a secure communication channel between the smart phone and the eID card.
5. To reach the goal of mutual authentication, EAC protocol is used to authenticate both the smart phone and the eID card. The terminal certificate retrieved from the eID server is used for TA.
6. After the mutual authentication is passed, the selected data groups are read by MONA and passed to the eID server, then the user can use the web application.

The basic idea of MONA is similar to the terminal of our project, the protocols used in MONA is similar to our design introduced in Section 4.7.2. The main difference is German eID card is a common smart card without biometric sensor, using MONA

can't avoid social engineering attacks if social engineers achieve both the eID card and the PIN code. In our project, fingerprint verification is the key to prevent social engineering attacks, this is also a breakthrough comparing with project like MONA. But the system architecture and authentication process of MONA is worth referencing. In addition, MONA can only verify single eID card, but in our project, we can also verify multiple eID cards and retrieve or create transactions between them, it is much more complex than MONA.

For more details about MONA at [49], check the demonstration video of MONA and German eID card at Youtube[1].

[1]MONA - Mobile Usage of the New German ID Card (neuer Personalausweis), https://www.youtube.com/watch?v=r9vTuVN5vJw [Online; Accessed 18 May 2016]

Chapter 6
Implementation

For the purpose of verifying whether the our design has reached our expectation on usability and security, we implement the design in lab environment. In this chapter, we introduce the selection of hardware, the environment setup with software tools, as well as the specific process of implementation. The implementation aims at proving the feasibility of the whole face-to-face authentication in the specific scenarios, therefore we introduce less about performances of biometrics or NFC function, these are not the research focuses of our project.

6.1 Hardware

First of all, in order to implement our design in a lab environment, we need some hardware to support the project. Table 6.1 lists out all hardware we used for implementation, the subsections explain the reasons of choosing these hardware for our project.

Table 6.1: Hardware list

Hardware	**Brand**	**Area of application**
Smartphone	Nexus 5X [77]	• Terminal of the face-to-face authentication APP • Fingerprint scanner to capture fingerprint to do the verification • NFC reader to read data from smart cards
Smart cards	SmartCafe Expert 3.2 144K Dual [29]	Store personal information of card holder
Smart card reader	Omnikey 5321 V2 [28]	Write card holder's personal information into smart card

6.1.1 Nexus 5X

Since we haven't receive ideal fingerprint Java card from Zwipe in time, we adjusted our implementation plan to use a fingerprint smart phone instead. We did research on fingerprint scanners in different smart phones we have[1].

Table 6.2: Comparsion between different smartphones

	Fingerprint sensor	**NFC**	**Decision**
iphone 6s	Touch ID [6] • 1 / 50,000 probability of match fingerprint • only allows five unsuccessful fingerprint match attempts • doesn't store any images of fingerprint • can get fingerprint verification decision	limited to use only with Apple Pay	✗
Huawei G8	Fingerprint ID [50] • only 0.5 seconds • fingerprint verification decision is not open	enabled and no limit	✗
Nexus 5X	Nexus Imprint [76] • fingerprint data is stored securely and not shared • fingerprint capture and recognition happens in TEE • can get fingerprint verification decision	enabled and no limit	✓

From Table 6.2, it is obvious to see that iphone 6s can provide fingerprint verification decision to application, but the NFC function is limited to use only with Apple Pay, so iphone 6s is not appropriate for our project. Huawei G8 is also NFC-enabled and has no limit to use, however, the fingerprint verification decision is not open to other applications, so Huawei G8 is also not apporiate for our project either. Both fingerprint verification and NFC functions of Nexus 5X have no limit to use, so compares with iphone 6s and Huawei G8, Nexus 5X is the only appropriate smart

[1]In fact, there are more options of smart phones, check [98] to see the list of all fingerprint scanner enabled smart phones, we did comparison only based on the existing devices in our lab.

phone for the implementation, Nexus 5X uses FPC's OneTouch® FPC1025 fingerprint sensor [27] which can provide great performance of fingerprint verification.

6.1.2 Smart Cards

In our project, each party holds a smart card, so we need at least two smart cards for the implementation. The smart cards we use are the cards from project *Smart Identity Card*, which are SmartCafe Expert 3.2 144K Dual bought from [29]. It is a contact and contactless Java Card (Java Card Platform Specification 2.2.1) from Giesecke & Devrient (G&D)[2].

This card has 144K on-board EEPROM can be used for applications and data storage. It supports ISO/IEC 14443[3] and ISO/IEC 7816[4].

The platform of this card supports RSA key generation up to 2048 bits, AES key generation up to 256 bits, and Secure Hash Algorithm (SHA) key generation up to 256 bits. This card is certified to Federal Information Processing Standard (FIPS) 140-2 Level 3[5] and Common Criteria Evaluation Assurance Level (EAL) 5+[6], thus, it can provide high security level and powerful general-purpose for the implementation applications.

6.1.3 Smart Card Reader

We also need a smart card reader to write information into smart cards, the smart card reader we use is also from project *Smart Identity Card*. It is Omnikey 5321 V2 from [28]. It can both read from and write to contact and contactless smart cards, and it can work with a variety of smart cards, thus, it is suitable for us to integrate RFID technology to write data into smart cards in our project.

6.2 Software

Table 6.3 shows the different tools we used in this thesis. The main tools used to write Java Card application is Eclipse, Java Card Development Kit (JCDK) and Java Card Development Environment (JCDE) are extensions of Eclipse for writing Java

[2]Giesecke & Devrient (G&D) is a German company provides banknote and securities printing, smart cards, and cash handling systems. Official website: https://www.gi-de.com/usa/en/index.jsp [Online; Accessed 18 May 2016]

[3]Wikipedia, "ISO/IEC 14443", https://en.wikipedia.org/wiki/ISO/IEC_14443 [Online; Accessed 18 May 2016]

[4]Wikipedia, "ISO/IEC 7816", https://en.wikipedia.org/wiki/ISO/IEC_7816 [Online; Accessed 18 May 2016]

[5]Wikipedia, "FIPS 140-2", https://en.wikipedia.org/wiki/FIPS_140-2 [Online; Accessed 01 June 2016]

[6]Wikipedia, "Evaluation Assurance Level", https://en.wikipedia.org/wiki/Evaluation_Assurance_Level [Online; Accessed 01 June 2016]

Card. We use Netbeans as the tool to write Java Graphical User Interface (GUI) and web applications, Spring Boot is used to establish the web server, H2 Database is used to establish the database. In addition, we use Axure RP Pro to design the Android application and then use Android Studio to implement the application.

Table 6.3: Software tools list

Tool	**Version**	**Area of application**
Axure RP Pro [8]	7.0	Wireframing tool for designing the application on terminal
Eclipse [25]	4.2 Juno	IDE for writing Java Card applications
Java Card Development Environment (JCDE) [26]	0.2	Java Card development plugin for Eclipse
Java Card Development Kit (JCDK) [82]	2.2.2	A development environment for building Java Card applets running on smart cards using the Java programming language
GPShell [100]	1.4.4	A script interpreter for communicating with a smart card
Netbeans [74]	8.1	IDE for writing Java GUI and web applications
Spring Boot [101]	1.2.8	A set of tools for creating standalone, production-grade Spring based web applications with embedded Tomcat servers
H2 Database [42]	1.4.191	Embedded relational database, just like MySQL but in addition can be easy embedded into Java applications
Java Development Kit (JDK) [84]	1.8	A development environment for building applications and components using the Java programming language
Android Studio [104]	2.1.1	IDE for writing Android applications

6.3 Overview

Figure 6.1 shows the overview of the implementation. We establish the personalization client for “issuing” the eID card, then register the eID card by Application Protocol Data Unit (APDU) through the smart card reader. When using NFC function of Nexus 5X to read data from the eID card, it actually reads APDU. To execute the

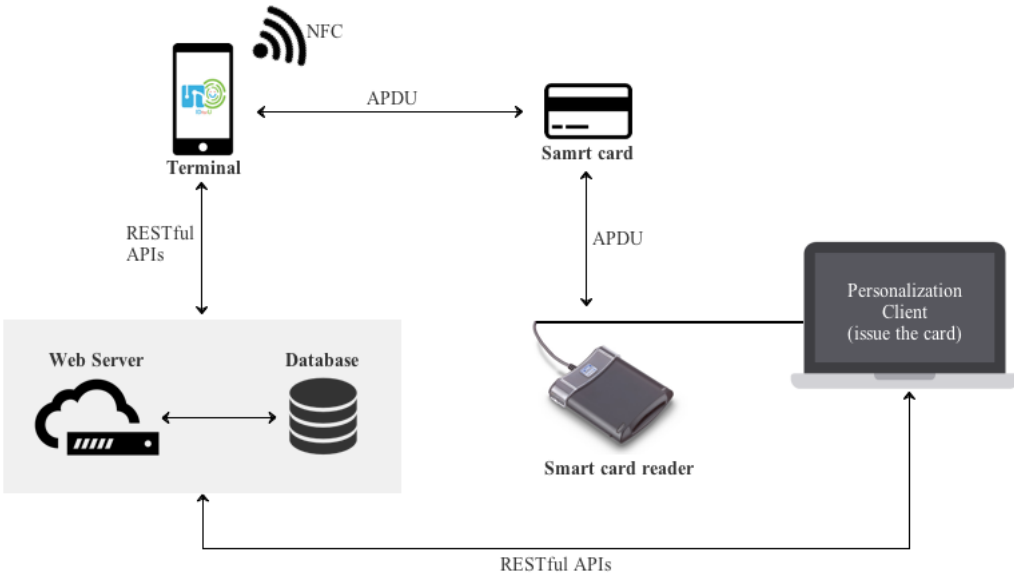


Figure 6.1: Implementation overview

authentication procedure in the Android application on the phone, it fetches RESTful Application Program Interface (API) from remote server and database, carries out fingerprint verification on the phone, and combines data from the eID card.

6.4 Environment Setup

To setup the software environment of the whole system, we started with the establishment of the web server and database.

6.4.1 Web Server

In order to provide back-end web services (RESTful APIs[7]) for terminal to read data from and write data to the database, we use Spring Boot to set up a web server. A Spring Boot application is able to have an embedded Tomcat web server, which helps us save time of configurations to speed up the implementation progress.

To run the Spring Boot application, just run the main class, i.e. *Application.class*. The main code is as following:

[7]Wikipedia, "Representational State Transfer", https://en.wikipedia.org/wiki/Representational_state_transfer [Online; Accessed 01 June 2016]

```
@Controller
@EnableAutoConfiguration
@SpringBootApplication(exclude = {SecurityConfiguration.class})
@ComponentScan("no.ntnu.idforu")
@EnableTransactionManagement
public class Application {

  @RequestMapping("/")
  @ResponseBody
  String home() {
    return "Hello IDforU!";
  }

  public static void main(String[] args) throws Exception {
    SpringApplication.run(Application.class, args);
  }
}
```

Listing 6.1: Run the Spring Boot application

Then it will automatically start the Tomcat web server and H2 database, and the back-end services are ready to be executed:

```
@RestController
@RequestMapping(value = "/transaction")
public class TransactionController {

  @Resource
  TransactionService transactionService;

  @Resource
  PersonService personService;

  @RequestMapping("/hello")
  @ResponseBody
  String person() {
    return "Hello transaction!";
  }

  @RequestMapping(value = "/all", method = RequestMethod.GET, produces =
    ↪ MediaTypes.JSON_UTF_8)
  public String list() {
    return JSON.toJSONString(transactionService.findAllTransactions());
  }
  ......
}
```

Listing 6.2: Execute back-end services

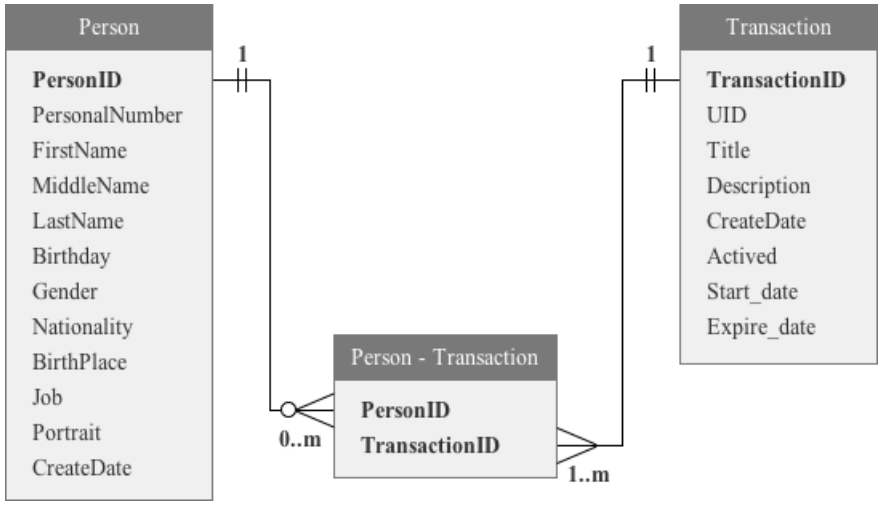


Figure 6.2: Relational Database Design

6.4.2 Database

Figure 6.2 is the relational database design of the database. We use the junction table (person - transaction) and two tables (person, transaction) to illustrate the many-to-many relationships between person and transaction. Considering of the two face-to-face authentication scenarios, the one-to-many relationship between *person* and the junction table *person - transaction* specifies that transaction amount could be 0, the reason is that when authenticate a stranger or person who have no shared transaction, then the transaction in this authentication case is 0. However, to any transaction, there is always persons involved, so the one-to-many relationship between *transaction* and the junction table *person - transaction* specifies that person amount must bigger than 1. We choose to use the embedded H2 database instead of using a standalone MySQL database server.

For the purpose of creating the whole data scheme, simply run the following SQL statements:

```
create table person_person (
  id bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  personal_number varchar(64) not null,
  first_name varchar(64) not null,
  middle_name varchar(64),
  last_name varchar(64) not null,
  birth_day date not null,
  gender varchar(8) not null,
  nationality varchar(64) not null,
  birth_place varchar(64) not null,
  job varchar(64) not null,
```

```
  portrait blob not null,
  create_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  primary key (id)
);
create table transaction_transaction (
  id bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  uid varchar(64) not null,
  title varchar(64) not null,
  description varchar(128),
  create_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  actived boolean NOT NULL DEFAULT false,
  start_date timestamp NULL DEFAULT NULL,
  expire_date timestamp NULL DEFAULT NULL,
  primary key (id)
);
create table mid_person_transaction (
  person_id bigint(20) unsigned NOT NULL,
  transaction_id bigint(20) unsigned NOT NULL,
  primary key (person_id,transaction_id)
);
```

Listing 6.3: SQL statements to create data scheme

6.5 Personalization

After the general environment setup of the web server and the database, we then need to "issue" the eID cards. In order to personalize the smart cards we have, we start with building a Java application to control the smart card reader, as well as establishing a Java Card applet on smart cards.

6.5.1 Smart Card Reader Setup

Since browsers cannot access to a smart card reader directly, we choose to follow the previous work, i.e. *Smart Identity Card*. This project developed a standard Java application that can access smart card readers in order to interact with a Java Card. Since Netbeans has better support for GUI Java applications than Eclipse, thus this previous work was done with Netbeans. Considering of time limit of this thesis, as well as the good basis provided by this previous work, we used most of the code from *Smart Identity Card* project, the code is open source that compliances with GPLv2 from Github [8]. In addition, we extended the code with additional new functionalities, e.g. http requests to the back-end web server.

To access back-end server in our Java application, using the following code for calling a POST request, then the corresponding JavaScript Object Notation (JSON) String will be returned:

[8]SmartID, Available at: https://github.com/Qingbao/SmartID [Online; Accessed 01 Jun 2016]

```
@Override
public void doPost(String url, Map<String, String> parameter) throws
    ↪ Exception {
  URL obj = new URL(url);
  HttpURLConnection con = (HttpURLConnection) obj.openConnection();

  //add reuqest header
  con.setRequestMethod("POST");
  con.setRequestProperty("User-Agent", USER_AGENT);
  con.setRequestProperty("Accept-Language", "en-US,en;q=0.5");

  String urlParameter = HttpUtil.urlParameter(parameter);

  // Send post request
  con.setDoOutput(true);
  DataOutputStream wr = new DataOutputStream(con.getOutputStream());
  wr.writeBytes(urlParameter);
  wr.flush();
  wr.close();

  int responseCode = con.getResponseCode();
  System.out.println("\nSending 'POST' request to URL : " + url);
  System.out.println("Post parameters : " + parameter);
  System.out.println("Response Code : " + responseCode);
}
```

Listing 6.4: Call a POST request

6.5.2 Java Card Applet Setup

For developing our Java Card applet, we choose to use Eclipse 4.2 Juno together with JCDE plugins, since the newest Eclipse 4.5 Mars does not have good support for JCDE plugins. For the purpose of running the application on a Java Card, Java code needs to be compiled to a *.cap* file. Then we use GPShell 1.4.4, a script interpreter that communicates with a smart card, to install a *.cap* file onto a Java Card.

To communicate with a Java Card, simply write a text file with the following contents:

```
mode_211
enable_trace
enable_timer
establish_context
card_connect
select -AID a000000003000000
open_sc -security 1 -keyind 0 -keyver 0 -key 404142434445464748494
    ↪ a4b4c4d4e4f -keyDerivation emvcps11 // Open secure channel
//show packet aid
get_status -element 20
//show applet aid
get_status -element 40
card_disconnect
```

```
release_context
```

Listing 6.5: Communicate with a Java Card

6.5.3 Write Data to Java Card Applet

Once the *.cap* file has been uploaded to our Java Card applet, we can then write the necessary data into it by send APDU. The following code shows an example of creating new files:

```
private CommandAPDU createCreateFileAPDU(short fid, short length, boolean
    ↪ eacProtection) {
  byte p1 = eacProtection ? (byte) 0x01 : (byte) 0x00;
  byte p2 = (byte) 0x00;
  int le = 0;
  byte[] data = {0x63, 4, (byte) ((length >>> 8) & 0xff), (byte) (length &
    ↪ 0xff), (byte) ((fid >>> 8) & 0xff), (byte) (fid & 0xff)};
  CommandAPDU apdu = new CommandAPDU(ISO7816.CLA_ISO7816, ISO7816.
    ↪ INS_CREATE_FILE, p1, p2, data, le);
  return apdu;
}

private byte[] sendCreateFile(SecureMessagingWrapper wrapper, short fid,
    ↪ short length, boolean eacProtection) throws CardServiceException {
  CommandAPDU capdu = createCreateFileAPDU(fid, length, eacProtection);
  if (wrapper != null) {
    capdu = wrapper.wrap(capdu);
  }
  ResponseAPDU rapdu = service.transmit(capdu);
  if (wrapper != null) {
    rapdu = wrapper.unwrap(rapdu, rapdu.getBytes().length);
  }
  return rapdu.getData();
}
```

Listing 6.6: Write data to a Java Card applet

If the Java Card applet returns *"0x9000"*, that means the file has been created. Otherwise, that might be some error has occurred. In our case, we can use the Java GUI application for writing the data to a Java Card applet, as shown in Figure 6.3 below. By pressing the *"upload"* button, then the data will be uploaded to the Java Card applet.

6.6 Terminal Implementation

In spite of various devices can be the authentication terminal, in the implementation, we use NFC-enabled smartphone – Nexus 5X as the terminal device. And the application we will introduce in this section will execute the face-to-face authentication

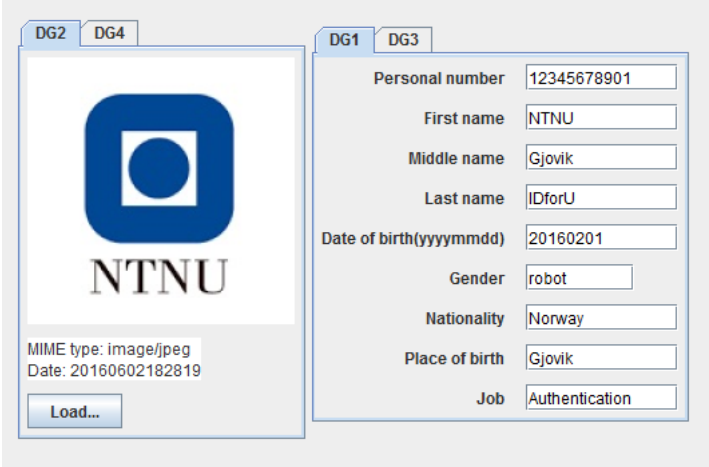


Figure 6.3: Java GUI application for writing the data to a Java Card applet

procedure. In the following sections, we use APP to represent the eID application on the phone.

6.6.1 APP Design

Figure 6.4 illustrates the design of business flow of the Android APP. It mainly used for face-to-face authentication procedure in daily life. It follows our design introduced in Section 4.7.1. One different point is the fingerprint verification process is happened in Nexus, the match-on-card fingerprint verification isn't implemented because of ideal card lack.

6.6.2 APP Development

We use Android Studio 2.1.1 to develop the Android APP. Android Studio has rich functionalities and advance GUI design tools, it is also recommended by Google official for developing Android APPs.

6.6.2.1 Fingerprint Verification Decision

Since the Zwipe fingerprint card was not ready during the implementation phase, thus, we use the fingerprint sensor on the Google Nexus 5X to simulate the fingerprint verification action. Before the new Android API level 23[9] was released, there was not an official or in other word "standard" Android API to access fingerprint sensor on smart phones. Each manufacturer has their own APIs for accessing fingerprint

[9]"android.hardware.fingerprint", https://developer.android.com/reference/android/hardware/fingerprint/package-summary.html [Online; Accessed 01 Jun 2016]

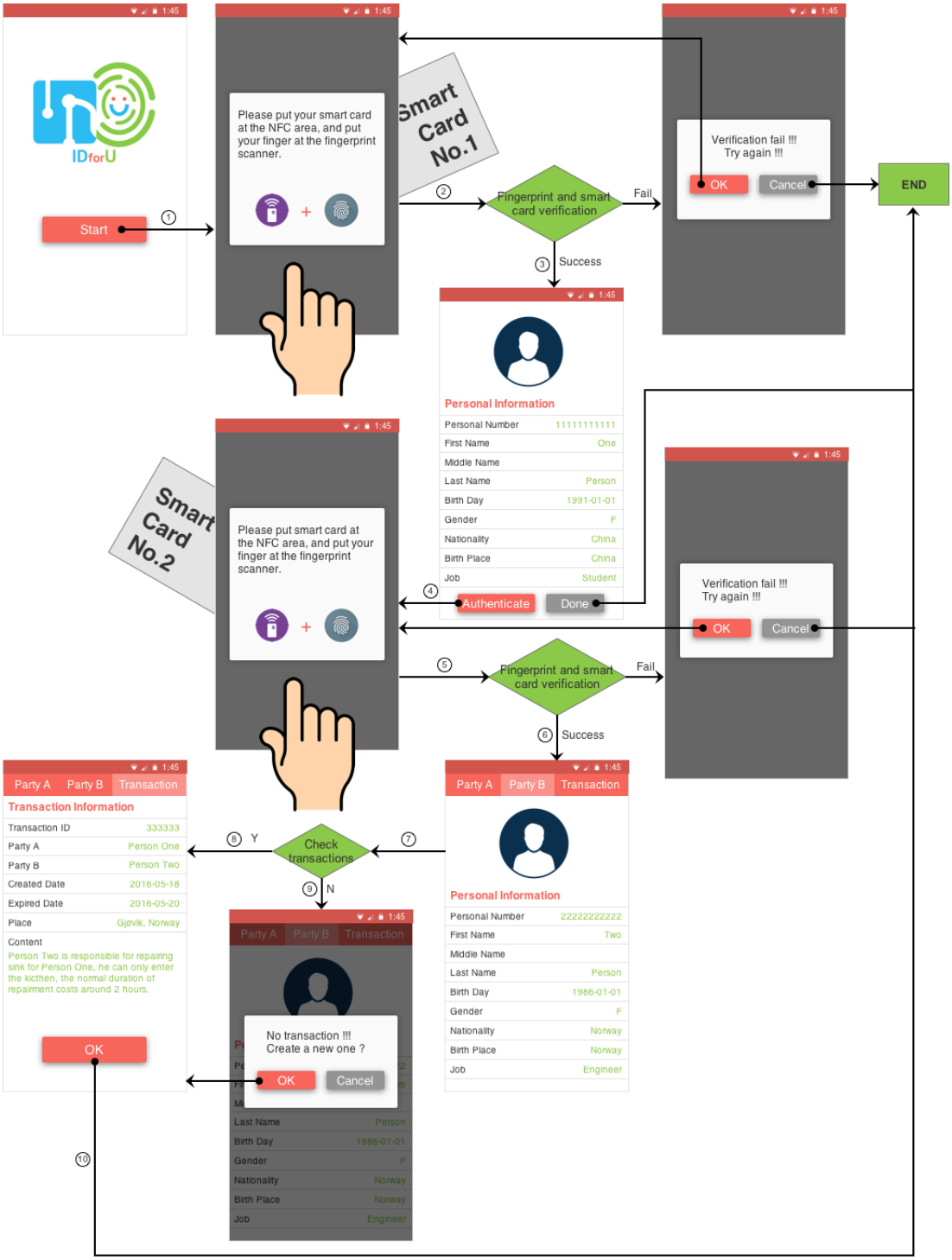


Figure 6.4: Business flow of the eID APP

sensors , for example Samsung and Huawei. One main reason of using standard API is that from long-term vision of future development, it is to follow a common API. Fortunately, Google just released the Android 6.0 Marshmallow[10] and corresponding API level 23, so we can achieve our goals.

In order to use the fingerprint sensor and get fingerprint verification decision from Nexus 5X, we need to add permission of fingerprint sensor in file *AndroidManifest.xml*, as code below shows:

```
<uses-permission android:name="android.permission.USE_FINGERPRINT" />
```

Listing 6.7: Add permission to access fingerprint sensor

Then we can get decision callback by using the following functions:

```
/**
 * Called when a fingerprint is recognized.
 * @param result An object containing authentication-related data
 */
public void onAuthenticationSucceeded(AuthenticationResult result) { }

/**
 * Called when a fingerprint is valid but not recognized.
 */
public void onAuthenticationFailed() { }
```

Listing 6.8: Get fingerprint verification decision

Apparently, for the final implementation of face-to-face authentication, the card holders need to add their fingerprints on Nexus 5X, Figure 6.5 shows the process of adding a new fingerprint to Nexus 5X.

6.6.2.2 Enable NFC Function

In our design, the main actions are occurred in face-to-face authentication with smart phones and eID cards, all data transmitted between these two devices are realized by the NFC function of the smart phone. Thus, we need to add NFC permissions when developing the APP. The following code shows the way to add permission of NFC function.

```
<uses-permission android:name="android.permission.NFC" />
    <uses-feature android:name="android.hardware.nfc" android:required="
    ↪ true" />
```

Listing 6.9: Add permission to NFC function

[10]"Android 6.0 Marshmallow", https://www.android.com/versions/marshmallow-6-0/ [Online; Accessed 01 Jun 2016]

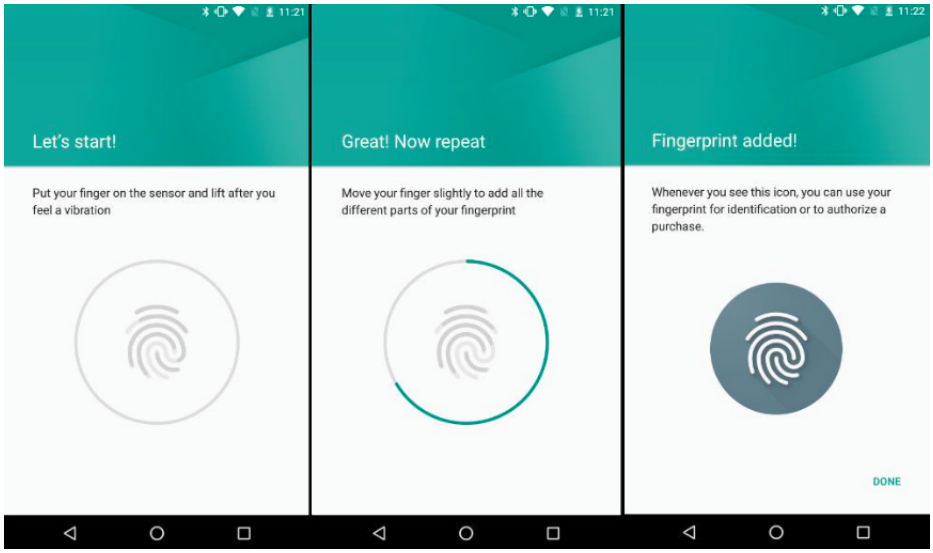


Figure 6.5: Add a new fingerprint to Nexus 5X

Generally, using NFC function to read data from a Java Card is described below. The specific implementation process of reading data from Java Card is described in the subsequent section.

```
try {
  basicService = new BasicService(IsoDep.get(nfcTag));
  reader = new Reader();
  basicService.open();
  readCard();
} catch (CardServiceException e) {
  e.printStackTrace();
}
```

Listing 6.10: Read data from the Java Card via NFC function

6.6.2.3 Read Data from a Java Card Applet via NFC Function

When a smart card is placed in the area of NFC, the NFC function should be able to discover the smart card. The following code shows the function of discovering a smart card.

```
if (NfcAdapter.ACTION_TECH_DISCOVERED.equals(action) || NfcAdapter.
    ↪ ACTION_TAG_DISCOVERED.equals(action)) {
  nfcTag = intent.getExtras().getParcelable(NfcAdapter.EXTRA_TAG);
  fingerprintDialogFragment.show(getFragmentManager(), DIALOG_FRAGMENT_TAG)
    ↪ ;
```

```
}
```

Listing 6.11: Discover a smart card via NFC function

Because of limit of time for implementation, we didn't follow the method of securing communication channel and mutual authentication in Section 4.7.2. We kept the solution in the previous work *Smart Identity Card*, which is using BAC to establish the secure communication channel, and using EAC for the mutual authentication. After passing the fingerprint verification procedure, we can establish the secure communication channel with BAC, the listing below shows the function of establishing BAC.

```
public synchronized void doBAC(byte[] keySeed) throws CardServiceException {
    try {
        if (keySeed == null) {
            return;
        }
        if (keySeed.length < 16) {
            throw new IllegalStateException("Key seed too short");
        }
        SecretKey kEnc = CryptoUtil.deriveKey(keySeed, CryptoUtil.ENC_MODE);
        SecretKey kMac = CryptoUtil.deriveKey(keySeed, CryptoUtil.MAC_MODE);
        byte[] rndICC = sendGetChallenge(wrapper);
        byte[] rndIFD = new byte[8];
        random.nextBytes(rndIFD);
        byte[] kIFD = new byte[16];
        random.nextBytes(kIFD);
        byte[] response = sendMutualAuth(rndIFD, rndICC, kIFD, kEnc, kMac);
        byte[] kICC = new byte[16];
        System.arraycopy(response, 16, kICC, 0, 16);
        keySeed = new byte[16];
        for (int i = 0; i < 16; i++) {
            keySeed[i] = (byte) ((kIFD[i] & 0xFF) ^ (kICC[i] & 0xFF));
        }
        SecretKey ksEnc = CryptoUtil.deriveKey(keySeed, CryptoUtil.ENC_MODE);
        SecretKey ksMac = CryptoUtil.deriveKey(keySeed, CryptoUtil.MAC_MODE);
        long ssc = CryptoUtil.computeSendSequenceCounter(rndICC, rndIFD);
        wrapper = new SecureMessagingWrapper(ksEnc, ksMac, ssc);
        BACEvent event = new BACEvent(this, rndICC, rndIFD, kICC, kIFD, true);
        notifyBACPerformed(event);
        state = BAC_AUTHENTICATED_STATE;
    } catch (GeneralSecurityException gse) {
        throw new CardServiceException(gse.toString());
    }
```

```
}
```

Listing 6.12: BAC establishment

To realize mutual authentication with EAC, we use code below:

```
public synchronized void doEAC(int keyId, PublicKey key, List<CVCertificate
    ↪ > terminalCertificates, PrivateKey terminalKey, String sicId) throws
    ↪  CardServiceException {
    KeyPair keyPair = doCA(keyId, key);
    byte[] challenge = doTA(terminalCertificates, terminalKey, sicId);
    EACEvent event = new EACEvent(this, keyId, keyPair,
    ↪ terminalCertificates, terminalKey, sicId, challenge, true);
    notifyEACPerformed(event);
    state = EAC_AUTHENTICATED_STATE;
}
```

Listing 6.13: EAC establishment

After the protocols (BAC and EAC) have been established, when reading data from the Java Card applet, the transmitted information will be encrypted by Triple DES. For reading a file, using *SELECT_FILE* then *READ_BINARY*:

```
private CommandAPDU createSelectFileAPDU(byte[] fid) {
    CommandAPDU apdu = new CommandAPDU(ISO7816.CLA_ISO7816, ISO7816.
    ↪ INS_SELECT_FILE, (byte) 0x02, (byte) 0x0c, fid, 256);
    return apdu;
}

CommandAPDU createReadBinaryAPDU(short offset, int le) {
    byte p1 = (byte) ((offset & 0x0000FF00) >> 8);
    byte p2 = (byte) (offset & 0x000000FF);
    CommandAPDU apdu = new CommandAPDU(ISO7816.CLA_ISO7816, ISO7816.
    ↪ INS_READ_BINARY, p1, p2, le);
    return apdu;
}
```

Listing 6.14: Read data from Java Card

6.6.2.4 Access Web Server

On the other hand, our APP also needs to access back-end APIs, fox example:

```
public void httpRequest(final String person1, final String person2) {
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
    ↪ url, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.i(TAG, response);
            if (response.equals("empty")) {
                Toast.makeText(getActivity(), "no transaction!", Toast.
    ↪ LENGTH_SHORT).show();
```

```
            } else {
                List<Transaction> list = JSON.parseObject(response, new
    ↪ TypeReference<List<Transaction>>() {
                });
                if (list != null) {
                    showData(list.get(0));
                    SharedPreferences preferences = getActivity().
                            getSharedPreferences("idforu", getActivity().
    ↪ MODE_PRIVATE);
                    SharedPreferences.Editor editor = preferences.edit();
                    editor.clear();
                    editor.apply();
                }
            }
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e(TAG, error.getMessage());
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("person1", person1);
            params.put("person2", person2);

            return params;
        }
    };

    VolleyHelper.addToRequestQueue(stringRequest);
}
```

Listing 6.15: One example of the eID APP accessing the back-end web server

6.7 Demonstration

When simulating the face-to-face authentication procedure, because of restrictions from hardware, we need to touch the fingerprint sensor and put the eID card at the NFC area at the same time, as Figure 6.6 shows. Both parties need to follow this way for fingerprint verification and for the APP reading data from the eID card.

Figure 6.7 shows when processing the fingerprint verification, the verification decision is “No”, and the verification decision is “Yes”.

After Party A is authenticated successfully, the information of Party A is displayed, then click *“AUTHENTICATION”* button, then Party B repeats same authentication

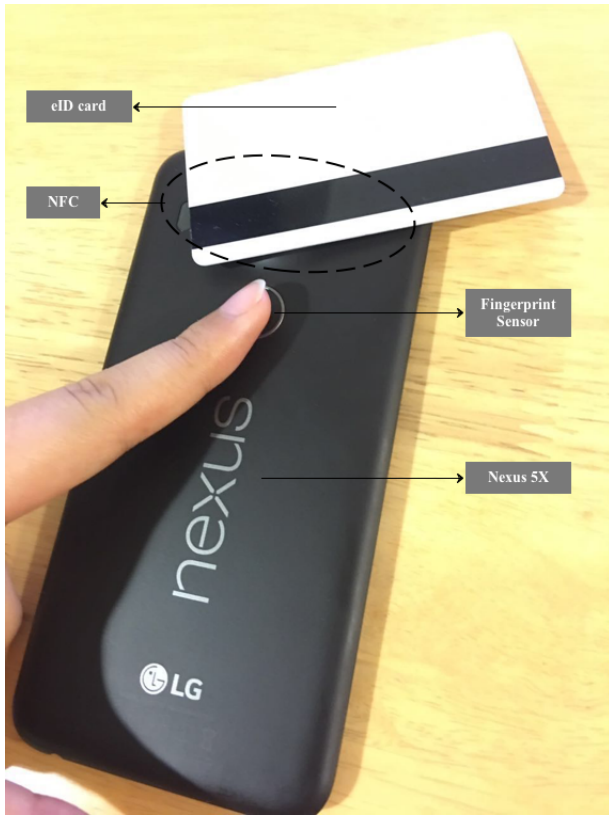


Figure 6.6: Using eID card and fingerprint sensor to verify identity

procedure as Party A did. If Party B is also authenticated successfully, the personal information is also displayed, as shown in Figure 6.8. Then the APP asks the back-end services if there has a certain transaction that links the parties by calling the RESTful API, as the code below shows.

```
1  public void httpRequest(final String person1, final String person2) {
2      StringRequest stringRequest = new StringRequest(Request.Method.POST, "/
       ↪ transaction/findPT", new Response.Listener<String>() {
3          @Override
4          public void onResponse(String response) {
5              Log.i(TAG, response);
6              if (response.equals("empty")) {
7                  Toast.makeText(getActivity(), "no transaction!", Toast.
       ↪ LENGTH_SHORT).show();
8              } else {
9                  List<Transaction> list = JSON.parseObject(response, new
       ↪ TypeReference<List<Transaction>>() {});
10                 if (list != null) {
```

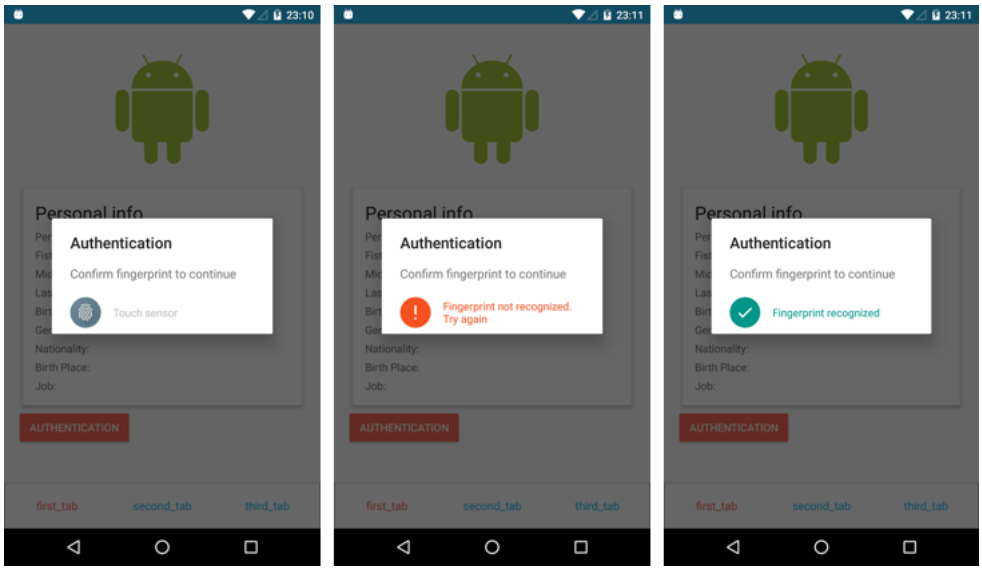


Figure 6.7: Fingerprint verification process and decision

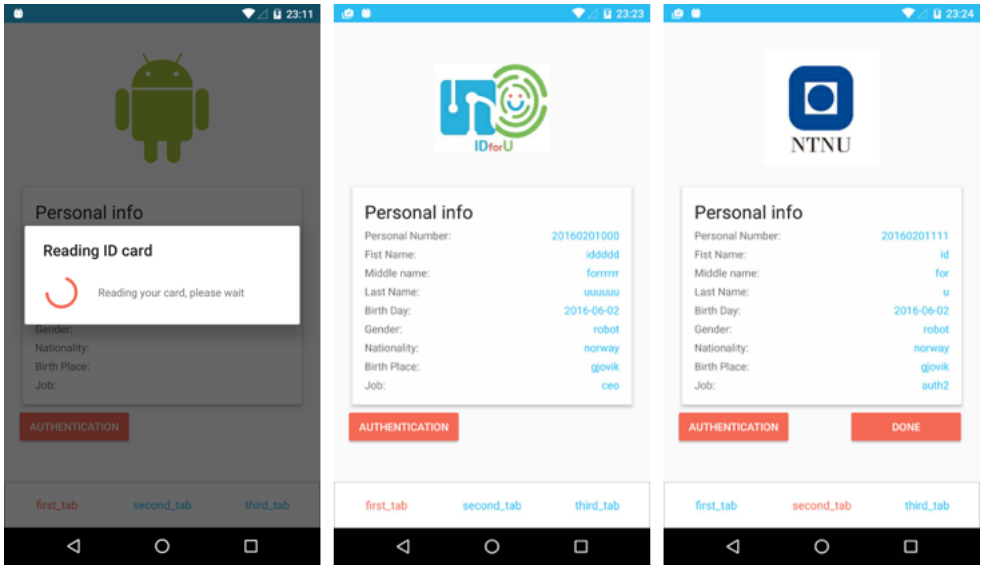


Figure 6.8: Both parties are authentic

```
                    showData(list.get(0));
                    SharedPreferences preferences = getActivity().
    ↪ getSharedPreferences("idforu", getActivity().MODE_PRIVATE);
                    SharedPreferences.Editor editor = preferences.edit();
                    editor.clear();
                    editor.apply();
                }
            }
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e(TAG, error.getMessage());
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("person1", person1);
            params.put("person2", person2);

            return params;
        }
    };
};
```

Listing 6.16: Check if there is are certain transaction link both parties

If there is a transaction between both parties, information of the transaction then is displayed as shown in Figure 6.9.

Otherwise, a new transaction will be created and saved to the database if button *"OK"* in Figure 6.10 is clicked. The code below shows how to create a new transaction by calling another API.

```
public void httpRequestCreate(final String person1, final String person2) {

    final Transaction transaction = new Transaction();
    transaction.setUid(UUID.randomUUID().toString());
    transaction.setTitle("Create from Android!");
    transaction.setDescription("for testing");
    transaction.setActived(true);
    transaction.setCreateDate(new Date());
    transaction.setStartDate(new Date());
    transaction.setExpireDate(new Date());

    StringRequest stringRequest = new StringRequest(Request.Method.POST, "/
    ↪ transaction/createPT", new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
```

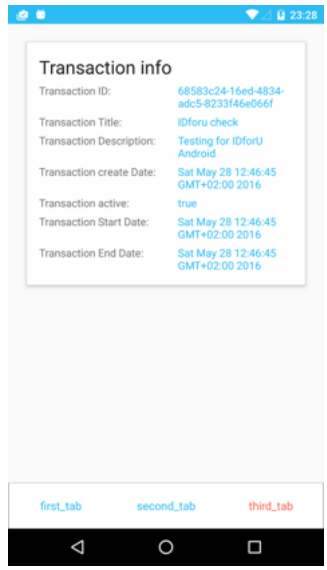


Figure 6.9: Transaction detail

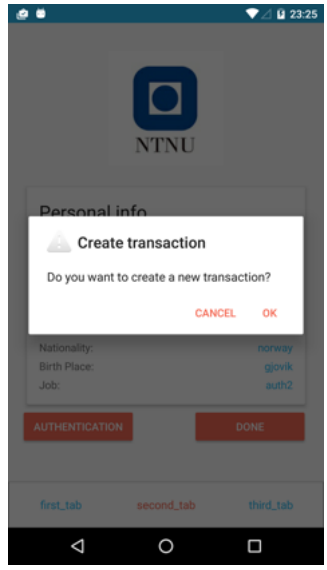


Figure 6.10: Create a new transaction when is no exist transaction links both parties in database

```
15                  Log.i(TAG, response);
16                  Transaction transaction = JSON.parseObject(response,
    ↪ Transaction.class);
17                  showData(transaction);
18                  SharedPreferences preferences = getActivity().
19                          getSharedPreferences("idforu", getActivity().
    ↪ MODE_PRIVATE);
20                  SharedPreferences.Editor editor = preferences.edit();
21                  editor.clear();
22                  editor.apply();
23              }
24          },
25          new Response.ErrorListener() {
26              @Override
27              public void onErrorResponse(VolleyError error) {
28                  Log.e(TAG, error.getMessage());
29              }
30          }) {
31
32          @Override
33          protected Map<String, String> getParams() throws AuthFailureError {
34              Map<String, String> params = new HashMap<>();
35              params.put("transaction", JSON.toJSONString(transaction));
36              params.put("person1", person1);
37              params.put("person2", person2);
38
39              return params;
40          }
41      };
42
43      VolleyHelper.addToRequestQueue(stringRequest);
44  }
```

Listing 6.17: Check if there is are certain transaction link both parties

At this point, the face-to-face authentication is completely carried out.

6.8 Performance

After several tests in the lab[11], we found several problems of the implementation. First of all, if we execute both BAC and EAC, because of the power supply issue to smart card, and complexity of EAC, we never succeed to execute both protocols, the APP always quit before executing EAC.

Secondly, when reading data from the smart card, the photograph file of card holder is much bigger file than other data files. Unfortunately, it takes over 30 minutes to

[11]Since we don't have enough volunteers for multiple tests, we used different fingerprints to represent different persons for testing.

read all information from one smart card. Thus, for better performance, we adjusted the method of reading photo. Instead of reading directly from the smart card, we read the photograph file from the back-end database through the APP.

After stopping executing EAC and reading photo from database, we can finish each identity verification and data read within 30 seconds. The transaction details were nicely retrieved and new transaction was created successfully. When modeling face-to-face authentication procedure as we demonstrated in Section 6.7, the fingerprint verification and transactions can increase security of identity verification to a certain degree. Formal security tests and analysis are necessary without a doubt, we also keep this task for future work.

Chapter 7
Security Analysis

In Chapter 4, we designed a new mechanism for minimizing social engineering risks in face-to-face transactions. Although it is efficient for verifying an individual's identity and their purposes as well as their limits of authority, please refer to the five assumptions introduced in Section 4.8. However, there are still some potential risks in the design. Coercive attacks is a type of attack which can't be avoid of the design, for example, a victim is forced to disclose sensitive information, a social engineer broke into the victim's home, a victim or a social engineer refused to execute the authentication, etc. Thus, we do not discuss such coercive attacks when analyzing the security properties of our design.

Table 7.1: Security goals

No	Security Goal	Definition
SG-1	**Strong Human Authentication**	Have high strength to authenticate identity of an individual
SG-2	**Disclosure Prevention**	Avoid information disclosure by eavesdropping, guessing, or other attacks
SG-3	**Information Limit**	Limit the amount of PII can be read and displayed to the absolute minimum, e.g. only read and display age information when buying wine
SG-4	**Linkability**	Authentication parties can be proved whether to be linked to each other, e.g. has/ no transaction between two parties
SG-5	**Forgery Resistance**	Avoid forge parties or devices, e.g. forge card, forge terminal, forge biometric

This chapter analyzes the security properties of the complete mechanism we designed in Chapter 4. Even though the primary goal of our design is to solve identity verification to prevent social engineering attacks, but for security considerations, we do not restrict to social engineering techniques or attacks, we think thoroughly from all potential threats from all kinds of adversaries.

Table 7.2: Security measures

No	Security Measure	Definition
SM-1	**TFA**	Combination of "something you have" (smart card) and "something you are" (biometrics)
SM-2	**Match-on-Card Biometric Verification**	Biometric verification process happens on the smart card
SM-3	**Secure Channel Establishment**	Establish a secure communication channel before terminal reading data from the card
SM-4	**Mutual Authentication**	Authenticate both the terminal and the smart card before terminal reading data from the card
SM-5	**Transaction Concept**	Use transaction to link two parties, design a junction table (Person - Transaction) in database, e.g. a transaction between Party A (booked home repair service) and Party B (assigned to offer the home repair service)
SM-6	**Digital Certificate**	Each card holds card certificate, each terminal holds terminal certificate, and the authority agency holds a ROOT certificate

7.1 Security Goals

Generally, the goal of the new mechanism is to minimize social engineering risks in face-to-face interactions. If we elaborate the security goals of the new mechanism, we can specify the goals as shown in Table 7.1.

7.2 Security Measures

The security measures we took when designing the mechanism are shown in Table 7.2.

The relation between the security goals and secure measures is listed in Table 7.3.

Table 7.3: Relation between security goals and measures

Security Goal	Security Measure
SM-1 (**Strong Human Authentication**)	SM-1 (**TFA**) SM-2 (**Match-on-Card Biometric Verification**) SM-5 (**Transaction Concept**)
SG-2 (**Disclosure Prevention**)	SM-2 (**Match-on-Card Biometric Verification**) SM-3 (**Secure Channel Establishment**) SM-4 (**Mutual Authentication**) SM-5 (**Transaction Concept**) SM-6 (**Digital Certificate**)
SG-3 (**Information Limit**)	SM-5 (**Transaction Concept**) SM-6 (**Digital Certificate**)
SG-4 (**Linkability**)	SM-5 (**Transaction Concept**)
SG-5 (**Forgery Resistance**)	SM-4 (**Mutual Authentication**) SM-6 (**Digital Certificate**)

7.3 Potential Threats

The potential threats may come from different aspects of the complete mechanism, including threats from smart cards, biometrics, terminals, communication channels, etc.

7.3.1 Threats to Smart Cards

According to [69, 109], there are three main threats to smart cards.

7.3.1.1 Side Channel Attacks

Small cards are small physical devices and easily carried out, so for attackers, it is not easy to physically reach smart cards, so they turned to different mechanisms of attack.

As we introduced in Section 3.2.1.2, smart cards use well-known cryptographic algorithms like Triple DES and RSA to encrypt sensitive data. Generally, the

cryptosystems of smart cards are not usually vulnerable to algorithms weaknesses because of rigorous mathematical analysis efforts [109]. However, side channel attack is a class of attacks which exploits information leaked during physical implementation of smart cards. Information like power consumption, timing information, electromagnetic emanations and so on would be extracted by attacker, then they can use these information to break the system[1]. [60] introduces the first known side channel attack against cryptographic algorithms like RSA, it exploited time cost of computations in smart cards. There are more cases of side channels which exploited power consumption [59] and electromagnetic emanations [36].

7.3.1.2 Fault Attacks

Since [12] first introduced a theoretical fault attack against the RSA with Chinese Remainder Theorem, from then on, fault attacks became a possible means of attacking smart cards. When smart card has faults, by injecting faults in algorithm during the computation of the algorithm, usually based on the effect of a 1 bit error, then the cryptosystems of the smart card is broken.

7.3.1.3 Multi-Application Security

As we introduced in Section 4.5.1.1, Java Card technology can support multiple Java applets running on a same smart card. However, this feature causes two security considerations – secure bytecode interpretation and secure resource partitioning [69]. Secure bytecode interpretation represents threats like a malicious applet writer interprets smart card security. Secure resource partitioning represents that one applet accesses the secret of other applets.

7.3.2 Threats to Biometrics

Let's refer to Figure 4.3 again, it shows the general operators of a biometric system. In the new mechanism, the enrollment of biometrics is carried out by the authority agency, thus, we do not consider attacks in the enrollment process. Hence, only the verification process has possibilities to be attacked. Since the biometric verification process is happened on the card, the sensor, feature extractor, and the matcher is one complete module on the biometric chip. According to book [14], we can sum the following threats to biometrics.

7.3.2.1 Sensor

Capturing biometric characteristic is always the first step of biometric verification process, thus, the security of biometric sensor is critical. Since the biometric verifica-

[1]Wikipedia, "Side-Channel Attack", https://en.wikipedia.org/wiki/Side-channel_attack [Online; Accessed 02 June 2016]

tion is happened on the citizen's biometric eID card, thus there are two risks in the fingerprint verification process.

1. **Spoofing attack** and **mimicry attack:** Attackers may copy the enrolled user's biometrics, then use the biometrics to fool the sensor on the card.
2. **Device substitution:** Attackers may substitute the entire legitimate biometric sensor with a illegitimate sensor to control its behavior.

7.3.2.2 Feature extractor

After biometrics is captured by the sensor, the feature extractor then extracts feature from the generated sample. However, by inserting the attackers' biometric data or replacing component, attackers may force the feature extractor to produce pre-selected features.

7.3.2.3 Matcher

When comparing the extracted feature set the only one biometric template stored in the card, social engineers may also attack the matcher by the following attacks:

1. **Manipulation of the match scores:** A social engineer captures the value of the match score after the matching process, but before the verification decision is made.
2. **Reply attack:** Attackers may inject a recorded version of the true feature set to affect the match decision.
3. **Component replacement:** Attackers may substitute the related components of the matcher in order to control its behavior.

7.3.2.4 Database

Even though the database stores biometric templates is created and maintained by the authority agency, it is still under risk if attackers can break into the database. Attackers may steal, modify, and replace biometric templates, as well as change links between ID and biometric templates. These attacks can cause serious consequences.

7.3.3 Threats to Channels

Let's refer to the secure channel establishment procedure in Section 4.7.2, we use Diffie-Hellman Key Exchange to compute a share secret as the key seed. Because of the discrete logarithm problem [2], the share secret can keep secret even an attacker

obtains Diffie-Hellman key agreement parameters (p and g). But investigations in [2] shows that Diffie-Hellman key exchange is less security than widely believed.

Chapter 8
Conclusion

In this thesis, in order to minimize social engineering risks in face-to-face interactions, a new smart-card based eID manager is proposed. By investigating social engineering attacks, and we found that how to manage the trust relationship between victims and social engineers is critical. The reason is in any social engineering attacks, victims are manipulated and they indeed trust social engineers' identities and rhetorics. Hence we then investigated several common eID solutions used for identity verification in daily life. We also designed a questionnaire and collected a lot of useful statistics about eID and technologies like smart cards and biometrics.

Based on the investigations, we adopted smart card technology and biometrics to create a new TFA mechanism of a combination of "something you have" (smart cards) and "something you are" (biometrics).

For the purpose of verifying the security of the new mechanism, we implemented a simplified system for modeling face-to-face authentication in the lab environment. Even though the hardware and technology realization restricts the performance of the final demo, from the test statistics and evaluation, we can see the potential of this mechanism.

8.1 Achievements

There are some achievement of our project:

1. We designed a mechanism for face-to-face authentication to mitigate social engineering attacks , which might become a habitual way of human authentication in various scenarios.

2. We proposed to use TFA of "something you have" (smart cards) and "something you are" (biometrics) for identity verification, and the verification process is

happened on the card. Thus, the eID card is equipped with a biometric sensor and chip.

3. We proposed to use Diffie-Hellman Key Exchange to generate key seed for secure messaging between the smart card and the terminal, this solved the problem brought from "blank card" – we do not print personal information on the card, so there is no MRZ or CAN for extraction of shared secret π for PACE.

4. We proposed the "transaction" concept into face-to-face authentication, which can enhance the authentication accuracy, as well as control limits of authorities and accessible information. It can also record any unplanned authentication actions automatically.

8.2 Limitations

There are some limitations of our project:

1. We adopted a combination of Diffie-Hellman key exchange, PACE, TA, and CA for establishing a secure communication channel. But we can't prove the security performance of the protocol we proposed for now. Thus, formal security tests and analysis are needed to verify this protocol.

2. The Java cards we used do not have fingerprint sensor, the ideal smart cards should have an accurate fingerprint sensor.

3. We used the fingerprint sensor on Nexus 5X to execute the fingerprint verification procedure, we enrolled cardholders' fingerprints on the Nexus 5X and we can only get the verification decision. Therefore, the APP can't distinguish which fingerprint belongs to which smart card, which may cause security risks.

4. Due to lack of time, we followed the implementation of previous work *Smart Identity Card*, in which protocols used for securing channel and mutual authentication are BAC and EAC. We believe that using the protocol we introduced in Section 4.7.2 is a better choice.

5. The limit of PII is not realized in the implementation, it should be controlled by terminal certificate and transactions.

6. The performance of our implementation is far from good. For example, the process duration takes about 30 minutes if the application retrieves photographs from smart cards. To optimize the user experience, there is still a lot of work to be done.

7. In order to verify the usability and security of the prototype, more formal tests and security analysis are necessary for the prototype.

8. The implementation is done in lab environment. In order to fully implement our design in real life, we need support from national and government organizations.

8.3 Future Work

For better development and optimization of our project, there are some work worth to do in the future.

8.3.1 Full Implementation

The shortages of the implementation we mentioned in Section 8.2 is one of the motivations for future work. The implementation performance reached can't fully present the pros and cons of our design. Hence, full implementation is a meaningful future work to do.

8.3.2 FIDO (Fast IDentity Online)

The Fast IDentity Online (FIDO) Alliance [4] is devoted to develop technical specifications to reduce the reliance on passwords to authenticate users, also one feature of our project. In addition, defining worldwide adoption authentication specifications is also a mission of FIDO Alliance. Thus, if our mechanism follows FIDO specifications, the realization in real world will be easier. But FIDO specifications are used for online authentication, our project mainly restricts to face-to-face authentication. Combining with FIDO specifications, our mechanism can be developed to be useful in both online and offline authentication scenarios.

Figure 8.1 shows two specifications defined by FIDO, which are Universal 2nd Factor (UAF) specification and Universal Authentication Framework (U2F) specification. Generally, in UAF specification, user can use biometric to authenticate identity on local device for accessing online service. In U2F specification, the user can use a hardware device to authenticate identity. More documents about FIDO specifications at [5].

8.3.3 Multiple Biometrics

In both design and implementation, we only use one characteristic of biometrics for authentication. However, biometric verification could fail because of various reasons. In the scenario when fingerprint verification on TouchID failed repeatedly, it would offer you a second option – password. Similar situations would also happen when using biometric cards; it would be a useful action to add an additional verification

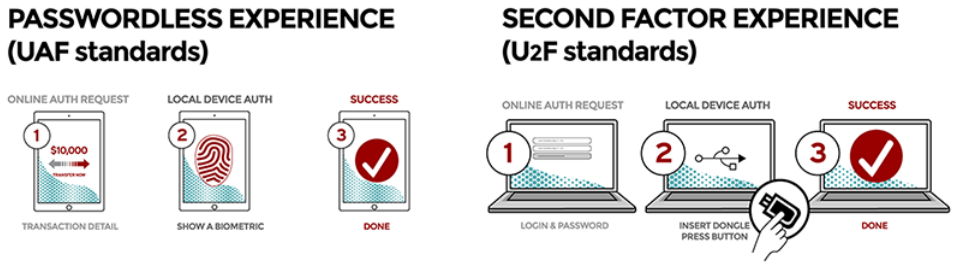


Figure 8.1: FIDO specifications from [4]

method into our mechanism. However, it brings risks at the same time. Malicious attackers would fail the more secure option on purpose, then access sensitive data by the second option easily. In the TouchID example, forging fingerprint is quite difficult but memorizing or guessing a 4-digit password is much easier. Hence, if we decide to use an additional authentication method, using another characteristic of biometrics, for instance, voice, face, iris, etc, would be a good choice, it can offer the same security as the first adopted characteristic of biometrics.

References

[1] Statens Vegvesen. Norwegian Pulic Roads Administration. "New Norwegian Driving Licence Model from January 19th 2013". http://www.vegvesen.no/_attachment/493456/binary/801364?fast_title=New+Norwegian+driving+licence+model+2013.pdf, 2013. [Online; Accessed 07 Jun 2016].

[2] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, et al. "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice". In *"Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security"*, pages 5–17. ACM, 2015.

[3] Aisha Suliaman Alazri. "The Awareness of Social Engineering in Information Revolution: Techniques and Challenges". In *"2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)"*, pages 198–201. IEEE, 2015.

[4] FIDO Alliance. "About The FIDO Alliance". https://fidoalliance.org/about/overview/. [Online; Accessed 03 Jun 2016].

[5] FIDO Alliance. "Download Specifications". https://fidoalliance.org/specifications/download/. [Online; Accessed 03 Jun 2016].

[6] Apple. "About Touch ID security on iPhone and iPad". https://support.apple.com/en-us/HT204587. [Online; Accessed 13 May 2016].

[7] APWG. "APWG Phishing Attack Trends Reports". http://www.antiphishing.org/resources/apwg-reports/. [Online; Accessed 06 Apr 2016].

[8] Axure. "Axure RP Pro Version 7.0". http://www.axure.com/download. [Online; Accessed 23 May 2016].

[9] Smart Card Basics. "Types of Smart Card". http://www.smartcardbasics.com/smart-card-types.html. [Online; Accessed 27 May 2016].

[10] Patrik Bichsel, Jan Camenisch, Thomas Groß, and Victor Shoup. "Anonymous Credentials on A Standard Java Card". In *"Proceedings of the 16th ACM Conference on Computer and Communications Security"*, pages 600–610. ACM, 2009.

[11] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pages 164–173. IEEE, 1996.

[12] Dan Boneh, Richard A DeMillo, and Richard J Lipton. "On the Importance of Checking Cryptographic Protocols for Faults". In *Advances in Cryptology—EUROCRYPT'97*, pages 37–51. Springer, 1997.

[13] Luca Calderoni and Dario Maio. "Cloning and Tampering Threats in e-Passports". *Expert systems with applications*, 41(11):5066–5070, 2014.

[14] Patrizio Campisi. "Security and Privacy in Biometrics: Towards a Holistic Approach". In *"Security and Privacy in Biometrics"*, pages 1–23. Springer, 2013.

[15] CNET. "Serious Security Flaw in OAuth, OpenID Discovered". http://www.cnet.com/news/serious-security-flaw-in-oauth-and-openid-discovered, May 2014. [Online; Accessed 20 Apr 2016].

[16] Piotr Cofta. "Confidence, Trust and Identity". *BT technology Journal*, 25(2):173–178, 2007.

[17] James S Coleman. "Social Capital in The Creation of Human Capital". *American Journal of Sociology*, pages S95–S120, 1988.

[18] Arthur W Coviello. "Open Letter to RSA Customers". http://www.validian.com/pdfs/Open-Letter-to-RSA-Customers-Mar11.pdf, 2011. [Online; Accessed 07 Jun 2016].

[19] C. Coville, Adam Wears, and Douglas A. McDonnell. "6 Military Fakes You Won't Believe Fooled the World". http://www.cracked.com/article_20191_6-military-fakes-you-wont-believe-fooled-world.html, January 2013. [Online; Accessed 01 Mar 2016].

[20] Kevin Curran, Amanda Millar, and Conor Mc Garvey. "Near Feld Communication". *International Journal of Electrical and Computer Engineering*, 2(3):371, 2012.

[21] Michael K Daly. "Advanced Persistent Threat". *"Usenix, Nov"*, 4, 2009.

[22] Rachna Dhamija, J Doug Tygar, and Marti Hearst. "Why Phishing Works". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 581–590. ACM, 2006.

[23] Diksha Dwivedi. "UAE Rape Victim 'Pardoned' – Dubai highlights Religious and Social Clash". http://theglobalpanorama.com/uae-rape-victim-pardoned-dubai-highlights-religious-and-social-clash/norwegian-passport/, August 2013. [Online; Accessed 26 May 2016].

[24] E-SEC. "Comic Strips. Make training content funny and enjoyable.". https://www.e-sec.com/en-us/products/comicstrips. [Online; Accessed 29 Feb 2016].

[25] Eclipse. “Eclipse Juno”. https://eclipse.org/juno/. [Online; Accessed 23 May 2016].

[26] EclipseJCDE. “EclipseJCDE”. http://eclipse-jcde.sourceforge.net/. [Online; Accessed 23 May 2016].

[27] fingerprints. “FPC’s OneTouch® FPC1025 Fingerprint Sensor in Google’s Nexus 5X and Nexus 6P Smartphones”. http://www.fingerprints.com/blog/2015/09/29/fpcs-onetouch-fpc1025-fingerprint-sensor-in-googles-nexus-5x-and-nexus-6p-smartphones. [Online; Accessed 13 May 2016].

[28] Smartcard Focus. “(Obsolete) Omnikey 5321 V2”. http://www.smartcardfocus.us/shop/ilp/id~436/-obsolete-omnikey-5321-v2/p/index.shtml. [Online; Accessed 17 Mar 2016].

[29] Smartcard Focus. “SmartCafe Expert 3.2 144K Dual”. http://www.smartcardfocus.us/shop/ilp/id~523/smartcafe-expert-3-2-144k-dual/p/index.shtml. [Online; Accessed 17 Mar 2016].

[30] Regionale Forskningsfond. “Regional Research Funds in Norway”. http://www.regionaleforskningsfond.no/prognett-rff-hovedside/RFF_in_English/1253976860326. [Online; Accessed 07 Jun 2016].

[31] Paul Foulkes and Anthony Barron. “Telephone Speaker Recognition Amongst Members of a Close Social Network”. *Forensic Linguistics*, 7:180–198, 2000.

[32] Alicea Francis. “5 Most Infamous Impostors”. http://www.historyanswers.co.uk/people-politics/5-most-infamous-impostors, July 2015. [Online; Accessed 10 Mar 2016].

[33] Voices from Allen’s Cross. “Elmdale Crescent”. https://allenscross.wordpress.com/streets/elmdale-crescent/. [Online; Accessed 19 May 2016].

[34] Francis Fukuyama. *“Trust: The Social Virtues and The Creation of Prosperity”*. Number D10 301 c. 1/c. 2. JSTOR, 1995.

[35] Walter Fumy and Manfred Paeschke. *“Handbook of EID Security: Concepts, Practical Experiences, Technologies”*. John Wiley & Sons, 2010.

[36] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. “Electromagnetic Analysis: Concrete Results”. In *Cryptographic Hardware and Embedded Systems—CHES 2001*, pages 251–261. Springer, 2001.

[37] Joan Goodchild. “Rogues Gallery: 9 Infamous Social Engineers”. http://www.csoonline.com/article/2358755/social-engineering/rogues-gallery--9-infamous-social-engineers.html, January 2012. [Online; Accessed 09 Mar 2016].

[38] Tyrone Grandison and Morris Sloman. “Trust Management Tools for Internet Applications”. In *Trust Management*, pages 91–107. Springer, 2003.

[39] Sarah Granger. "Social Engineering Fundamentals, Part I: Hacker Tactics". *Security Focus, December*, 18, 2001.

[40] Slade E Griffin and Casey C Rackley. "Vishing". In *Proceedings of the 5th Annual Conference on Information Security Curriculum Development*, pages 33–35. ACM, 2008.

[41] Qingbao Guo. "Smart Identity Card". Master's thesis, Gjøvik University College, 2014.

[42] H2. "H2 Database Engine". http://www.h2database.com/html/main.html. [Online; Accessed 23 May 2016].

[43] Christopher Hadnagy. *"Social Engineering: The Art of Human Hacking"*. John Wiley & Sons, 2010.

[44] Christopher Hadnagy. *"Unmasking the Social Engineer: The Human Element of Security"*. John Wiley & Sons, 2014.

[45] Christopher Hadnagy and Michele Fincher. *"Phishing Dark Waters: The Offensive and Defensive Sides of Malicious Emails"*. John Wiley & Sons, 2015.

[46] Roger Hall. *"The World of William Notman: The Nineteenth Century through A Master Lens"*. David R. Godine Publisher, 1993.

[47] Ernst Kristian Henningsen. "The Defense and Popularity of Social Engineering in Norway". Master's thesis, Gjøvik University College, 2013.

[48] Patrick Hoffman. "RSA Catches Financial Phishing Kit". http://www.eweek.com/article2/0,1895,2082039,00.asp, January 2007. [Online; Accessed 20 Apr 2016].

[49] Moritz Horsch, Johannes Braun, and Alex Wiesmaier. "Mobile eID Application for the German Identity Card". https://www.cdc.informatik.tu-darmstadt.de/reports/TR/Mobile_eID_app_for_the_German_ID_card.pdf. [Online; Accessed 29 Apr 2016].

[50] Huawei. "Huawei G8 Fingerprint ID". http://consumer.huawei.com/minisite/worldwide/g8/fingerprint.htm. [Online; Accessed 13 May 2016].

[51] ICANN. "ICANN Targeted in Spear Phishing Attack | Enhanced Security Measures Implemented". https://www.icann.org/news/announcement-2-2014-12-16-en, December 2014. [Online; Accessed 08 Apr 2016].

[52] David Icove, Karl Seger, and William VonStorch. *"Computer Crime: A Crimefighter's Handbook"*. O'Reilly & Associates Sebastopol, CA, 1995.

[53] Light Reading Inc. "Social Engineering, the USB Way". https://web.archive.org/web/20060713134051/http://www.darkreading.com/document.asp?doc_id=95556&WT.svl=column1_1, June 2006. [Online; Accessed 24 May 2016].

[54] ISO. "ISO/IEC 18092:2013". http://www.iso.org/iso/catalogue_detail.htm?csnumber=56692. [Online; Accessed 30 May 2016].

[55] Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. "Social Phishing". *Communications of the ACM*, 50(10):94–100, 2007.

[56] Markus Jakobsson and Steven Myers. *"Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft"*. John Wiley & Sons, 2006.

[57] Audun Jøsang, Claudia Keser, and Theo Dimitrakos. "Can We Manage Trust?". In *Trust Management*, pages 93–107. Springer, 2005.

[58] Ari Juels, David Molnar, and David Wagner. "Security and Privacy Issues in E-passports". In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 74–88. IEEE, 2005.

[59] Paul Kocher, Joshua Jaffe, and Benjamin Jun. "Differential Power Analysis". In *Advances in Cryptology—CRYPTO'99*, pages 388–397. Springer, 1999.

[60] Paul C Kocher. "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems". In *Advances in Cryptology—CRYPTO'96*, pages 104–113. Springer, 1996.

[61] Kaspersky Lab. "Cartoon Social Engineering". http://cyberpsychology.kaspersky.com/hp416/Social-Engineering.htm. [Online; Accessed 05 Apr 2016].

[62] lebabi.net. "Anarque Historique : Victor Lustig, l'homme qui vendit la Tour Eiffel.". http://www.lebabi.net/surlenet/anarque-historique-victor-lustig-l-homme-qui-vendit-la-tour-eiffel-170.html, September 2013. [Online; Accessed 10 Mar 2016].

[63] Dimitrios Lekkas and Dimitris Gritzalis. "e-Passports as a Means Towards the First World-wide Public Key Infrastructure". In *Public Key Infrastructure*, pages 34–48. Springer, 2007.

[64] Uintah County Library. "Hugh Richens and Douglas R. Stringfellow". http://content.lib.utah.edu/cdm/ref/collection/VE_Photos/id/914. [Online; Accessed 16 Mar 2016].

[65] Shuo Ma, Ouri Wolfson, and Jie Lin. "A Survey on Trust Management for Intelligent Transportation System". In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 18–23. ACM, 2011.

[66] Davide Maltoni, Dario Maio, Anil Jain, and Salil Prabhakar. *"Handbook of Fingerprint Recognition"*. Springer Science & Business Media, 2009.

[67] MANDIANT. "M-Trends 2015: A View From The Front Lines". 2015.

[68] Mr Ian Mann. *"Hacking The Human: Social Engineering Techniques and Security Countermeasures"*. Gower Publishing, Ltd., 2012.

[69] Konstantinos Markantonakis et al. *"Smart Cards, Tokens, Security and Applications"*. Springer Science & Business Media, 2007.

[70] Marcos Martinez-Diaz, J Fierrez-Aguilar, Fernando Alonso-Fernandez, Javier Ortega-García, and JA Siguenza. "Hill-Climbing and Brute-Force Attacks on Biometric Systems: A Case Study in Match-on-Card Fingerprint Verification". In *Carnahan Conferences Security Technology, Proceedings 2006 40th Annual IEEE International*, pages 151–159. IEEE, 2006.

[71] Kevin D Mitnick and William L Simon. *"The Art of Deception: Controlling the Human Element of Security"*. John Wiley & Sons, 2011.

[72] Paul Mutton. "PayPal Security Flaw allows Identity Theft". https://news.bbc.co.uk/1/hi/technology/3608943.stm, June 2006. [Online; Accessed 19 Apr 2016].

[73] Markus Mösenbacher. "Preventing Fraud in ePassports and eIDs". *Security protocols for today and tomorrow*, 2013.

[74] NetBeans. "NetBeans IDE 8.1 Download". https://netbeans.org/downloads/. [Online; Accessed 23 May 2016].

[75] BBC News. "Phishing Con Hijacks Browser Bar". http://news.netcraft.com/archives/2006/06/16/paypal_security_flaw_allows_identity_theft.html, April 2006. [Online; Accessed 20 Apr 2016].

[76] Nexus. "Fingerprint Security on Nexus Devices". https://support.google.com/nexus/answer/6300638?hl=en. [Online; Accessed 13 May 2016].

[77] Nexus. "Nexus 5X". https://www.google.no/nexus/5x/. [Online; Accessed 13 May 2016].

[78] Rishab Nithyanand. "A Survey on the Evolution of Cryptographic Protocols in ePassports". *IACR Cryptology ePrint Archive*, 2009:200, 2009.

[79] Liz O'Connell. "Report: Email Phishing Scam Led to Target Breach". http://bringmethenews.com/2014/02/12/report-email-phishing-scam-led-to-target-breach, February 2014. [Online; Accessed 08 Apr 2016].

[80] Federal Office of Information Security. "Advanced Security Mechanisms for Machine Readable Travel Documents –Extended Access Control (EAC)". Feburary 2008.

[81] Bernard Oosterloo. "Managing Social Engineering Risk: Making Social Engineering Transparent". Master's thesis, Gjøvik University College, 2008.

[82] Oracle. “Java Card Development Kit 2.1.2”. http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javame-419430.html#java_card_kit-2.2.2-oth-JPR. [Online; Accessed 23 May 2016].

[83] Oracle. “Java Card Technology”. http://www.oracle.com/technetwork/java/embedded/javacard/overview/default-1969996.html. [Online; Accessed 24 May 2016].

[84] Oracle. “Java SE Development Kit 8 Downloads”. http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html. [Online; Accessed 23 May 2016].

[85] International Civil Aviation Organization. “Machine Readable Travel Documents: PKI for Machine Readable Travel Documents offering ICC Read-Only Access”. October 2004.

[86] International Civil Aviation Organization. “Updated Technical Report Supplemental Acceess Control”. May 2014.

[87] Gregory L Orgill, Gordon W Romney, Michael G Bailey, and Paul M Orgill. “The Urgency for Effective User Privacy-education to Counter Social Engineering Attacks on Secure Computer Systems”. In *Proceedings of The 5th Conference on Information Technology Education*, pages 177–181. ACM, 2004.

[88] C. Enrique Ortiz. “An Introduction to Java Card Technology”. http://www.oracle.com/technetwork/java/javacard/javacard1-139251.html, May 2003. [Online; Accessed 24 May 2016].

[89] Bimal Parmar. Protecting against spear-phishing. *Computer Fraud & Security*, 2012(1):8–11, 2012.

[90] Sara Peters. “The 7 Best Social Engineering Attacks Ever”. http://www.darkreading.com/the-7-best-social-engineering-attacks-ever/d/d-id/1319411, March 2015. [Online; Accessed 08 Apr 2016].

[91] Andreas Poller, Ulrich Waldmann, Sven Vowé, and Sven Türpe. “Electronic Identity Cards for User Authentication—Promise and Practice”. *IEEE Security & Privacy*, (1):46–54, 2012.

[92] Bart Preneel and Paul C van Oorschot. “Key Recovery Attack on ANSI X9. 19 Retail MAC”. *Electronics Letters*, 32(17):1568–1569, 1996.

[93] Robert D Putnam. “Tuning in, Tuning out: The Strange Disappearance of Social Capital in America”. *PS: Political Science & Politics*, 28(04):664–683, 1995.

[94] Dimensional Research. “The Risk of Social Engineering on Information Security: A Survey of IT Professionals”. September 2011.

[95] RSA. “Anatomy of An Attack”. https://blogs.rsa.com/anatomy-of-an-attack. [Online; Accessed 08 Apr 2016].

[96] R Sanchez-Reillo, Luis Mengibar-Pozo, and C Sanchez-Avila. "Microprocessor Smart Cards with Fingerprint User Authentication". *Aerospace and Electronic Systems Magazine, IEEE*, 18(3):22–24, 2003.

[97] Fred B. Schneider. "Something You Know, Have, or Are". https://www.cs.cornell.edu/courses/cs513/2005fa/nnlauthpeople.html. [Online; Accessed 15 Mar 2016].

[98] Shams. "List of All Fingerprint Scanner Enabled Smartphones". http://webcusp.com/list-of-all-fingerprint-scanner-enabled-smartphones/, Apr 2016. [Online; Accessed 18 May 2016].

[99] Anshuman Sinha. "A Survey of System Security in Contactless Electronic Passports". *International Journal of Critical Infrastructure Protection*, 4(3):154–164, 2011.

[100] sourceforge. "GPShell 1.4.4". https://sourceforge.net/projects/globalplatform/files/GPShell/GPShell-1.4.4/. [Online; Accessed 03 Jun 2016].

[101] Spring. "Spring Boot". http://projects.spring.io/spring-boot/. [Online; Accessed 23 May 2016].

[102] Mark Stanislav. "Two-Factor Authentication". 2015.

[103] Debbie Stephenson. "Spear Phishing: Who's Getting Caught?". http://www.firmex.com/thedealroom/spear-phishing-whos-getting-caught/, May 2013. [Online; Accessed 12 Apr 2016].

[104] Android Studio. "Android Studio The Official IDE for Android". https://developer.android.com/studio/index.html. [Online; Accessed 23 May 2016].

[105] Colin Tankard. "Advanced Persistent Threats and How to Monitor and Deter Them". *Network Security*, 2011(8):16–19, 2011.

[106] Tetraph. "Covert Redirect Vulnerability Related to OAuth 2.0 and OpenID". http://tetraph.com/covert_redirect/oauth2_openid_covert_redirect.html, May 2014. [Online; Accessed 20 Apr 2016].

[107] Tim Thornburgh. "Social Engineering: The Dark Art". In *Proceedings of the 1st annual conference on Information security curriculum development*, pages 133–135. ACM, 2004.

[108] The New York Times. "Data Breach at Security Firm Linked to Attack on Lockheed". http://www.nytimes.com/2011/05/28/business/28hack.html, May 2011. [Online; Accessed 08 Apr 2016].

[109] Michael Tunstall. "Smart Card Security". In *"Smart Cards, Tokens, Security and Applications"*, pages 195–228. Springer, 2008.

[110] Umut Uludag, Sharath Pankanti, Salil Prabhakar, and Anil K Jain. "Biometric Cryptosystems: Issues and Challenges". *Proceedings of the IEEE*, 92(6):948–960, 2004.

[111] Verizon. “2013 Data Breach Investigations Report”. 2013.

[112] Jamie White. “Dumpster Diving”. https://www.lifelock.com/education/dumpster-diving/#/results/, November 2013. [Online; Accessed 24 May 2016].

[113] Michael Winter. “Home Depot Hackers Used Vendor Log-on to Steal Data, E-mails”. http://www.usatoday.com/story/money/business/2014/11/06/home-depot-hackers-stolen-data/18613167, November 2014. [Online; Accessed 08 Apr 2016].

[114] Zwipe. “About Zwipe”. http://zwipe.com/company. [Online; Accessed 18 May 2016].

[115] Zwipe. “Zwipe Access: 13.56 MHZ”. http://zwipe.com/products. [Online; Accessed 18 May 2016].

Appendix A

IDforU Introduction

Prosjektbeskrivelse

Contact: Dr. Bian Yang, associate professor at NISlab, Høgskolen i Gjøvik

Email: bian.yang@hig.no Tel.: +47 611 35 486 (office) +47 41 38 70 92 (cell phone)

Tittel: Benchmarking Private eID Solutions for Understanding End Users' Needs (IDforU)

1. Introduksjon

People living in this millennium have to manage an increasing volume[1-3] of e-identities (eID) to gain access to physical (door, smart devices, *etc.*) and digital (email, social media, online shopping / banking, *etc.*) resources. The high penetration rate of ICT in Nordic countries exacerbates the challenge. This trend will be difficult to reverse, if not exacerbated, by Internet of Things (IoT) in the next decades with large amounts of smart devices to be identified for their ownership and rights configuration [22] for coordinated tasks. Private eIDs are one of the weakest links in security management due to human factors while mismanagement of eIDs can have serious consequences such as identity theft[4,5] and data breach[6-8] by spear phishing, *e.g.*, exploiting a compromised email account from a trusted colleague or service provider [18][23].

While eID management have been extensively studied[9-12] from the identity and service providers' perspectives, not so many solutions can be called "private eID management" addressing the needs from end users' personal perspective such as usability, cost, and privacy concerns in a holistic way – *i.e.*, **what do end users really want and need**? The solutions closest to this end could be the online password managers[13], single-sign-on[14,15], and outsourced online authentication[16,17], but they are just another solution and not compatible to each other, which force users to choose them exclusively. On the other hand, **to improve eID management's efficiency and trustworthiness, we need to understand not only the technologies but the people too.** However, there lacks such a requirement analysis from different user groups, especially from **juvenile, elderly, and physically or mentally challenged end users** in Norway.

A desirable private eID management solution should maximally **leverage users' control** (**user-defined**, *e.g.*, add, renew, revoke, update, and independently protect all eID profiles) be well protected when a service provider is compromised [7]. Furthermore, private identity management should **imply trust management** such as remotely delegating ID authentication for a family member, or configuring authentication in varied trust levels[18] (*e.g.*, via public or personal devices). Existing private eID solutions rarely considered such contexts. It would be **interesting to know how trust management can be efficiently integrated to private eID management, and whether they can be defined and configured under the users' control**?

To answer the above questions, IDforU plans research efforts on user requirement analysis, criteria and metrics generation, eID solutions benchmarking, and concept designs with possibly prototyping towards a future private eID management architecture featured with user definability and trust management.

2. Prosjektmål

As a "forprosjekt", IDforU has multiple aims: (1) create knowledge and concept designs with enough Technology Readiness Level for research topics in EU H2020. (2) provide guidelines for Norwegian eID industry and organizations to develop competitive private eID solutions and promote their participation to EU research programs. (3) provide guidelines for Norwegian end users on managing their private eIDs.
Task 1: Requirement analysis from different user groups. **Deliverable 1**: a set of criteria and metrics to describe the user requirements in a quantitative and operable way, and user groups' definition.

Task 2: Benchmarking typical eID management solutions. **Deliverable 2.1**: testing platform establishment: implementation of lab and real-life testing scenarios, testing tools and protocols. **Deliverable 2.2**: eID solutions benchmarking results over different user groups with SWOT analysis.
Task 3: Generating guidelines in private eID management for Norwegian eID technology providers and personal end users. **Deliverable 3**: Development guideline for private eID management solutions.
Task 4: Concept designs and innovations towards a future private eID management architecture.
Deliverable 4: a final project report recording project findings and new concept designs.
A list of potential main research projects extending IDforU is given as follows (the bold ones are prioritized).

Main research project target	Topics	Application deadline
FTIPilot-2015-1	**Fast track to innovations**	01 December 2015 and sometime 2016
H2020 SCI-PM-12-2016	PCP-eHealth innovations	16 February 2016
H2020 DS-03-2016	Digital security of health-related data	16 February 2016
H2020 DS-01-2016	**Assurance and certification for trustworthy and secure ICT**	12 April 2016
H2020 DS-06-2017	Cryptography	25 April 2017
H2020 DS-08-2017	**Privacy, data protection, and digital identifiers**	24 August 2017
FETOPEN-01-2016-2017	Future emergent technologies	11 May 2016, 17 January and 27 September 2017

3. Problemstillingen (e)

The project idea was based on the following problems challenging the existing eID management solutions:

1. The number of eIDs grows quickly for an ordinary end user to manage. While identity theft and privacy breach threats make eID management unprecedentedly crucial, there lacks enough research done to analyze the requirements (especially the usability and privacy aspects), and benchmark those typical market solutions from the user perspective over different user groups.
2. Facing so many eID solutions, an ordinary end user have no operable guideline to follow in order to choose a best-suited solution instead of a most market-influenced one. This is especially the case for the lack of security and privacy guidelines which may cost a user dearly in privacy and economy.
3. Most existing private eID solutions are exclusive in use and hard to manage eIDs across the applications or configurable in a user-defined way.
4. Existing private eID solutions are single-user based and disregard the trust level of the authentication context. Whether trust management functionalities are achievable on top of private eID management is worthy of investigation.

Our solution is to benchmark existing eID solutions in the market and public literature based on user requirement analysis over different user groups, and then explore the possibility of a general private eID management architecture concept design from the benchmarking results. Tentative methodologies include

1. Generating user requirements via both **subjective evaluation** (wish/concern list, review of use, *etc.*) and **objective evaluation** (measuring and analyzing the human-system interaction via visual-audio sensors and HBSI[19] model for biometrics-based solutions).
2. Iterative refinement of the user requirement analysis by taking as the objective evaluation input the intermediate benchmarking results.
3. Cohorts based study to test the hypothesis of the performance improvement of the concept designs for a user-defined private eID management architecture, against the generated criteria and metrics developed from the user requirement analysis results
4. Preliminary survey of national and international laws and regulations as another dimension to user requirement analysis

Potential risks to the project targets listed in the Section 2 as follows.

1. Difficult to recruit volunteered participants to form user groups (middle level risk)

Mitigation plan -> (1) establish collaboration with the EU project PIDaaS[17] for data collection from Italy, Lithunian, or Spain for at least the testing of the PIDaaS eID solutions; (2) Zwipe, as the project partner, will assist in participants' recruitment via its marketing channels; (3) GUC's student and staff resource will be motivated for user group formation; (4) exploiting the successful experience in volunteer recruitment learnt from the RFF project CrowdAir.

2. Too divulged user requirement analysis results (low level risk)
   Mitigation plan -> it is reasonable to check if the grouping criteria did not characterize the people well; if so, re-consider and define the grouping criteria and re-group the participants
3. Difficulty in evaluating the security aspect due to the diversity of solutions (middle level risk)
   Mitigation plan -> refer to NISlab and GUC's EU project partner KU Leuven (Belgium) for acquiring consultancy service if necessary
4. Not enough budget to acquire all widely-adopted eID solutions for testing purposes (high level risk)
   Mitigation plan -> as a "forprosjekt", IDforU will start from the free eID solutions such as single-sign-on provided by Gmail, Yahoo, Facebook, and then KeePass/LastPass and Norwegian eID solutions such as FEIDE, bankID, buypass, and Zwipe's fingerprint ID card product. If the resource permits, we will continue to PayWave, PayTag, Yubikey and the prototype in EU project PCAS[20].

The RFF project funding, as a whole, is expected to provide financial resource to the above mitigation plans.

4. Prosjektgjennomføring

a. Informasjonsinnhenting/datainnsamling

In the project, we plan to collect the following data / information

1. Test-oriented user credentials (account and password/hard token/biometrics) for user eID use habit analysis (e.g., password patterns and use habits), which are created only for research use.
2. User-ID system interaction logs (e.g., user use behavior, user movement, time to complete a full ID authentication session, and incident responses under both announced and unannounced tests)

For privacy consideration, we shall not collect more data than enough for the data analysis purpose. For information storage, we shall use physical access control, encryption, and possibly biometric template protection mechanism to protect the data collected. Such infrastructure can be provided by Gjøvik University College (the proposal coordinator). GUC has privacy-related data collection authorization for research purpose from NSD (Norsk samfunnsvitenskapelig datatjeneste). For unannounced tests, a separate application to NSD will be made.

b. Analyse av data

All user credential data are used to check with the users' group characteristics and use patterns and system logs for usability evaluation. Credential data generated from different eID solutions are to be analyzed to estimate the security and privacy-preserving levels.

c. Prosjektgruppe

The consortium consists of three project partners with strong expertise in each professional fields:

1. Gjøvik University College (GUC): identity management, privacy enhancing technologies, biometrics, user experience analysis, information security
2. Zwipe AS (Zwipe): fingerprint based wireless ID card technologies
3. Ko-Aks AS (Koaks): technology solutions survey and innovation planning

Coordinator (project main applicant): Gjøvik University College (GUC)
Project Manager: Bian Yang (GUC)
Partner institution (invited project partner): Zwipe and Koaks

Task assignment is planned as follows.

1. User grouping and requirement analysis
   Task leader: Bian Yang (GUC)
   Participants: Ådne Midtlin (Koaks), Edlira Martiri (GUC), Jingjing Yang (GUC), Miriam Begnum (GUC)
2. Benchmarking typical eID management solutions
   Task leader: Bian Yang (GUC)
   Participants: Pawel Dworzecki (Zwipe), a research engineer to be recruited (GUC)
3. eID management guideline generation
   Task leader: Pawel Dworzecki (Zwipe)
   Participants: Bian Yang (GUC), the research engineer to be recruited (GUC)
4. New concept design for future user-defined eID solution architecture
   Task leader: Bian Yang (GUC)
   Participants: Pawel Dworzecki (Zwipe), the research engineer to be recruited (GUC), Ådne Midtlin (Koaks)

d. Tidsplan og milepæler

Milestones for IDforU are planned as follows.

1. User grouping and first draft of user requirements (M1)
2. Criteria and metrics for benchmarking eID solutions (M2)
3. eID solution test list for benchmarking (M3)
4. Benchmarking result (M4)
5. Second draft of user requirements and eID management guideline (M5)
6. New concept designs towards future private eID management architecture (M6)

Arbeidsoppgave	Tidsperiode													
	j	f	m	a	m	j	j	a	s	o	n	d		
Task 1		M1		M2										
Task 2		M3								M4				
Task 3												M5		
Task 4												M6		

5. Budsjett og finansieringsplan

Please refer to the detailed budget in the online application form. Several notes are given as follows.

1. We realize while the funding is limited, it should be enough for the preliminary eID solutions benchmarking purposes.
2. We shall fully exploit this “forprosjekt” funding for laying a basis to main research projects to be applied for in the next 12 months (see planned targets in Section 2).
3. One research engineer or research assistant (10 PM) is to be recruited in GUC for Task 2-4.

6. Prosjektkrav

GUC has a very strong research experience in information security, privacy enhancement, and ID management fields and have contributed to several EU, national, and externally-funded projects (TURBINE, FIDELITY, NIST-BTP-Metrics, PIDaaS, SWAN[21]) since year 2008. However, **there so far lacks enough collaboration with local industries in this research field**. IDforU shall pave a way to establishing concrete collaboration between GUC and Norwegian eID technology providers such as Zwipe AS towards a future eID concept design. Zwipe demonstrated strong innovation capability in biometrics and ID management research and development and can well complement GUC on this research topic and Koaks has expertise in

innovation planning and technology survey. The collaboration is supposed to establish an academic-industry collaboration model for ICT innovations. The proposal fits the RFF's "forprosjekt" well since it aims at synergizing regional academic and industrial expertise to generate knowledge which is crucial in Norway and the whole EU scope but has not been available to technology providers and end users.

7. Strategi – ambisjoner

By the IDforU project, we aim at exploring a long-term research collaboration between GUC, which has NISlab and CCIS with strong expertise in information security fields, and local industries towards wider future research collaboration, technology innovation, and student training.

On the other hand, via IDforU, GUC would act as a bilateral agent: (1) to **transfer its research expertise and knowledge gained from the EU** and international research projects on biometrics and ID management **to Norwegian industries** in a technology transfer way or a collaborated research way; (2) **link more national and local industries to EU** and global partners in research and development collaboration (the NFR funded SWAN project, which differs in topic focus from IDforU, has stroke the first move towards this goal).

By the IDforU project, we aims at a main research project (see potential targets listed in Section 2) with enhanced resources to test the newly-created knowledge the new concept designs developed from IDforU based on those eID benchmarking criteria and metrics developed in IDforU.

8. Referanser

[1] D. Florencio and C. Herley, " A Large-Scale Study of Web Password Habits," Proc. of WWW 2007.
[2] No wonder hackers have it easy: Most of us now have 26 different online accounts - but only five passwords - http://www.dailymail.co.uk/sciencetech/article-2174274/No-wonder-hackers-easy-Most-26-different-online-accounts--passwords.html (accessed on 12.10.2015)
[3] F. Khatibloo, "Personal Identity Management," Forrester Research Report, 2011.
[4] 240,000 Norwegians exposed to ID theft - http://www.dagbladet.no/2013/10/06/nyheter/id-tyveri/krim/29631041/
[5] Identitetstyveri, report from Datatilsynet, 2009. - https://www.datatilsynet.no/Global/04_analyser_utredninger/2009/utredning-om-id-tyveri.pdf
[6] Telenor data breach incident - http://www.tu.no/it/2013/03/18/-bedre-opplaring-kunne-trolig-avverget-angrepet-mot-telenor-ledelsen
[7] Ashley Madison data breach - https://en.wikipedia.org/wiki/Ashley_Madison_data_breach
[8] Anthem data breach - https://en.wikipedia.org/wiki/Anthem_medical_data_breach
[9] eID Interoperability for PEGS - http://ec.europa.eu/idabc/en/document/6484.html
[10] Internal Market Information System - http://ec.europa.eu/internal_market/scoreboard/performance_by_governance_tool/internal_market_information_system/index_en.htm
[11] Open ID - http://openid.net/
[12] Estonia citizen card - https://en.wikipedia.org/wiki/Estonian_ID_card
[13] The Best Password Managers for 2015 - http://www.pcmag.com/article2/0,2817,2407168,00.asp
[14] Single sign on - https://en.wikipedia.org/wiki/Single_sign-on
[15] FEIDE - https://www.feide.no
[16] FIDO Alliance - https://fidoalliance.org
[17] PIDaaS project - http://www.pidaas.eu/
[18] Note: **it is not necessary a weaker authentication mode can be configured for seemingly "high-trust" case** such as in a "spear phishing" attack. For a recent instance, in October 2015, several HiG's employee's eID were compromised by a malware disguised as an email attachment sent from their trusted administrative colleagues' email accounts.
[19] HBSI model - http://icbrpurdue.org/the-hbsi-model/
[20] PCAS project - https://www.pcas-project.eu/
[21] SWAN project - http://nislab.no/biometrics_lab/swan
[22] Gartner Says Managing Identities and Access Will Be Critical to the Success of the Internet of Things - http://www.gartner.com/newsroom/id/2985717
[23] Dridex malware aiming at users' bank credentials - http://www.theguardian.com/technology/2015/oct/14/what-is-dridex-how-can-i-stay-safe

Appendix B

Electronic Identity(eID) Questionnaire

IDforU Questionnaire: The Usage of Electronic Identity(eID), Smart cards, and Smartphones

Dear participant,

Thanks for your effort and kindness on this questionnaire about the usage of electornic identity(eID), smart cards, and smartphones.

The project called "IDforU" with the purpose of designing a better eID solution that combines with Biometrics (fingerprint) and JavaCard into one device for multiple usages.

For all questions, please consider events that has occurred in the last 18 months.
All answers are anonymous, and will approximate take you 15 minutes.

Thanks again!
Contacts: qingbao.guo@ntnu.no, shic@stud.ntnu.no

*Required

Part 1: Questions in relation to your profile

What is your age? *

- 0 - 10
- 11 - 20
- 21 - 30
- 31 - 40
- 41 - 50
- 51 - 60
- 61 ++

What is your gender? *

- Male
- Female

Which country are you from? *

Which language do you speak? *

- [ ] English
- [ ] Norwegian
- [ ] Other:

What is the highest level of education you have succeeded? *

- ○ Primary or lower secondary school, no formal education
- ○ Upper secondary school
- ○ High education (university, college, polytechnic)

How would you describe your current professional situation? *

- ○ I'm a student
- ○ I'm working for the public sector: administration, education…
- ○ I'm working in IT industry (software, hardware, services…)
- ○ I'm working for the private sector outside of the IT industry
- ○ I'm presently unemployed
- ○ Other:

Part 2: Use of electronic Identity (eID)

Electronic Identity (from now on abbreviated to eID) are the credentials we use to prove our identity, such as username/password on certain websites, credit card/PIN code for banking systems, etc.

What type of eID and how frequently you are using to prove your electronic identity(ies)? *

	daily	at least once a week	at least once a month	at least once a year	have but (almost) never use them	don't have
Nickname/pseudonym + password	○	○	○	○	○	○
Real name/ID(student/personal ID) + password	○	○	○	○	○	○
Supported by a token (generates temporary random numbers for one-time usage), e.g. BankID	○	○	○	○	○	○
Card + PIN code (employee card, bank card, etc)	○	○	○	○	○	○
Supported by a SIM card + mobile device (SMS message one-time	○	○	○	○	○	○

passcode with manual input), e.g. MinID						
Supported by a SIM card + mobile device (SMS message one-time passcode with visual comparison), e.g. BankID on Mobile	○	○	○	○	○	○
Supported by biometric information (fingerprint, iris scan, etc), e.g TouchID on iPhone	○	○	○	○	○	○
Other	○	○	○	○	○	○

How did you obtain the type of eIDs you are using? *

	from professional activities (work, education, etc)	eBanking client	offical eID in my country issued by the government	member of a social network	to purchase goods or services for eCommerce websites	registe by mysel
Nickname/pseudonym + password	○	○	○	○	○	○
Real name/ID(student/personal ID) + password	○	○	○	○	○	○
Supported by a token (generates temporary random numbers for one-time usage), e.g. BankID	○	○	○	○	○	○
Card + PIN code (employee card, bank card, etc)	○	○	○	○	○	○
Supported by a SIM card + mobile device (SMS message one-time passcode with manual input), e.g. MinID	○	○	○	○	○	○
Supported by a SIM card + mobile device (SMS message one-time passcode with visual comparison), e.g. BankID on Mobile	○	○	○	○	○	○
Supported by biometric information (fingerprint, iris scan, etc), e.g TouchID on iPhone	○	○	○	○	○	○

Do you trust the eID you are using? If you don't fully trust some eID, what are the reasons? *

	Fully	not aware of the	don't believe	don't need	too	don't trust the issuers of	don't trust the issuers of these

	trust	existence of these tools	these are technically secure	these tools	complicate for me	these tools belong to the PUBLIC sector	tools belong to the PRIVATE sector
Nickname/pseudonym + password	○	○	○	○	○	○	○
Real name/ID(student/personal ID) + password	○	○	○	○	○	○	○
Supported by a token (generates temporary random numbers for one-time usage), e.g. BankID	○	○	○	○	○	○	○
Card + PIN code (employee card, bank card, etc)	○	○	○	○	○	○	○
Supported by a SIM card + mobile device (SMS message one-time passcode with manual input), e.g. MinID	○	○	○	○	○	○	○
Supported by a SIM card + mobile device (SMS message one-time passcode with visual comparison), e.g. BankID on Mobile	○	○	○	○	○	○	○
Supported by biometric information (fingerprint, iris scan, etc), e.g TouchID on iPhone	○	○	○	○	○	○	○

How do you think the user experiences of the eID you are using? *

User experience (UX) refers to a person's emotions and attitudes about using a particular product, system or service. -Wikipedia. Including without limitation, UI (user interface), convenience, effectiveness, etc. From 1 to 7 where 1 refer to poorest UX you feeling, while 7 refer to greatest UX.

	poorest 1	2	3	4	5	6	greatest 7	don't have
Nickname/pseudonym + password	○	○	○	○	○	○	○	○
Real name/ID(student/personal ID) + password	○	○	○	○	○	○	○	○
Supported by a token (generates temporary random numbers for one-time usage), e.g. BankID	○	○	○	○	○	○	○	○
Card + PIN code (employee card, bank card, etc)	○	○	○	○	○	○	○	○
Supported by a SIM card + mobile device (SMS message one-time passcode with manual input), e.g. MinID	○	○	○	○	○	○	○	○
Supported by a SIM card + mobile device (SMS								

message one-time passcode with visual comparison), e.g. BankID on Mobile	○	○	○	○	○	○	○	○
Supported by biometric information (fingerprint, iris scan, etc), e.g TouchID on iPhone	○	○	○	○	○	○	○	○

Which eID are you using in Norway? *

- [ ] BankID
- [ ] MinID
- [ ] Buypass
- [ ] Commfides
- [ ] BankID on Mobile
- [ ] Yubikey
- [ ] Student/Employee card
- [ ] Password Manager
- [ ] Account name/ password (for e.g., email and social media)
- [ ] Other:

Which eID are you using most frequently in Norway? *

- [ ] BankID
- [ ] MinID
- [ ] Buypass
- [ ] Commfides
- [ ] BankID on Mobile
- [ ] Yubikey
- [ ] Student/Employee card
- [ ] Password Manager
- [ ] Account name/ password (for e.g., email and social media)
- [ ] Other:

Which eID are you using outside of Norway, if any?

Part 3: Use of smart cards

Smart card is any pocket-sized card that has embedded integrated circuits (IC chips), such as credit cards, access cards, etc.

How many cards do you have (including credit cards, bus cards, access cards, membership cards, etc.)? *

- ○ 0
- ○ 1 - 5

- 6 - 10
- 11 - 20
- 21 ++

How many cards you are using per day/week/month/year? *

	0	1 - 2	3 - 5	6 - 10	11 - 15	16 - 20	21 ++
daily							
at least once a week							
at least once a month							
at least once a year							
have but (almost) never use them							

Do you feel any inconvenience when you do have to use multiple cards in your daily life? *

- Yes
- No
- I don't know

Have you lost any of you cards? *

- Yes, once
- Yes, several times
- No
- I don't know

Have your credentials or properties ever been lost, or in other words, you were aware of the theft cases of your own credentials or properties caused by the loss of your cards? *

- Yes
- No
- I don't know

Have you ever had temporary or/and permanent access blocking caused by the loss of your cards? *

- Yes
- No
- I don't know

If there has a smart card that saves all of your IDs in your daily life, in other words you can use it as your credit card, employee card, driving licence, etc. Are you willing to use it? *

- Yes, Absolutely!
- Yes, but with some worry about card's lost and security/privacy of the card's content, etc.
- No, I am worry about about card's lost and security/privacy of the card's content, etc.

- ◯ I don't know
- ◯ Other:

Do you prefer a separate smart card based ID manager or directly using the SIM card at a phone as a ID manager (if technically possible)? *

- ◯ Separate smart card (I think it more secure)
- ◯ SIM card on a phone (I think it more convenient and less costly than using an extra smart card)
- ◯ I don't know
- ◯ Other:

Part 4: Use of passwords

How many passwords do you have (Including PIN code, any password you are using)? *

- ◯ 0
- ◯ 1 - 5
- ◯ 6 - 10
- ◯ 11 - 20
- ◯ 21 ++

How many passwords you are using per day/week/month/year? *

	0	1 - 2	3 - 5	6 - 10	11 - 15	16 - 20	21 ++
daily	◯	◯	◯	◯	◯	◯	◯
at least once a week	◯	◯	◯	◯	◯	◯	◯
at least once a month	◯	◯	◯	◯	◯	◯	◯
at least once a year	◯	◯	◯	◯	◯	◯	◯
have but (almost) never use them	◯	◯	◯	◯	◯	◯	◯

Do you feel any inconvenience when you do have to use multiple passwords in your daily life? *

- ◯ Yes
- ◯ No
- ◯ I don't know

Have you used the same password for multiple applications or/and websites? *

- ◯ Yes
- ◯ No
- ◯ I don't know

Have you forgot any of you passwords? *

- ◯ Yes, once
- ◯ Yes, several times
- ◯ No
- ◯ I don't know

Have your credentials or properties ever been lost, or in other words, you were aware of the theft cases of your own credentials or properties caused by the loss of your passwords? *

- ◯ Yes
- ◯ No
- ◯ I don't know

Have you ever had temporary or/and permanent access blocking caused by the loss of your passwords? *

- ◯ Yes
- ◯ No
- ◯ I don't know

If there has a smart card that saves all of your passwords in your daily life, are you willing to use it? *

- ◯ Yes, Absolutely!
- ◯ Yes, but with some worry about card's lost and security/privacy of the card's content, etc.
- ◯ No, I am worry about about card's lost and security/privacy of the card's content, etc.
- ◯ I don't know
- ◯ Other:

Part 5: Use of Smartphone & NFC & Fingerprint

Near field communication (NFC) is a set of communication protocols that enable two electronic devices, one of which is usually a portable device such as a smartphone, to establish communication by bringing them within 4 cm of each other.

Which kind of smartphones are you using? *

- ☐ iPhone
- ☐ Android
- ☐ Nokia
- ☐ Windows Phone
- ☐ BlackBerry
- ☐ Other:

Do your smartphones have NFC? *

- ◯ Yes
- ◯ Some of my smartphones have
- ◯ No
- ◯ I don't know

Have you ever used NFC function on your smartphones? *

- ○ Yes
- ○ No
- ○ I don't know

Do your smartphones have fingerprint recognition? *

- ○ Yes
- ○ Some of my smartphones have
- ○ No
- ○ I don't know

Have you ever used fingerprint recognition on your smartphones? *

- ○ Yes
- ○ No
- ○ I don't know

Have you ever lost your smartphones? *

- ○ Yes, once
- ○ Yes, several times
- ○ No
- ○ I don't know
- ○ Other:

Do you think, from your own perspective, fingerprint recognition on a smartphone is a good option compared to other modalities? *

- ○ Yes, it's very convenient
- ○ No, it's less secure and may have privacy concerns
- ○ I don't know
- ○ Other:

Have your credentials or properties ever been lost, or in other words, you were aware of the theft cases of your own credentials or properties caused by the loss of your smartphones? *

- ○ Yes
- ○ No
- ○ I don't know

Have you ever had temporary or/and permanent access blocking caused by the loss of your smartphones? *

- ○ Yes
- ○ No
- ○ I don't know

If there has a smart card that saves all of your IDs in your daily life which is also combining fingerprint recognition instead of using passwords, are you willing to use it? *

- ○ Yes, Absolutely!
- ○ Yes, but with some worry about card's lost and security/privacy of the card's content, etc.
- ○ No, I am worry about about card's lost and security/privacy of the card's content, etc.
- ○ I don't know
- ○ Other:

Please give some advices you may have on the future eID solutions or ideas.

If you like to be informed of the survey results or would like to try our prototype of IDforU when it's ready, you can choose to leave your email address:

Thank you so much!
Project IDforU

Submit

Never submit passwords through Google Forms.