Non-identifying Data Management Systems

Dheeraj Kumar Bansal

Kongens Lyngby 2015

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary

In this thesis we performed an initial analysis of a single business process from administrative data management, with respect to identifying the need to bind authenticated identities to actions at the different steps of the process. Based on this analysis we proposed a new model for the business process. We evaluated our model with different evaluation criteria, set during the initial phase. Based on discussion with the stakeholders, we arrived at the conclusion that even though our proposed system solves a lot of privacy related problems for the stakeholders, in case of business processes, it is not easy to change the legacy systems. We also found out an interesting set of problems that can arise with such systems.

Preface

This thesis was prepared under the guidance of Professor Christian D. Jensen of Department of Applied Mathematics and Computer Science at the Technical University of Denmark and Professor Markus Hidell from the School of Information and Communication Technology at KTH Royal Institute of Technology in partial fulfillment of the requirements for acquiring an M.Sc degree in Security and Mobile Computing.

The work presented in this thesis was supported by Nykredit and Signicat, who provided support in terms of requirements and business domain specific knowledge.

Lyngby, 26-June-2015

Dheeraj

Dheeraj Kumar Bansal

Acknowledgements

I express my sincere gratitude towards my thesis supervisors Prof. Christian Damsgaard Jensen and Prof. Markus Hidell for providing me the opportunity and facilities to do my research, for giving me constant feedback during the thesis and help in time of need.

I would like to thank Prof. Sebastian Mödersheim from DTU compute for sharing his expertise during this project. I would like to thank Nykredit, specifically Lars Lundgaard Andersen for providing the business case and feedback whenever needed. I want to thank Signicat, specifically Lars Møller Kristensen and Jakob Braun for discussions about the solutions and providing helpful feedback.

I would also like to thank my colleagues and friends who helped me during my time here and encouraged me all the time. I would like to thank BEST Copenhagen for teaching me how to have fun while learning and also for the constant encouragement to give my best on everything.

A sincere thanks to my friend Ana Torres for all her support during this past semester.

Last but not the least I am thankful to my parents, who have been supportive of me all these years and have always encouraged me to pursue my dreams.

Contents

List of Figures

Chapter 1

Introduction

This chapter introduce the topic of the thesis. It defines the problem statement and also give a brief background about it.

1.1 Background

Denmark uses a social security number known as *Central Personal ID (CPR Nr.)* to provide digital identity to its citizens. This digital identity is known as *NemID*[1]. Administrative data systems in Denmark currently rely on the NemID to link customer data with a real world identity. This means that almost all data managed by the institutions must be classified as personal identifiable information and therefore managed according to strict confidentiality requirements as well as integrity and availability requirements. This data is still vulnerable to insider threats, such as the recent leak of celebrity data from *NETS* to the magazine *"Se & Hør"*[2]. However it is required by the authorities that the system should be able to link data with the real identity of the person, whenever there is some suspicion of criminal activity, e.g. fraud, insider trading, money laundering, etc.

1.2 Motivation

This thesis has been done in collaboration with 2 companies : *Signicat*[3] and *Nykredit*[4].

Signicat is a provider of digital identity and digital signature solutions. They have the highest coverage of national electronic identities in Europe. They want to be able to offer services to financial institutions like Nykredit to help them in achieving the privacy goals regarding their customers.

Nykredit is a major financial institution in Denmark, providing different services, such as mortgages, retail banking, investment banking etc. They also are part of a big group of companies, which includes other financial institutions providing similar services. Currently there are 61 regional banks and partner institutions which have this partnership with Nykredit. These financial institutions basically provide Nykredit services as their own services to the customers. Nykredit has mainly 2 types of customers:

1. Private Customers

2. Corporate Customers

1.2.1 Private Customers

Private customers are the individual customers, who have personal bank account with Nykredit and access their services themselves. Usually there is a single person accessing the services of the bank.

1.2.2 Corporate Customers

Corporate customers are either companies who are customers of Nykredit or other financial institutions which provide Nykredit services to their own private customers. In this case, there are many people who access Nykredit services on behalf of the corporate customer.

1.3 Problem

We consider the case of a person, who may be a private customer of Nykredit, and an employee of a company who is a corporate customer of the bank. For corporate customers, they have some employees who manage their bank account. In this case, the person may also be responsible for managing the accounts of his employer with Nykredit. This entity relationship is shown in the figure 1.1.

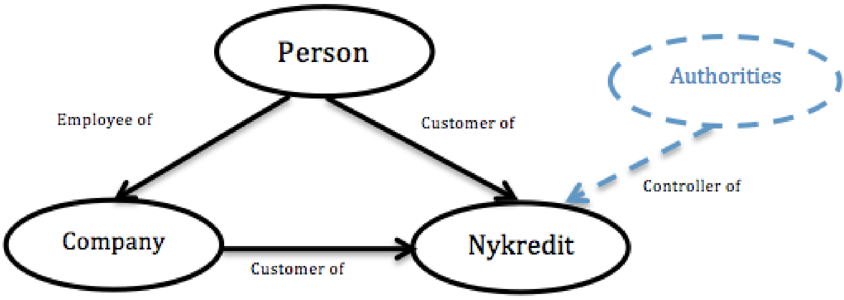


Figure 1.1: Identities in the system

Nykredit wants to setup an identity management system so that there is no need for the individual to disclose his personal identity to Nykredit to access the account on behalf of the company.

Nykredit, however, also has to comply with relevant legislation (KYC, AML, Hvidvaskningsloven etc.), e.g. in case the authorities (Tax, Police, etc.) find some suspicious transactions. They need to provide the identity of the person responsible for these transactions. This means that it is required that Nykredit, in case of a legal request, is able to identify the individual employee from the institution, who is accessing the account on the corporate customer's behalf.

So the main goal of the system is:

- Nykredit should not learn identity of the individual person accessing the services on behalf of corporate customer.
- To comply with legislation, Nykredit should be able to map the real identity of the individual person with the transaction, in case it is required by the law.

This project will perform an initial analysis of a single business process from administrative data management, with respect to identifying the need to bind authenticated identities to actions at the different steps of the process; this analysis will be presented to stakeholders from the specific administrative domain. Based on the initial analysis of the selected business process, the project will develop a full identity model for the chosen business process with anonymization and pseudonymization of actors whenever possible. The feasibility of the proposed model will be evaluated through a prototype that implements the model using standard components from identity management infrastructures whenever possible.

Companies do not want to disclose the personal identity of their employees to Nykredit, but they still need the ability to access all services online. Managing all identities, while maintaining privacy, is not easy and provides different challenges. We have to design a system, which fulfill the entire privacy requirement and still enables Nykredit to provide its services to its customers and meet the regulatory requirements of the authorities without any problem.

Chapter 2

State of the art Survey

This chapter introduces some of the technologies currently available that deal with anonymity and privacy. Some of the technologies are already commercially available and being used in the industry, while others are still in research phase. We will give a brief introduction for all of them and a brief idea of how they can be used.

2.1 Definitions and Common Terminology

Below are the main terminologies and definitions used in this thesis. Some of the terms are influenced by [5]:

- **Privacy :** It is the ability of an individual to control the distribution of information about himself. An individual should be able to choose which information about him should remain secret and which information can be revealed.
- **Anonymity[6] :** It refers to the ability of a user to not give any information about him at all to the system. It is the state of not being identifiable in the system. An anonymous system doesn't have any identity of the user.

- **Pseudonym[7] :** It is a name given to the user in pseudonymous systems. This name is given to hide the real identity of the user from the system. The system only knows the user by his pseudonym.
- **User :** It is the end user of the system. It is the person who will go online and get the services. In our system, most of the time, we refer to user as the employee of the company, who is accessing the services of the bank.
- **Bank :** It is the financial institution which provides online financial services to the user. In our system, we refer to Nykredit as the bank.
- **Third party :** A third party or trusted third party is the entity which is neither the bank or the user in the system. A third party provides different services to banks or users and hence reduces the burden on them to setup all the infrastructure by themselves. In our system, Signicat is referred to as the third party.
- **Service :** It is something that is provided to the user online by a system. It may include the ability to login, check his account balance, upload pictures, share spreadhseets etc.
- **Unlinkability :** It is the privacy property where it is not possible to link 2 different entities to each other even though they are the same. e.g. not to be able to link 2 different sessions by the same user in the system.
- **Revocation :** It is the property where a user credential is revoked by the user or some other authority. After revocation, this credential cannot be used for anything.
- **Partial Information Disclosure :** It is the ability of a user to only disclose some partial information about himself to the system. e.g. a user might just want to disclose his last name to the system but not his full name.
- **Legal Requirement :** It is something that is required by law. e.g. it may be required by law that the bank logs all the customer data. Also sometimes in case of suspicious transactions the bank may be required legally to give the user identity to the relevant authorities.
- **Conditional Anonymity Removal :** It is the ability of the system to remove anonymity of the user if some conditions that were set before are met. This is mainly used for escrow purposes.
- **Verifiable Encrytion[8] :** It is a type of encryption in which encrypted data can be verified i.e. someone who doesn't know the actual data can verify that the encrypted data is the same as claimed by the person who encrypted it. e.g. if the person who is encrypting the data has to encrypt

his key in it, it can be verified that he has encrypted his real key and not some garbage value.

- **Zero Knowledge Proof[9][10] :** It is a type of proof where the prover is able to convince the verifier that a statement is valid without giving any other information about the statement except that it is valid. e.g. if a user has to prove that he has the private key to a particular public key, he can prove it without giving away any other information about his private key.

2.2 Technologies

After getting the requirements and based on our terminology we looked at the current technologies. We came out with a map as in figure 2.1. Basically we divided our type of technologies in four different types, depending on the functions:

1. Secure Multi-Party Computing
2. Escrow Technologies
3. Identity Management Systems
4. Zero Knowledge Technologies

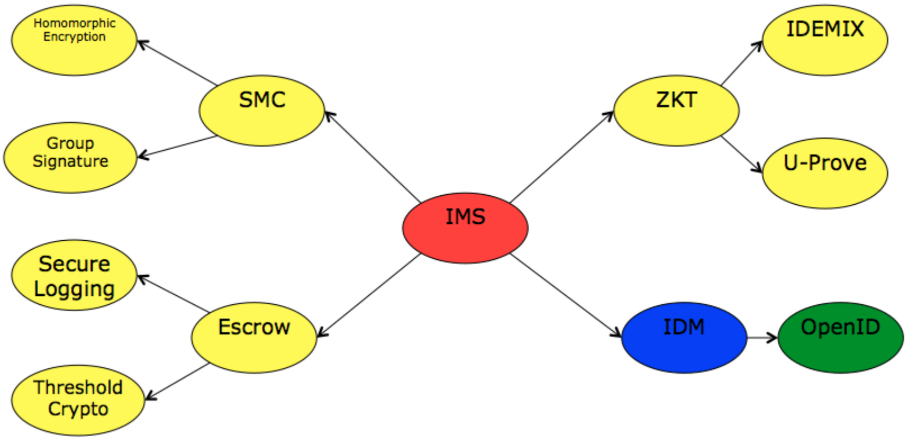


Figure 2.1: Technology Overview

2.3 Secure Multi-Party Computing

These technologies are the ones which involve multiple parties to do computations[11]. It is a subfield of cryptography which involves multiple parties getting an input and compute a joint function on them while keeping these inputs private. We basically looked at 2 technologies of interest here:

2.3.1 Homomorphic Encryption

Homomorphic encryption[12] is a type of encryption where certain arithmetic operations can be performed on the *ciphertext* so that when the resultant ciphertext is decrypted, the decrypted text is the same as if the operations were performed on the [?]. This is a new field of cryptography and is very useful where we need some parties to perform such operations without revealing the underlying data to those parties. Homomorphic encryption is also useful for the chaining of different services without exposing data to any of those services.

2.3.2 Group Signature

A *group signature*[13] is a scheme which allows a member of the group to sign the message on behalf of the group but anonymously. To outsiders the message has been signed by someone from the group but the exact identity of the person is now known. Also if the same member signs 2 different messages; it is not possible to know if the message is signed by the same member or 2 different members. There is a notion of *group manager* in these scheme. Group manager is someone who manages the membership to the group. He can add/remove members from the group, find out who actually signed the message from the group. This scheme is useful where the only thing that needs to be validated is that a certain person is part of the group, but his real identity is not required.

2.4 Escrow Technologies

Escrow technologies are those which are helpful in escrow purposes i.e. getting real data/identity later on in time from encrypted data if needed. We look at the 2 following technologies:

2.4.1 Secure Logging

Secure logging[14] is the process of saving the data in a secure manner, as saved data is really crucial and vulnerable to attacks. We need to make sure that the data is saved securely and its integrity is protected. This can be done in several ways. One way is to encrypt all the logs while storing them so that even if someone gets hold of the logs, they can't use them without having access to the decryption key. Another way is to store logs at a third party after encrypting them. For escrow purposes these logs can be decrypted later on with the decryption key.

2.4.2 Threshold Cryptography

Threshold cryptography[15] is a field of public key cryptography where in order to decrypt an encrypted message, several parties must cooperate in the decryption. This message is encrypted using a *public key* and the corresponding *private key* is shared among different parties who will participate in the decryption process i.e. multiple parties hold the private key for a single public key. There is a term known as *threshold*, and if there are n parties who share the private key and at least t parties which are required to decrypt the message such system is called *(t,n)* threshold cryptosystem. Threshold cryptosystem is useful in escrow purposes where a minimum number of parties are required to decrypt the ciphertext in order to get the plaintext.

2.5 Identity Management Systems

These are traditional identity management systems. For our purposes we look at the *OpenID*[16] system.

2.5.1 OpenID

OpenID is an open and decentralized protocol, which can be used to authenticate with different co-operating sites with the use of a third party service. It has the notion of a *relying party* and *OpenID identity provider*.

- **OpenID Identity Provider :** It is the service, which actually provides authentication services. End user registers at OpenID identity provider to get his OpenID identity.
- **Relying Party :** It is the website which user wants to authenticate to and which rely on the OpenID identity provider to provide authentication.

In addition to this an extension called *OpenID attribute exchange*[17] helps facilitate the transfer of user attributes from the identity provider to the relying party.

- The user goes to the relying party service page.
- The service page presents different OpenID providers to login to the service.
- The user chooses the provider with whom he has registered his OpenID.
- The relying party redirects the user to the OpenID provider url so that the user can authenticate.
- The user can be authenticated by the method provided by the OpenID provider.

- The OpenID provider asks the user to get his permission to share the attributes with the relying party.
- After the user gives his consent, he is redirected to the relying party website with the user credentials.
- The relying party can verify the credentials and then login the user to the service.

2.6 Zero Knowledge Technologies

These are the technologies which use the concept of zero knowledge[9] i.e. proving knowledge about something without divulging the information. For our purpose we focus on 2 main technologies – *IDEMIX*[18] and *U-Prove*[19], which are based on the concept of zero knowledge[9] and verifiable encryption[8]. They both have a lot in common and have been studied a lot.

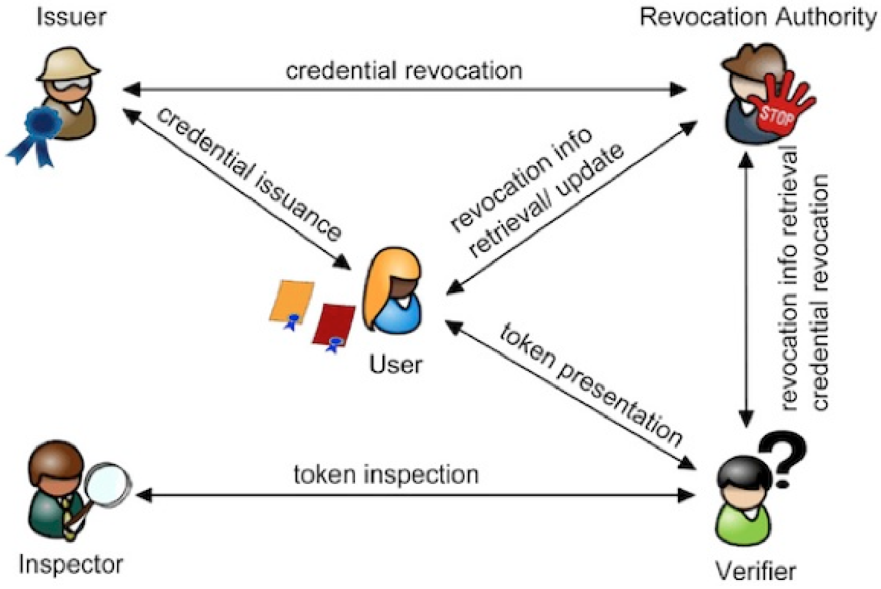
2.6.1 IDEMIX

IDEMIX[20][21]is a digital credential technology by IBM. It relies on anonymous credentials known as IDEMIX tokens. It is based on Camenisch-Lysyanskaya (CL) signature scheme[22] which provides efficient zero-knowledge proofs. IDEMIX has different entities in the system:

- **User :** It is basically the entity who is proving his identity in the system.
- **Verifier :** It is the entity that verifies the identity of the prover.
- **Issuer :** It is the entity that issues the credentials to the prover to prove his identity
- **Inspector :** It is the entity which, in case of discrepancy or legal requirement, can actually come and get the real identity of the prover.

Figure 2.2 shows these roles in detail.

For IDEMIX we need *computing devices* that work on behalf of each entity in the system.



Source: https://github.com/p2abcengine/p2abcengine/wiki/Concepts-and-features

Figure 2.2: IDEMIX Roles

2.6.1.1 IDEMIX Credential

An IDEMIX credential is a *CL Signature*[22] by issuer on the user's private key and on the attribute values. A user has independent public keys or pseudonyms for the same private key. These pseudonyms are IDEMIX tokens which are then used by the user to prove his identity to the different verifiers. IDEMIX has been studied a lot and many EU projects on anonymous credentials are based on it e.g. FutureID[23], ABC4Trust[24] etc. An example of an IDEMIX credentials is shown in figure 2.3.

2.6.1.2 Issuance

The first step is the credential issuance. It involves the following steps:

- The user sends a credential request to the issuer.
- The issuer presents the *issuance policy* specifying:

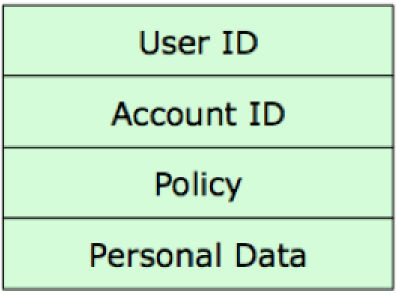


Figure 2.3: IDEMIX Credential

  - What attributes to present.
  - Which pseudonym/existing credentials to present.

- The issuer also present a *credential template* specifying:
  - Which attributes of the new credentials will be generated at random.
  - and which attributes will be carried over from existing credential or pseudonym.
- The user then presents the issuance token satisfying the issuance policy.
- Then multi-round cryptographic protocol ensues, at end of which, the user gets the IDEMIX credential.

2.6.1.3 Presentation

The next step is to present the IDEMIX token for authentication to the verifier. It consists of the following steps:

- The user gets the presentation policy from the verifier which specifies:
  - Which credentials the user must present.
  - What information the user should reveal from these credentials.
- The user generates a presentation token in accordance with the presentation policy revealing only the necessary attributes.
- The user presents this presentation token to the verifier.
- The verifier can then verify the attributes from the presentation token presented by the user.

2.6.1.4 ID Escrow

IDEMIX provides the ID escrow ability in case it is required. The steps below need to be followed for ID escrow purposes in IDEMIX:

- The presentation policy can have the following optional specifications for the purpose of ID Escrow:
  - Public keys of the inspectors.
  - Attribute values to be encrypted using the keys.
  - Inspection conditions under which these attributes can be revealed.
- The user can prove that he has put these values in the presentation token with verifiable encryption[25] to the verifier.
- Once the token in presented, the inspection conditions are fixed and cannot be changed.
- In case of some discrepancy or legal requirement, an inspector can come and get the identity of the user from the token.

  A visual representation of creating an IDEMIX presentation token from the IDEMIX credential can be seen in figure 2.4.

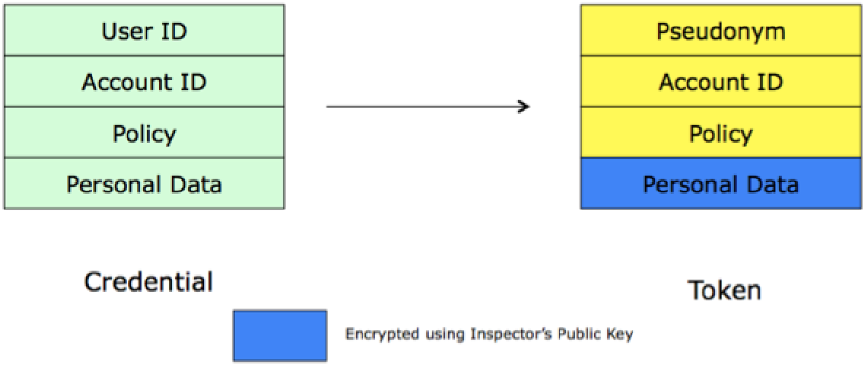


Figure 2.4: IDEMIX Presentation Token

2.6.2 U-Prove

U-Prove is a digital credential technology by Microsoft[26][27]. It relies on anonymous credentials known as U-Prove tokens. It provides users a way to minimaly disclose their personal information while interacting with different online services. U-prove have different entities in the system

- **Prover :** It is basically the entity, which is proving his identity in the system.
- **Verifier :** It is the entity, which verifies the identity of the prover.
- **Issuer :** It is the entity, which issues the credentials to the prover to prove his identity
- **Auditor :** It is the entity, which in case of discrepancy or legal requirement , can actually come and get the real identity of the prover.

For U-Prove we need computing devices which work on behalf of each entity in the system.

2.6.2.1 U-Prove Token

A U-Prove token is basically cryptographically protected information of any kind e.g. attributes. These are issued by an issuer to the prover by *issuance protocol*. These tokens are then presented by the prover to the verifier. Issuance and presentation of U-Prove tokens is unlinkable.

2.6.2.2 Issuance

The first step is the credential issuance. It involves the following steps:

- The prover invokes U-Prove issuance protocol.
- The prover provides the attributes to be encoded.
  - Using the *Collaborative Issuance*[28] property user can make sure that the issuer doesn't actually know the attributes.
- Then multi-round cryptographic protocol ensues at end of which the user get the U-Prove token from the issuer.

2.6.2.3 Presentation

The next step is to present the U-Prove token for authentication to the verifier. It consists of the following steps:

- The prover invokes the U-Prove *presentation protocol.*
- The user generates a presentation token in accordance with the presentation policy revealing only the necessary attributes.
- The user presents this presentation token to the verifier.
- The verifier can then verify the attributes from the token presented by the user.

It must be noted that a revocation check can be added if needed before verifying the token.

2.6.2.4 ID Escrow

ID Escrow in U-Prove is actually an extension[29] to existing U-Prove technology. It uses a type of ElGamal encryption which is verifiable.

- During the presentation protocol, the prover proves that his ID is encrypted in the token by the use of verifiable encryption[8] technology.
- De-anonymization cannot be done by the verifier or the issuer.
- A special entity called Auditor is responsible for de-anonymization in case of some discrepancy or legal requirement
- Threshold cryptography[15] can be used and the decryption key can be split among multiple auditors.

All these different technologies provide different *levels of anonymization*[5] in the system. Some of them are easy to integrate in existing technology, while others are still not mature enough. From now on we focus mainly on two technologies:

- OpenID
- IDEMIX

Chapter 3

Application Scenario

This chapter will discuss the *information flow* of the current system. We will present our understating of the current banking system. We will also give different types of data that exist in the current system and the different operations that are necessary to support the system.

3.1 Information Flow

As show in figure 3.1 a person has different information associated with him.

To Nykredit it can be:

- Personal ID

  This Personal ID can be one of the following identifiers:

  - External ID (e.g. NemID)
  - Internal ID (e.g. login credentials of the bank)
- Account Data

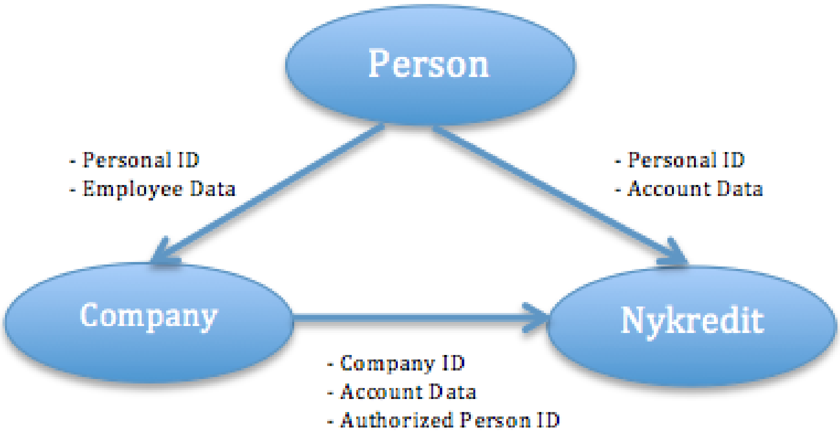


Figure 3.1: Example of information maintained for each relationship

To a company it can be:

- Personal ID
- Employee Data

A company has the following information that it might share with Nykredit:

- Company ID
- Account Data
- Authorized Person ID

This *Authorized Person ID* is the identifier that is used by the authorized person on behalf of the company. This can be stored in some database where it is matched to the Personal ID of the person.

The Authorized Person ID can either be the same identifier as the Personal ID, the Company ID or a different identifier that can be authenticated.

In case the Personal ID is used as the Authorized Person ID, it gives Nykredit some additional capabilities:

- Nykredit can use this information to recruit new customers. If the authorized person is not a customer before, Nykredit can use this info to contact them.
- If the person is already a customer, then Nykredit can use this information to provide additional services to him on his *personal account*, so as to influence the authorized person for his decisions regarding the company's account (similar to the way airlines reward frequent flyers).
- As Nykredit already knows about company accounts, the performance of the company might influence their decision regarding the private account of the employee (e.g. it may be difficult for a person to take out a mortgage if Nykredit knows that the company they work for is in financial difficulties).
- In the case where the customer is accessing Nykredit services on behalf of a smaller financial institution, the capability of matching the authorized ID to the Personal ID gives Nykredit a chance to recruit this customer away from the smaller financial institutions.

While good corporate governance at Nykredit would prevent these issues, it is desired to completely and demonstratively remove the link between the Personal ID and the Authorized Person ID. This, however, creates difficulties with respect to regulatory requirements for accountability at Nykredit and KYC, AML, "Hvidvaskningsloven, etc., so there must be some way to map the identifier used in a financial transaction to the real person, i.e. map a given Authorized Person ID to a Personal ID.

3.2 Separation of Identities

A way to remove the link between the Personal ID and the Authorized Person ID is to use separate IDs and to maintain a database which links the Authorized Person ID back to the Personal ID. This database can be protected in different ways, so that the information can be used only in case legal authorities need to link the two IDs.

This database can be maintained at 3 places :

- Company
- Nykredit
- Trusted 3rd party

3.2.0.5 Database on the Company's Side

Advantages

- Nykredit doesn't have to invest extra in IT infrastructure.

  It is expensive to maintain the entire IT infrastructure by Nykredit, so it is easier and cheaper for Nykredit to let the company maintain the database.

- Nykredit can easily prove that it cannot link different Identities.

  Nykredit does not have access to the database, so it can easily be proved that Nykredit cannot link the different identities.

- The company maintains their own private data.

  Companies can be sure that Nykredit does not have access to the personal data of their employees

Disadvantages

- There is no way to retrieve data if the company stops existing.

  In this case the entire mapping database may be lost.

- Authorities have to go to the company to get the data.

  Nykredit does not have access to the database, so the authorities have to go to Nykredit first to obtain the Authorized Person ID and then to the individual companies next to get the Personal ID.

- The company might tamper with the database.

  In the case of a rogue employee at the company, which is exactly the case that the legislation is intended to identify, this employee will have easy access to this database and hence the ability to tamper with the database and remove the authorities ability to identify him.

- It might prove too difficult for new customers to fulfill all the technical requirements

  Larger companies can have their own IT infrastructure, but for smaller companies it might prove to be a difficult task to become a new Nykredit customer if they have to invest extra in IT infrastructure just for this purpose.

3.2.0.6 Database on Nykredit's Side

Advantages

- In case the company stops existing, the data can still be retrieved.

  The database is always with Nykredit, so if some company stops existing, it can still be accessed.

- Authorities have a single place to obtain all the data.

  Authorities do not have to go to individual companies to get the relevant data as everything is at one place.

- The company cannot tamper with the database.

  As companies have no direct access to the database, they cannot tamper with it.

Disadvantages

- Nykredit has to invest extra in IT infrastructure.

- The company does not have control over their own private data.

  The database is on Nykredit side, so companies have to store the data there and hence they do not have control over their own private data.

- It is difficult for Nykredit to prove that they cannot link different identities when they are managing the database.

  Nykredit will be managing everything in-house, so it is difficult to prove that they cannot access the database and link the identities.

- It may be difficult for customers to adhere to the Nykredit technological standards.

  Nykredit may not be able to support all available technologies for their customers, so some customers, who are using a different setup than Nykredit, may find it difficult to comply with the Nykredit standard.

3.2.0.7 Database on 3rd Party's Side

Advantages

- Neither companies nor Nykredit have to invest extra in IT infrastructure.

  The database is managed by the trusted 3rd party, who will invest in the infrastructure, so neither Nykredit nor the companies will have to invest extra in IT infrastructure.

- It is easier for new and old customers to be a customer at Nykredit.

  The trusted 3rd party can support a wide range of technologies, so it is easier for customers to use their existing technology when becoming a new customer at Nykredit.

- Nykredit can easily prove that it cannot link different Identities.

  Nykredit is not hosting the database, so it is easier for them to prove that they cannot link the identities.

- Data can still be retrieved in case the company stops existing

  The database is always with the trusted 3rd party, so it does not matter if some company stops existing, the data can still be accessed. Special arrangements have to be made in case the trusted 3rd party ceases to exist, but this will be rare and in that case, Nykredit may decide to take over that part of the trusted 3rd party.

- The company cannot tamper with database.

  The companies do not have access to the database, so they cannot tamper with the data.

- The authorities only have to go to the trusted 3rd party to get the data in case its needed.

Disadvantages

- The 3rd party must be trusted by both Nykredit and its customers.

  The database is neither with the company nor Nykredit, so the external service provider should be trusted by both parties to hold their sensitive data.

- In case the trusted 3rd party goes out of business it might be difficult to retrieve the data.

- Companies do not have control over their own data.

  The database is maintained by an external service provider, so the companies have to store the data there and hence they do not have control over their own private data.

3.3 Current Banking System

The current banking system according to us is as shown in figure 3.2. There are 2 parts of the system:

- Authentication Service
- Bank

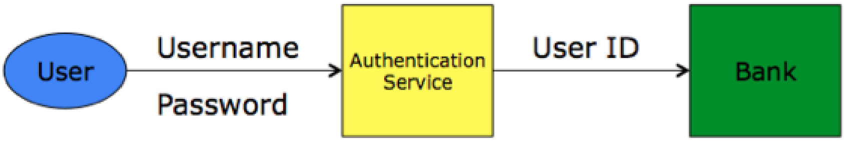


Figure 2: Current Banking System

Figure 3.2: Current Banking System

The end user interacts with the service as follows:

- The user goes to the authentication service and enters his credentials.
- The authentication service authenticates the user and gives the bank the user ID of the user.
- All other details of the user are stored at the bank's side in relation with his user ID.
- The bank provides the services to the user .

The authentication service can either be controlled by the bank or a 3rd party (e.g. NemID)

In this system the bank has all the mappings:

- User ID → Account ID
- User ID → Policy
- User ID → User Personal Data

- User ID → Transactions

This makes the bank the most powerful entity in the system. It is very easy for the bank to get any private data of the customers using the data stored on the bank's side.

In a nutshell, the current banking system gives the bank a lot of power over the user's data. Also, we have seen that traditional methods are not sufficient to provide proper anonymity to the users, or they don't fit all the requirements properly.

Chapter 4

Proposed System

This chapter will present our proposed system as the solution. We will define our system and show how it solves the problem of maintaining privacy for the users. Then we will discuss about a generic prototype we made using the design from the proposed system. We will describe different parts of the system and how the end user perceives it.

4.1 User Identification

In order to solve the problem we first look at how the users are identified in the system. There are 2 ways by which a bank can identify the actual users

- From the logs i.e. transaction data

  The bank store all the logs or transaction data with real identity of the user. It is easy to access this data by the bank for a given user and extract all of his data.

- From a database in the bank system where personal details of the user are stored.

> The bank stores personal data about all its users in a database. As this database lies at the bank, it is possible for the bank to use the database to get the personal information about a user.

We can remove this identification in the following ways:

- Remove the user identity from the logs and transaction data

  Like this there is no way for the bank to get transaction data for a given user as there will be no user identity linked to the logs or transactions.

- Either remove the user's personal data or limit the access to this personal database.

  If the bank removes the personal database of the users from its side, then there is no way the bank can get personal information of the users.

  If the bank limits the access to such database and doesn't really use it for day to day banking purposes then it is also possible for the bank to limit access to the user's personal information.

4.2 Pseudonym System

In order to provide privacy to the users, we suggest a new pseudonym system as shown in figure 4.1. In our system we modify the authentication service and replace it with an *Identity Mapping System (IMS)*

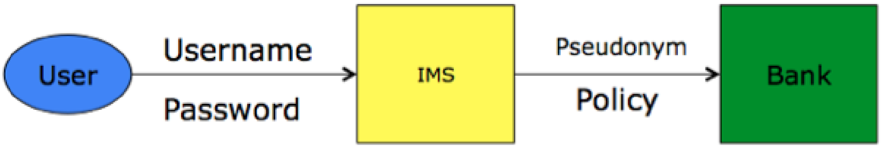


Figure 4.1: Pseudonym Banking System

- Instead of giving the *User ID* to the bank we actually replace it with a *pseudonym.*
- The IMS sends the *policy* as well as the *account ID* to the bank.
- The bank then uses this information to provide services to the customer.

- The bank doesn't need to store the mapping databases.

The IMS can either be in the bank, in a separate department,or it can be managed by a 3rd party. This way the bank doesn't need to know the exact identity of the user to provide them with the services. Also, the bank can still store personal details of the user in case of the legal requirement, but it can be stored at a separate place as it is not needed for day to day operation of the bank.

4.3 Properties

The privacy properties mentioned below are desirable in our pseudonym system:

- **Unlinkability :** If the same pseudonym is used with different identities, or different pseudonyms are used for the same identity, it should not be possible to link these different transactions to the same person.

- **Partial Information Disclosure :** The information given by a user should be minimum and he should be able to choose what information values he actually wants to be made available to the bank.

- **Conditional Anonymity Removal :** In case of some discrepancy or legal requirement, the authorities should be able to come in and identify the real user from the Pseudonym.

- **Revocation :** It should be easy to revoke any user. Also, it should be easy to check whether a certain user is revoked or not.

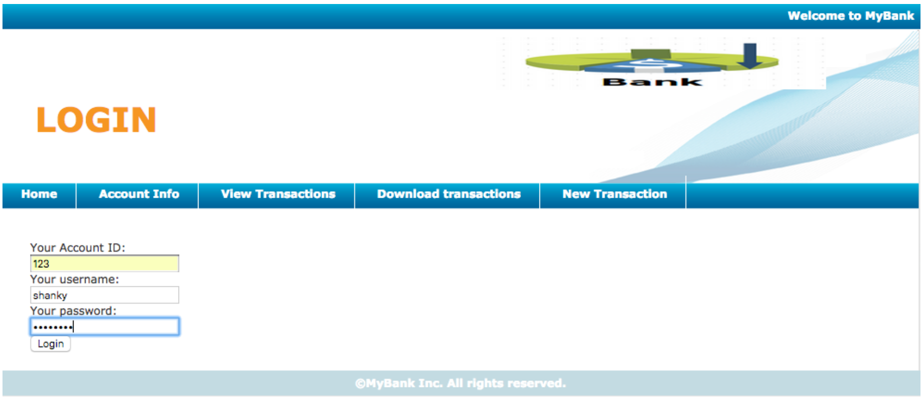
4.4 User Privacy

As in our system the bank never gets the real identity of the user, the user's anonymity to the bank is maintained. Also, the bank doesn't need to store the policies for the users as its all coming from the IMS. As a result, it decreases a lot of load from the bank to store such data. The IMS service adds a layer of pseudonymity in the system.

4.5 Generic Prototype

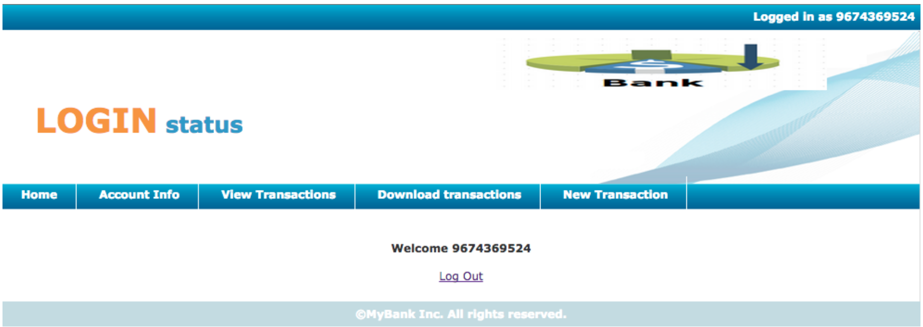
We will now talk about a *generic prototype* based on our proposed system. For the end user it doesn't change anything. The end user authenticates to the IMS and then the IMS creates a pseudonym for the user. This pseudonym is then used by the bank to provide services to the end user.

4.5.1 Authentication



Your Account ID: 123
Your username: shanky
Your password: ••••••••

Figure 4.2: User Login Page



Welcome 9674369524

Figure 4.3: Authenticated User

User authentication happens as follows:

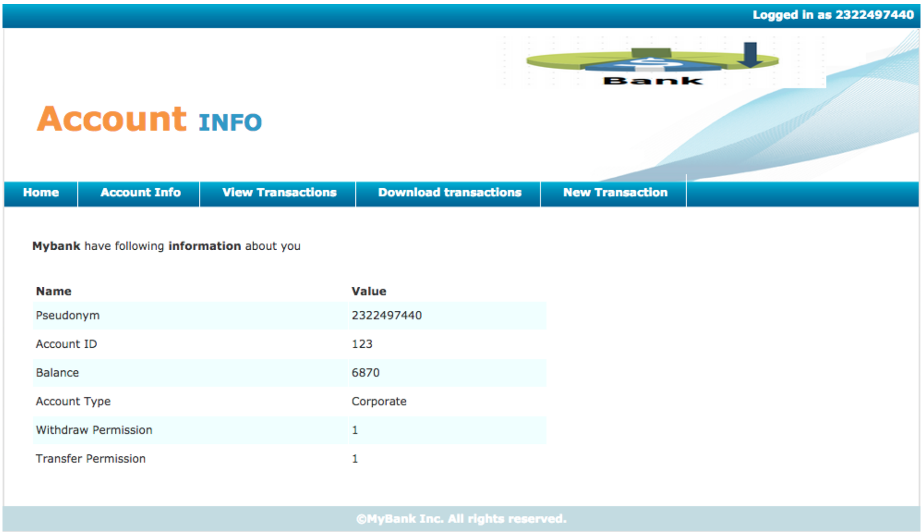
- The user goes to the bank login page.

- The user puts his credentials in the login system.
- The user's credentials are verified by the IMS and then he is redirected to his account page.

One thing to note here is that for a traditional user, this doesn't change anything on the user's end. The user still uses the same process to get access to his account.

As we can see in 4.3, after authentication, the user is logged in with a pseudonym.

4.5.2 User Information



Mybank have following **information** about you

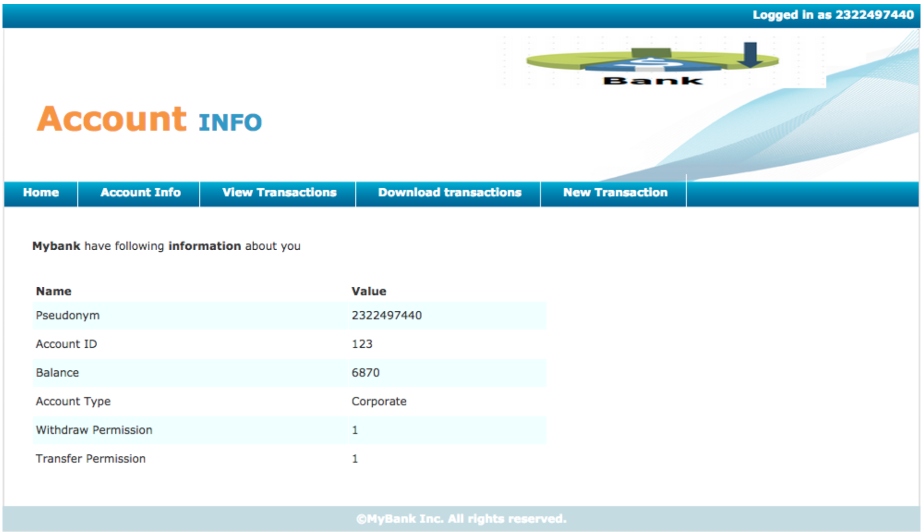
Name	Value
Pseudonym	2322497440
Account ID	123
Balance	6870
Account Type	Corporate
Withdraw Permission	1
Transfer Permission	1

Figure 4.4: User Information - Session 1

When we go to the user information page we can see the information that is given to the bank by the IMS for a given user.

In our system it is:

- Pseudonym
- Account ID
- Balance



Mybank have following **information** about you

Name	Value
Pseudonym	2322497440
Account ID	123
Balance	6870
Account Type	Corporate
Withdraw Permission	1
Transfer Permission	1

Figure 4.5: User Information - Session 2

- Account Type
- Policies
  - Withdraw Permission
  - Transfer Permission

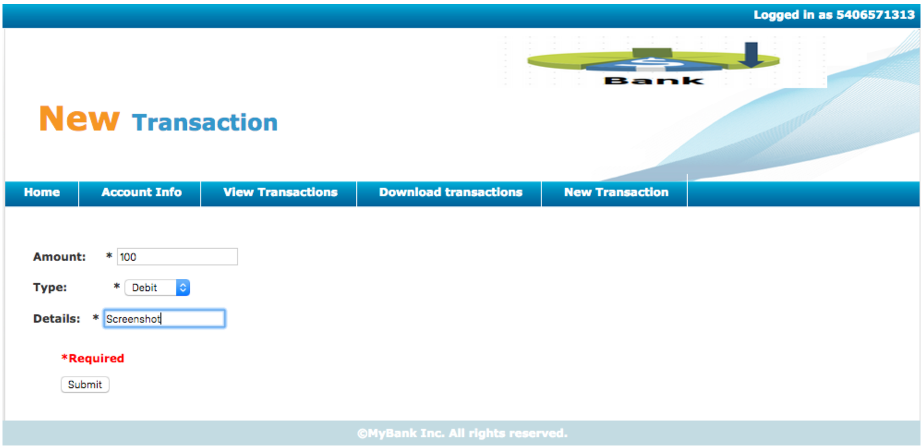
As we can see from the figure 4.4 and the figure 4.5, two different sessions of the same user are logged in with different pseudonyms. The bank has no way to find out that it is the same user who has logged into different sessions.

4.5.3 Transactions

In our prototype the user is allowed to do two types of transactions:

- Debit
- Credit

All the transactions that are done by the user are saved with the pseudonym. This pseudonym is the same with which the user has been logged in to the system.



Amount: * 100

Type: * Debit

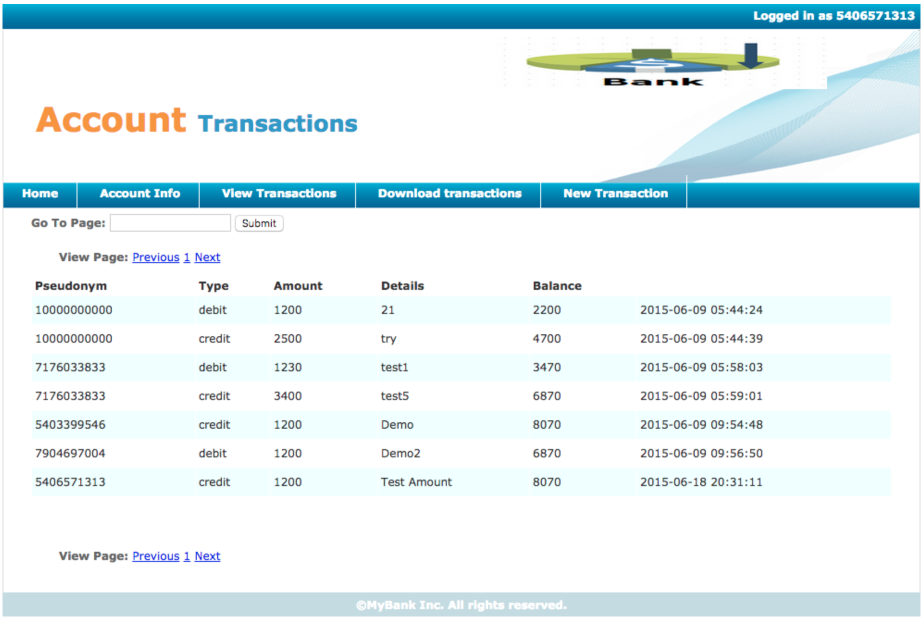
Details: * Screenshot

*Required

Figure 4.6: New Transactions

Figure 4.6 shows the *new transactions* page in the system where the user is allowed to do the transactions.

Figure 4.7 shows all the transactions that have been done on the given account by the users. As we can see, all the transactions are saved with the pseudonyms of the users. Our system also allows the users to download the transactions from the *download transactions* page as shown in figure 4.8. These transactions are stored in a csv file. This csv file can be opened by the user as can be seen in the figure 4.9.



Pseudonym	Type	Amount	Details	Balance	
10000000000	debit	1200	21	2200	2015-06-09 05:44:24
10000000000	credit	2500	try	4700	2015-06-09 05:44:39
7176033833	debit	1230	test1	3470	2015-06-09 05:58:03
7176033833	credit	3400	test5	6870	2015-06-09 05:59:01
5403399546	credit	1200	Demo	8070	2015-06-09 09:54:48
7904697004	debit	1200	Demo2	6870	2015-06-09 09:56:50
5406571313	credit	1200	Test Amount	8070	2015-06-18 20:31:11

Figure 4.7: Account Transactions

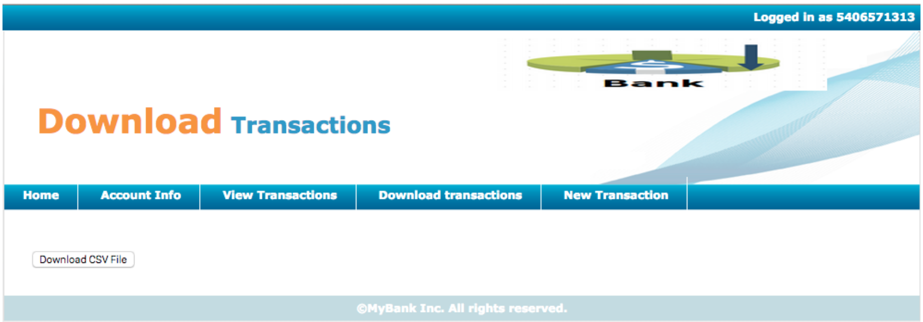
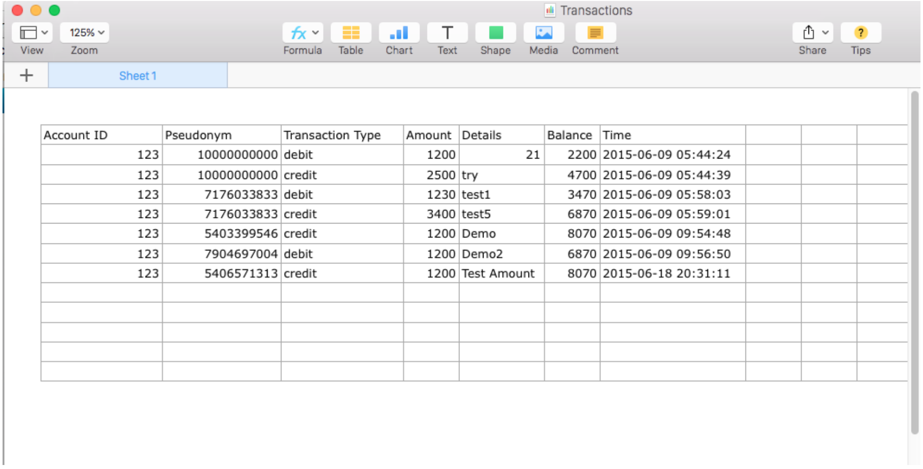


Figure 4.8: Download Transactions Option



Account ID	Pseudonym	Transaction Type	Amount	Details	Balance	Time
123	10000000000	debit	1200	21	2200	2015-06-09 05:44:24
123	10000000000	credit	2500	try	4700	2015-06-09 05:44:39
123	7176033833	debit	1230	test1	3470	2015-06-09 05:58:03
123	7176033833	credit	3400	test5	6870	2015-06-09 05:59:01
123	5403399546	credit	1200	Demo	8070	2015-06-09 09:54:48
123	7904697004	debit	1200	Demo2	6870	2015-06-09 09:56:50
123	5406571313	credit	1200	Test Amount	8070	2015-06-18 20:31:11

Figure 4.9: Downloaded Transactions File

In this chapter we presented our pseudonym system. Also we have given some certain privacy properties that our pseudonym system should be able to fulfill. Our generic system works with pseudonyms and in the next chapter we will talk about how we can replace our Identity Mapping System with OpenID and IDEMIX.

We have decided to analyze two different systems for our IMS:

- OpenID
- IDEMIX

.

Chapter 5

OpenID Based Solution

In this chapter we will provide a solution using OpenID based IMS. We will give more details about how this system can be implemented, and how it will behave for the end users.

5.1 OpenID IMS

We can replace IMS with *OpenID based IMS* in our pseudonymous system as illustrated in figure 5.1. This system will take the user's credentials and then send the pseudonym, account ID and policy to the bank. This IMS can be controlled by a separate identity inside the bank or by a 3rd party.

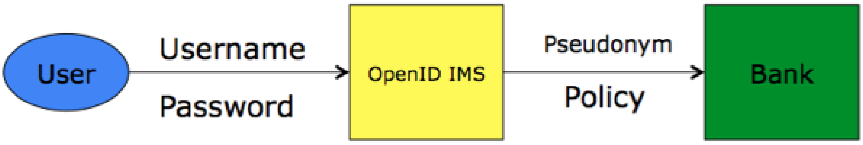


Figure 5.1: Pseudonym System with OpenID IMS

5.2 System Setup

The system can be setup in two ways:

- IMS controlled by a separate department at the bank.

  In this case the bank separates the authentication and the service part in two different departments internally. Authentication is controlled by IMS which holds the sensitive user data but the service department doesn't need to have access to that data to provide services to the user.

- IMS controlled by a third party.

  In this case the bank operates the service part while the authentication part is operated by a trusted third party.

In both cases, the bank and the IMS have to collaborate and the bank has to trust the IMS system that the pseudonym and the policy sent by the IMS system are correct.

5.2.1 Changes on the Bank Side

In order to provide services using a pseudonym, the bank needs to have a temporary policy database at its end. So that when the bank gets the pseudonym and the policy from the IMS system, it can store that policy with the pseudonym and provide the services according to the policy.

5.2.2 Information Stored at IMS

IMS needs the following user information to be stored:

- User ID
- Account ID
- Policy

The account ID and the policy can be stored in an encrypted form, which can then only be opened by the bank. OpenID IMS also needs to store a *mapping database* from the user ID to the pseudonym for escrow purposes.

5.2.3 Changes needed on the User's Side

On the user's side no changes are needed. The user accesses the system like before. The user doesn't need to install any special software or hardware on his side to access the services of the bank.

5.3 User Creation

Below are the steps for creation of a new user account in an OpenID based system:

- A user goes to the bank to open a new account.
- The user provides his details.
- The bank creates the user policy and sends this information to the IMS system along with other user information.
- The IMS system verifies the user information and provides the user with credentials to access his account.
- The user can then login to his account using the credentials.
- In case of corporate users, if the user is the administrator then he can add more users by means of a web interface at the IMS system directly and decide the account policies for those users.

5.4 User Authentication

Authentication steps are as follows in OpenID based system:

- The user goes to login page.
- The user provides his username and password.
- This is sent to OpenID IMS which verifies the user and creates a pseudonym for the given user ID.
- This user ID to the pseudonym mapping is stored in the database for escrow purposes.

- The IMS gets the policy for the given user ID from the policy database.
- The IMS then sends the pseudonym, account ID and policy to the bank.
- The bank gets this information and creates a temporary policy for the given pseudonym.
- The user can then access services from the bank using the pseudonym.
- All the user's transactions are logged with the pseudonym.

5.5 ID Escrow

Following are the steps for ID escrow in OpenID based system:

- The authorities come to the bank for the transaction data.
- After verifying, bank gives the transaction data to the authorities.
- The authorities then go to the IMS based system for the mapping data.
- After verifying, the IMS gives the mapping data to the authorities.
- The authorities then get the real identity of the user from the mapping and transaction data.

5.6 Analysis

With the use of OpenID IMS we add a *pseudonymous layer* in the system. This provides us the necessary privacy. But in order to do so, the OpenID provider needs access to a lot of data. Some of the example data is:

- UserID
- Account ID
- Policies

In addition to that, the provider needs to store the mapping database from the userID to the pseudonym. The bank really has to trust the provider with storage of all this sensitive data. In some cases the bank might not want the provider to store such data on their premises.

In case there is a discrepancy, the authorities need to go both to the bank to get the transaction data, as well as the provider for the mapping data.

5.7 OpenID implementation in the Real World

Now we will try to fit this implementation in our system, which includes Nykredit as the Bank, Signicat as the 3rd party, DTU as the corporate customer and other government institutions as the authorities.

5.7.1 Addition of the New User

Addition of the new user can happen as follows:

1. DTU registers the new user with Nykredit giving them the user details and policies that should apply to the particular user regarding the account.
    (a) Nykredit registers this new user with his user ID with the IMS system. Nykredit also adds policy for the user in the IMS system.
2. Nykredit issues user credentials for the given user to DTU.
3. DTU then uses this policy credentials to register the new user with Signicat.
    (a) Signicat inquires about the user data with the authorities.
    (b) The authorities verify the user data to Signicat.
4. Signicat issues the final OpenID credentials for the IMS system. These credentials are then used to login to the IMS system by the new user. The flow can be seen in figure 5.2.

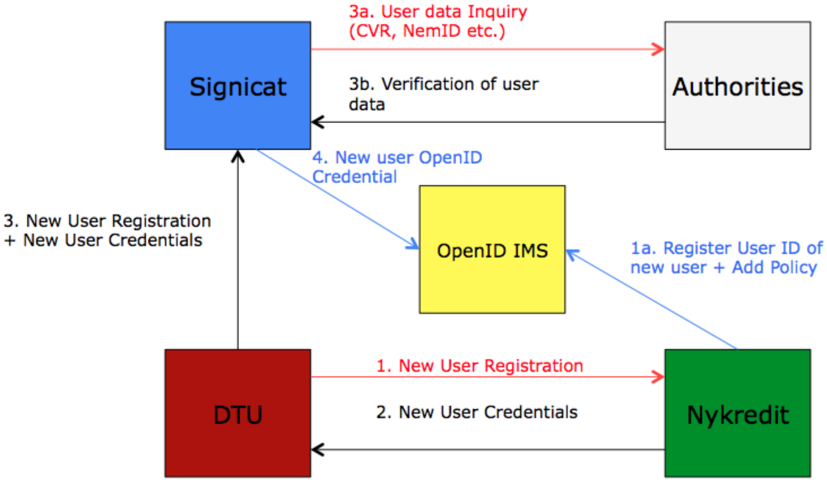


Figure 5.2: OpenID Registration for a new user

5.7.2 Addition of a New Customer

Addition of a new customer is almost the same as the addition of a new user:

1. An administrator goes to Nykredit to open a bank account on behalf of DTU.
   (a) Nykredit registers DTU as a new customer in their internal system.
   (b) Nykredit register the DTU administrator with his user ID with the IMS system.
2. Nykredit issues user credentials for the DTU administrator.
3. The administrator then uses these credentials to register himslef as an owner of the new DTU account with Signicat.
   (a) Signicat inquires about the data given in the credential with the authorities.
   (b) The authorities verify the data to Signicat.
4. Signicat issues the final OpenID credentials for the IMS system. These credentials are then used to login to the IMS system by the administrator.

5.7.3 Technical Requirements

In this system, DTU as a client doesn't need to change anything on their side to be a customer at Nykredit. All the system for DTU is web based, where they can just add/remove users. Moreover, DTU users login to the system using the normal web browser.

Nykredit has to implement the OpenID relying party service on their side. In this case they have to store the sensitive data with the 3rd party. The account details and policies are stored at IMS.

Signicat has to implement the OpenID identity provider service on their side.

IMS has to implement the OpenID identity provider service on their side. The details of the implementation of these services can be found in [16].

This chapter described the IMS system setup using the OpenID system. We described how the system will be setup and how it will affect all the parties involved. Finally we discussed how OpenID IMS will be implemented in the real world scenario.

Chapter 6

IDEMIX Based Solution

In this chapter we will provide a solution using IDEMIX based IMS. We will give more details about how this system can be implemented, and how it will behave for the end users.

6.1 IDEMIX IMS

As in the previous cases, we can replace the IMS with *IDEMIX based IMS* in our pseudonymous system as shown in figure 6.1. This system will take user's credentials and then send an *IDEMIX token* to the bank. This IDEMIX token contains the pseudonym as well as the account ID and policy for the user. This IMS can be controlled by a separate entity inside the bank or by a 3rd party.

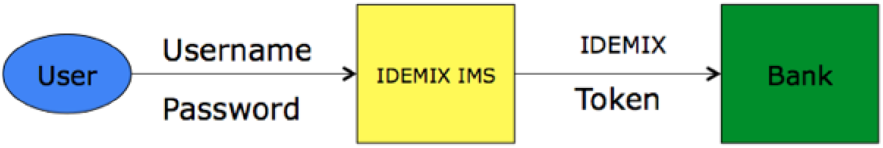


Figure 6.1: Pseudonym System with IDEMIX IMS

6.2 System Setup

The system can be setup in two ways:

- IMS controlled by a separate department at the bank.

  In this case the bank separates the authentication and service part in two different departments internally. Authentication is controlled by the IMS which holds the IDEMIX credentials for the user. The service department only gets the IDEMIX token from the authentication department.

- IMS controlled by a third party

  In this case the bank operates the service part while the authentication part is operated by a trusted third party.

In both cases, the bank and IMS have to collaborate. The bank has to trust the IMS system that the IDEMIX token sent by the the IMS system is correct.

6.2.1 Changes on the Bank's Side

In this system, the bank needs to behave as an IDEMIX *issuer and verifier*. It will issue IDEMIX credentials for the users and it will also verify the tokens sent by the IMS system.

6.2.2 Information Stored at IMS

The IMS system will behave like the user in the IDEMIX system. The IMS needs the following user information to be stored:

- User IDEMIX credential

The account ID and policy can be stored in encrypted form in the credential itself. This IDEMIX credential for a particular user can be setup in the beginning and then can be used later to create authentication tokens.

6.2.3 Changes needed on the User's Side

On the user's side no changes are needed. The user accesses the system like before. He doesn't need to install any special software or hardware on his side to access the bank's services.

6.3 User Creation

Below are the steps for the creation of a new user account in the IDEMIX based system:

- The user goes to bank to open a new account.
- The user provides his details.
- The bank creates the user policy and sends this information to the IMS system along with other user information as an IDEMIX credential.
- The IMS system verifies the IDEMIX credential for the user and provides the user with credentials to access his account.
- The user can then login to his account using the credentials.
- In case of the corporate users, if the user is the administrator then he can add more users using a web interface at the bank directly and decide the account policies for those users.

6.4 User Authentication

Authentication steps are as following in the IDEMIX based system:

- The user goes to login page.
- The user provides his username and password.
- This is sent to the IDEMIX IMS which then gets the saved user credential and creates a presentation token with a pseudonym for the bank.
  - Also, for escrow purposes the real user identity is also encrypted with the public keys of authorities in the token.

- The bank receives this presentation token and gets the following information:
  - Pseudonym
  - Account ID
  - Policy
- The bank adds this information in a temporary policy database.
- The bank saves the token for future escrow purposes.
- The user can then access services from the bank using the pseudonym.
- All user transactions are logged with the pseudonym.

6.5 ID Escrow

Following are the steps for ID escrow in the IDEMIX based system:

- The authorities come to the bank for the transaction data and the IDEMIX token.
- After verifying, the bank gives the transaction data and corresponding IDEMIX token to the authorities.
- The authorities then, using their key, get the real identity of the user from the IDEMIX token.

6.6 Analysis

With the use of IDEMIX IMS we add a pseudonymous layer in the system. This provides us the necessary privacy. In order to do so, IDEMIX IMS just needs to store the IDEMIX credential of the user.

The provider doesn't need to store any mapping database on his side. It is easier for the bank to implement, as the bank doesn't really has to trust the IDEMIX IMS to store sensitive data.

In case there is a discrepancy, the authorities need to go only to the bank to get the transaction data as well as the mapping data from the IDEMIX tokens.

6.7 IDEMIX implementation in the Real World

Now we will try to fit this implementation in our system, which includes Nykredit as the Bank, Signicat as the 3rd party, DTU as the corporate customer and other government institutions as the authorities.

6.7.1 Addition of the New User

Addition of the new user can happen as follows:

1. DTU registers the new user with the Nykredit giving them the user details and policies that should apply to the particular user regarding the account.
   (a) Nykredit registers this new user with his user ID with the IMS system
2. Nykredit issues an *IDEMIX policy credential* for the given user to DTU. This credential contains the policy information and account information for the user.
3. DTU then uses this policy credential to register the new user with Signicat.
   (a) Signicat inquires about the user data with the authorities.
   (b) The authorities verify the user data to Signicat.
4. Signicat issues the *final IDEMIX credential* for the IMS system. This credential is then used to create pseudonym IDEMIX tokens for the user. The whole flow is illustrated in figure 6.2 and figure 6.3.

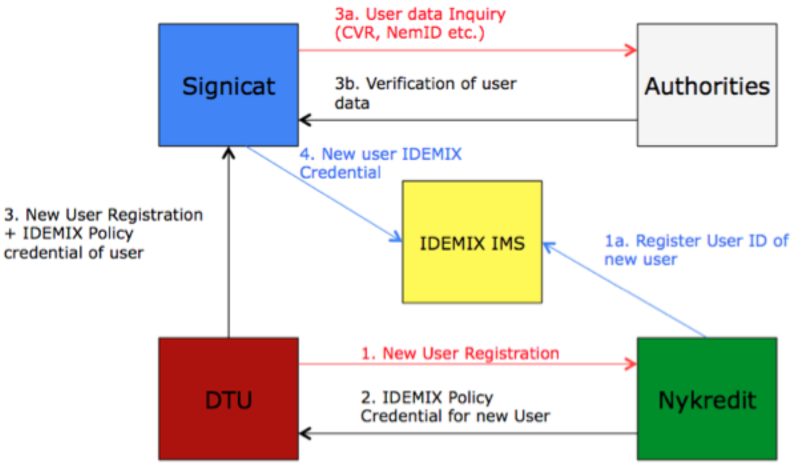

Signicat
Authorities
3a. User data Inquiry (CVR, NemID etc.)
3b. Verification of user data
4. New user IDEMIX Credential
IDEMIX IMS
3. New User Registration + IDEMIX Policy credential of user
1a. Register User ID of new user
DTU
Nykredit
1. New User Registration
2. IDEMIX Policy Credential for new User

Figure 6.2: IDEMIX Registration for a new user

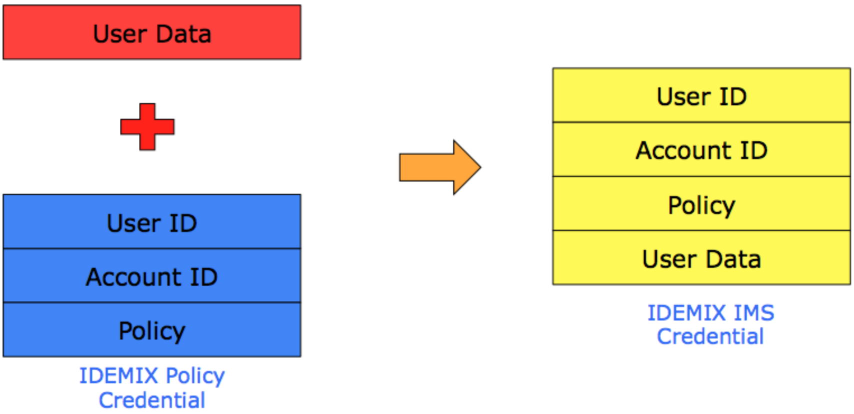


Figure 6.3: Final IDEMIX Credential from Policy Credential

6.7.2 Addition of a New Customer

Adding a new customer is almost the same as adding a new user:

1. An administrator goes to Nykredit to open a bank account on behalf of DTU.
    (a) Nykredit registers DTU as a new customer in their internal system.
    (b) Nykredit registers the DTU administrator with his user ID with the IMS system
2. Nykredit issues an IDEMIX policy credential for the DTU administrator. This credential contains the policy information and account information for the administrator.
3. The administrator then uses his policy credential to register himslef as an owner of the new DTU account with Signicat.
    (a) Signicat inquires about the data given in the credential with the authorities.
    (b) The authorities verify the data to Signicat.
4. Signicat issues the final IDEMIX credential for the IMS system. This credential is then used to create pseudonym IDEMIX tokens for the administrator.

6.7.3 Technical Requirements

In this system DTU as a client does not need to change anything on their side to be a customer at Nykredit. All the system used by DTU is web based where they can just add/remove users. Also DTU users login to the system using the normal web browser.

Nykredit has to implement *IDEMIX issuer service*[21] on their side to issue the IDEMIX Policy credential. This is done so that Nykredit doesn't have to store the sensitive data at the 3rd party. Use of this credential ensures that this data remains safe. Nykredit also has to implement *IDEMIX verifier service* to verify the user identity.

Signicat also has to implement IDEMIX issuer service to issue the final IDEMIX credential.

IMS has to implement the *IDEMIX user service* to create the IDEMIX tokens for the user while the user is logging in.

6.8 High Level Protocol Description

In this section, we will give protocol description of the IDEMIX based system. This is a high level description of the protocol and full details can be found in [20].

We assume that all the systems are secured and all the communication within them is encrypted. Let $Cred_{I,U}(data1, data2, ...)$ be an IDEMIX credential issued by issuer I to user U and $Token_{U,V}(data1, data2, ...)$ be IDEMIX token created by user U for verifier V . We define DTU employee as entity D, Nykredit as entity N, IMS as entity I, Signicat as entity S and authorities as entity A. Also the notation:

$$A \rightarrow B : \{m\}$$

means that a message m is sent from A to B in encrypted form such that only A and B can read it. We take three cases:

1. User Registration
2. User Authentication
3. User transaction

6.8.1 User Registration

The first part of protocol is the user registration. It involves all the parties in the system. The details of the protocol are as mentioned below:

1. Let *username* be the login name of the new user that DTU wants to give access to their corporate account, *account_id* be the account number of DTU account with Nykredit and $policy_D$ be the policy defined by the DTU for the user on their account. DTU sends this information to the bank for the user registration.

$$D \rightarrow N : \{username, account\_id, policy_D\}$$

(a) The bank registers this new user with the IMS system with the given username and receives the password for the user to login to the system.

$$N \rightarrow I : \{username\}$$
$$I \rightarrow N : \{username, password\}$$

2. DTU sends this password as well as an IDEMIX credential for the given username back to DTU.

$$N \rightarrow D :$$
$$\{username, password, Cred_{N,D}(username, account\_id, policy_N)\}$$

Where $policy_N$ is the policy created by Nykredit for the user for the given account. It is a mix of the policy given ty the DTU and also some internal Bank policies. $Cred_{N,D}(username, account\_id, policy_N)$ is the IDEMIX credential issued by Nykredit for the user with the given username.

3. DTU sends the user data and credential, given by Nykredit, to Signicat. This user data may contain *real name,CPR nr,address,contact information* etc. for the user.

$$D \rightarrow S :$$
$$\{username, userdata, Cred_{N,D}(username, account\_id, policy_N)\}$$

4. Signicat verifies the user data with the authorities and then issues its own credential $Cred_{S,I}(userdata, Cred_{N,D}(account\_id, policy_N))$ for the user. This credential contains data from the Nykredit credential $Cred_{N,D}(account\_id, policy_N))$ also. This makes sure that Signicat is able to issue the credential over parameters given by Nykredit credential without need to know the data inside that credential.

Signicat sends this data to the IMS system and the user is registered with his credential in the IMS system.

$$S \rightarrow I : \{username, Cred_{S,I}(userdata, Cred_{N,D}(account\_id, policy_N))\}$$

The user can now login to the bank using the IMS system with his username and password.

6.8.2 User Authentication

For user authentication, only DTU employee,the IMS and Nykredit systems are needed. The protocol works as follows:

1. The user sends his username and password to the IMS system.

   $$D \rightarrow I : \{username, password\}$$

   (a) The IMS system, after verifying the user credentials, creates an IDEMIX token from the user credential.

   $$Cred_{S,I}(userdata, Cred_{N,D}(account\_id, policy_N)) \rightarrow$$
   $$Token_{I,N}(pseudonym, account\_id, policy_N, \{userdata\}PK_A)$$

   pseudonym is the pseudonym generated by the IMS for the user. $Token_{I,N}(pseudonym, account\_id, policy_N, \{userdata\}PK_A)$ is the IDEMIX token created by IMS from $Cred_{S,I}(userdata, Cred_{N,D}(account\_id, policy_N))$. PK_A is the public key of authorities and *userdata* is encrypted using this public key. It is a verifiable encryption[25] and it can be verified that the token has actual *userdata* encrypted using PK_A.

2. The IMS system then sends the IDEMIX token to Nykredit.

   $$I \rightarrow N : Token_I(account\_id, policy_N, \{userdata\}PK_A)$$

3. Nykredit verifies the token and authenticates the user.

   $$N \rightarrow D : pseudonym, session\_secret, status$$

   Where *session_secret* is the secret value that Nykredit established with the user. This value is sent in subsequent requests by the user to Nykredit.

6.8.3 User Transcation

The user transaction happens in the same manner as in the current system. We assume that the user is already authenticated using the steps above. The only entities involved are DTU user and Nykredit in this case:

1. The DTU user sends the transaction request to Nykredit.

   $$D \rightarrow N : session\_secret, transaction_request$$

2. Nykredit performs the transcation and sends the result back.

$$N \rightarrow D : transaction_request, status$$

This chapter described the IMS system setup using the IDEMIX system. We described how the system would be setup and how it would affect all the parties involved.

Chapter 7

Evaluation

This chapter will present the evaluation criteria for our system. After presenting the criteria, we will evaluate the two systems presented in chapters 6 and 7.

7.1 Evaluation Criteria

Below are the evaluation criteria and project goals that we have setup for our system:

7.1.1 Unlinkability

We want to unlink the real identity of the user from the transactions. A user should not have to give his real identity to the bank in order to get the services. Also two different sessions of the same user should be unlinkable i.e. it should not be possible to find out that two sessions are from the same user or two different users.

Being able to link the real identity of the user with the transactions creates a lot of problems. It is possible for someone, who has access to such data, to learn about the financial life of the given individual. So this property is desired to avoid such problems.

7.1.2 Escrow

It should be possible for the authorities to get the real identity of the user in case of legal requirements or discrepancies. But, still the bank should not be able to get the real identities of the users.

Providing anonymity is good but sometimes people take advantage of anonymity on internet. e.g. they might perform some illegal transactions at the bank, while they are anonymous. So the escrow property is required to handle the cases.

7.1.3 Minimal Technical Requirements

It should be easy to be a customer at the bank. The user should not have to change a lot of systems on their side to be a customer. It should also be easier for existing customers to continue using the services of the bank.

In the end, its all about customer. Customers want security but they don't want to sacrifice ease of use. It would be difficult for the bank to keep the customers or to get new customers if it means that they have to invest heavily in IT infrastructure just to be a customer with them. So this requirement is needed to make it easy for customers to get the services from the bank.

7.1.4 User awareness

The user should be aware of the data they are sharing with the bank. The bank should have user consent before storing any data from the user.

People are getting more and more aware of their privacy. They want to know what personal data is being stored by the service providers in order to provide them with the services. This requirement takes care of the case where the user knows exactly what data he is sharing with the bank.

7.1.5 Protection of Data

All the sensitive data about the user should be kept protected. Also, all the bank related sensitive data e.g. account information, policies should remain secure.

Keeping the data protected is a big challenge. If the bank is not able to protect the user's data it gives them bad reputation in the market. Sensitive business data is also valuable to the bank. The bank doesn't want to give this data to anyone else.

7.2 OpenID based solution

Now we will evaluate our OpenID based solution as described in chapter 5.

1. **Unlinkability** In the OpenID based system all the transactions are logged using a pseudonym. Hence the user's real identity is unlinked from the transactions. If the same user logs in again, he is given a different pseudonym, hence its not possible to relate with sessions with each other.

2. **Escrow :** In case the authorities need to get the real identity behind a transaction there is an escrow capability. The IMS system stores the mapping database from the pseudonym to the real identity of the user. The authorities need to go to the bank to get the transactions and then to the IMS provider to get the mapping data.

3. **Minimal Technical Requirements :** For the end users, there aren't many technical requirements. They don't have to setup a special hardware to be a customer at the bank. All the interface is web based and can be used in any normal web browser.

4. **User Awareness :** In the OpenID based system, the user is told during authentication what data about him is being shared with the bank. In case some new data is needed, the user is asked about it.

5. **Protection of Data :** All the sensitive information about the user's personal identity resides with IMS. The bank's business information such as account information and policy also have to be stored at the IMS.

7.3 IDEMIX based solution

Now we will evaluate our IDEMIX based solution as described in chapter 6.

1. **Unlinkability :** In the IDEMIX based system all the transactions are logged using a pseudonym from the IDEMIX token presented by the IMS. Hence, the user's real identity is unlinked from the transactions. If the same user logs in again,IMS creates a new token with a different pseudonym and its not possible to relate 2 sessions with each other.

2. **Escrow :** As during creation of the presentation IDEMIX token, the real identity of the user is put in an inspectable field, so in case authorities need to get the real identity behind a pseudonym in transactions, they can just get it from the IDEMIX token from the bank.

3. **Minimal Technical Requirements :** For the end users, there are not many technical requirements. They don't have to setup a special hardware to be a customer at the bank. All the interface is web based and can be used in any normal web browser.

4. **User Awareness :** IMS makes sure that the user knows what data about him is being shared with the bank in the presentation token. In case new data is needed the user is asked about it.

5. **Protection of Data** All the sensitive information about the user's personal identity resides with IMS. Bank's business information such as account information and policy have to be stored in the IDEMIX credential at the IMS. Only the bank is able to open this information as it is from a credential issued by the bank itself.

7.4 Discussion

We discussed our solutions and evaluations with Nykredit and Signicat. Although it seems promising, the problem is that it is difficult to change the legacy systems. The IDEMIX solution seems better in fulfilling the goals of the system but it is still a new technology. To implement this system, a lot of data that Nykredit holds has to be moved out of the Nykredit systems. Even though that is the goal of the system, it seems like Nykredit is not ready to give up all the data out yet. Some of their internal business processes still rely heavily on the business data stored at their premises.

Signicat, on the other hand, is really interested in the IDEMIX system. This will allow them to add one more service to their portfolio and strengthen their position in the market.

After all the discussion we reached the conclusion that even though the idea is revolutionary, maybe for Nykredit it is better to take one step at a time. Instead of completely anonymizing the system, they just want to replace user ID with a constant pseudonym. They want to keep all the policy and account database with them for the time being and want the IMS system to just replace user ID with a pseudonym for the banking purposes.

In this chapter we evaluated two different pseudonym systems as discussed in chapter 5 and 6. We then presented our discussion with the companies regarding these systems.

Chapter 8

Conclusion and Future work

This chapter will conclude the thesis and will provide directions for the future work that can be carried out in the field.

8.1 Conclusion

To conclude, we can say that technology is changing at an alarming rate. Service providers want to catch up with the technology advances and address the privacy issues currently prevalent. But it is not easy as they are not yet ready to give up on the legacy systems. Also, its difficult to make radical changes in already running systems.

Service providers still rely on the user data they have stored to provide their services and it is difficult to remove that reliability at once. These changes cannot be made overnight but it will take some time for the industry to understand the value in changing their business processes to not to rely that much on the user data.

8.2 Future Work

Providing anonymity in business processes is an interesting field of research, both for the industry and academic point of view. This section provides some insights on how the systems presented in the thesis may affect future research in the area and how it can be used in some alternate ways by the industry.

8.2.1 Reidentification possibilities

In our sytems we have removed the identification from the transactions. As mentioned in [30] the researchers were able to reidentify individuals based on the credit card transactions from the past three months, even though all those transactions were completely anonymized. It would be interesting to put our system into place and use the same methodology to see if the individuals can be reidentified.

8.2.2 Providing services using the third party

One more aspect, as we see from our point of view, is that our system has removed the identity from the the service part of the businesses. As most of the businesses are moving towards outsourcing their IT divisions for cost-cutting purposes, our system provides them a unique opportunity to do so. As there is no identity involved in the service system, the whole service side can be outsourced without the risk of giving away confidential data to the third parties.

This chapter concluded the thesis and provided some insights on future work that can be done in the field.

APPENDIX A

Appendix

A.1 ABC4Trust IDEMIX Implemetation

ABC4Trust has given APIs[31][32] to implement an IDEMIX system. We defined some specifications for this ABC4Trust system. The corresponding XML files are given in this appendix.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<!--
This is a sample ABC4Trust credential specification
-->

<abc:CredentialSpecification xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"
Version="Version 1.0"
KeyBinding="true"
Revocable="true"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://abc4trust.eu/wp2/abcschemav1.0
     ../../../../../../../../../abc4trust-xml/src/main/resources/xsd/schema.xsd">
<abc:SpecificationUID>http://mybank/credential</abc:SpecificationUID>
<abc:FriendlyCredentialName lang="en">Mybank IMS</abc:FriendlyCredentialName>
<abc:DefaultImageReference>http://MyBank/img</abc:DefaultImageReference>
<abc:AttributeDescriptions MaxLength="256">
<abc:AttributeDescription Type="http://abc4trust.eu/wp2/abcschemav1.0/
     revocationhandle" DataType="xs:integer" Encoding="urn:abc4trust:1.0
     :encoding:integer:unsigned"/>
```

```
<abc:AttributeDescription Type="FirstName" DataType="xs:string" Encoding="urn:abc4trust:1.0:encoding:string:utf-8">
<abc:FriendlyAttributeName lang="en">Name</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeDescription Type="LastName" DataType="xs:string" Encoding="urn:abc4trust:1.0:encoding:string:utf-8">
<abc:FriendlyAttributeName lang="en">Lastname</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeDescription Type="Birthday" DataType="xs:date" Encoding="urn:abc4trust:1.0:encoding:date:unix:signed">
<abc:FriendlyAttributeName lang="en">Birthday</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeDescription Type="UserID" DataType="xs:integer" Encoding="urn:abc4trust:1.0:encoding:integer:unsigned">
<abc:FriendlyAttributeName lang="en">User id</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeDescription Type="AccountID" DataType="xs:integer" Encoding="urn:abc4trust:1.0:encoding:integer:unsigned">
<abc:FriendlyAttributeName lang="en">Account ID</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeDescription Type="Withdraw" DataType="xs:boolean" Encoding="urn:abc4trust:1.0:encoding:boolean:unsigned">
<abc:FriendlyAttributeName lang="en">Withdraw</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeDescription Type="Transfer" DataType="xs:boolean" Encoding="urn:abc4trust:1.0:encoding:boolean:unsigned">
<abc:FriendlyAttributeName lang="en">Transfer</abc:FriendlyAttributeName>
</abc:AttributeDescription>
</abc:AttributeDescriptions>
</abc:CredentialSpecification>
```

Listing A.1: IDEMIX Credential Specificataion

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<abc:IssuerParametersInput
xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://abc4trust.eu/wp2/abcschemav1.0 ../../../../../../../../abc4trust-xml/src/main/resources/xsd/schema.xsd"
Version="1.0">
<abc:ParametersUID>http://mybank/issuance:idemix</abc:ParametersUID>
<abc:FriendlyIssuerDescription lang="en">Issuer parameters for Mybank IMS</abc:FriendlyIssuerDescription>
<abc:AlgorithmID>urn:abc4trust:1.0:algorithm:idemix</abc:AlgorithmID>
<abc:CredentialSpecUID>http://mybank/credential</abc:CredentialSpecUID>
<abc:HashAlgorithm>urn:abc4trust:1.0:hashalgorithm:sha-256</abc:HashAlgorithm>
<abc:RevocationParametersUID>http://mybank/revocation</abc:RevocationParametersUID>
</abc:IssuerParametersInput>
```

Listing A.2: IDEMIX Parameters Input

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<abc:IssuancePolicyAndAttributes
```

```
xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://abc4trust.eu/wp2/abcschemav1.0 ../../../../../../../../abc4trust-xml/src/main/resources/xsd/schema.xsd">

<abc:IssuancePolicy Version="1.0">
<abc:PresentationPolicy PolicyUID="http://mybank/issuance/policy">
<abc:Pseudonym Exclusive="true" Scope="http://mybank" Established="false" Alias="#nym"/>
<abc:Message>
<abc:Nonce>KNsRu9cGzkaeabogeRVV</abc:Nonce>
<abc:ApplicationData>
<abc:TestApplicationData>
<abc:Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xsi:type="xs:string">Some data</abc:Data>
</abc:TestApplicationData>
</abc:ApplicationData>
</abc:Message>
</abc:PresentationPolicy>

<abc:CredentialTemplate SameKeyBindingAs="#nym">
<abc:CredentialSpecUID>http://mybank/credential</abc:CredentialSpecUID>
<abc:IssuerParametersUID>http://mybank/issuance:idemix</abc:IssuerParametersUID>
<abc:UnknownAttributes/>
</abc:CredentialTemplate>
</abc:IssuancePolicy>

<abc:Attribute>
<abc:AttributeUID>-5027215341191833963</abc:AttributeUID>
<abc:AttributeDescription DataType="xs:string" Encoding="urn:abc4trust:1.0:encoding:string:sha-256" Type="FirstName">
<abc:FriendlyAttributeName lang="en">first name</abc:FriendlyAttributeName>
<abc:FriendlyAttributeName lang="da">fornavn</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">John</abc:AttributeValue>
</abc:Attribute>
<abc:Attribute>
<abc:AttributeUID>-2715953330829768453</abc:AttributeUID>
<abc:AttributeDescription Type="FirstName" DataType="xs:string" Encoding="urn:abc4trust:1.0:encoding:string:utf-8">
<abc:FriendlyAttributeName lang="en">Name</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Doe</abc:AttributeValue>
</abc:Attribute>
<abc:Attribute>
<abc:AttributeUID>-2231744817504418816</abc:AttributeUID>
<abc:AttributeDescription Type="Birthday" DataType="xs:date" Encoding="urn:abc4trust:1.0:encoding:date:unix:signed">
<abc:FriendlyAttributeName lang="en">Birthday</abc:FriendlyAttributeName>
</abc:AttributeDescription>
```

```xml
<abc:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">1990-07-16Z</abc:AttributeValue>
</abc:Attribute>
<abc:Attribute>
<abc:AttributeUID>-2231744817504418826</abc:AttributeUID>
<abc:AttributeDescription Type="UserID" DataType="xs:integer" Encoding="urn:abc4trust:1.0:encoding:integer:unsigned">
<abc:FriendlyAttributeName lang="en">User id</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">123</abc:AttributeValue>
</abc:Attribute>
<abc:Attribute>
<abc:AttributeUID>-1231744817504418817</abc:AttributeUID>
<abc:AttributeDescription Type="AccountID" DataType="xs:integer" Encoding="urn:abc4trust:1.0:encoding:integer:unix:unsigned">
<abc:FriendlyAttributeName lang="en">Account ID</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:integer">23784638726</abc:AttributeValue>
</abc:Attribute>
<abc:Attribute>
<abc:AttributeUID>-1231744817504418818</abc:AttributeUID>
<abc:AttributeDescription Type="Withdraw" DataType="xs:boolean" Encoding="urn:abc4trust:1.0:encoding:boolean:unix:unsigned">
<abc:FriendlyAttributeName lang="en">Withdraw</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:boolean">1</abc:AttributeValue>
</abc:Attribute>
<abc:Attribute>
<abc:AttributeUID>-1231744817504418819</abc:AttributeUID>
<abc:AttributeDescription Type="Transfer" DataType="xs:integer" Encoding="urn:abc4trust:1.0:encoding:integer:unix:unsigned">
<abc:FriendlyAttributeName lang="en">Transfer</abc:FriendlyAttributeName>
</abc:AttributeDescription>
<abc:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:boolean">1</abc:AttributeValue>
</abc:Attribute>
</abc:IssuancePolicyAndAttributes>
```

Listing A.3: Issuance Policy and Attributes

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<!-- This is a sample ABC4Trust presentation policy for... -->

<abc:PresentationPolicyAlternatives
xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"
Version="1.0"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://abc4trust.eu/wp2/abcschemav1.0
     ../../../../../../../../abc4trust-xml/src/main/resources/xsd/schema.xsd">
<abc:PresentationPolicy
PolicyUID="http://mybank/presentation">
<abc:Message>
<abc:ApplicationData>
Corporate Account at Mybank
</abc:ApplicationData>
</abc:Message>
<abc:Pseudonym Exclusive="xs:boolean"? Scope "xs:string"Established="xs:boolean"?
      Alias="xs:anyURI"? SameKeyBindingAs="xs:anyURI"?>
<abc:PseudonymValue> </abc:PseudonymValue>?
</abc:Pseudonym>*
<abc:Credential Alias="#token">
<abc:CredentialSpecAlternatives>
<abc:CredentialSpecUID>http://mybank/credential</abc:CredentialSpecUID>
</abc:CredentialSpecAlternatives>
<abc:IssuerAlternatives>
<abc:IssuerParametersUID>http://mybank/issuance:idemix</abc:IssuerParametersUID>
</abc:IssuerAlternatives>
<abc:DisclosedAttribute AttributeType="UserID" >
<abc:InspectorAlternatives>
<abc:InspectorPublicKeyUID>http://mybank/inspection</abc:InspectorPublicKeyUID>
</abc:InspectorAlternatives>
<abc:InspectionGrounds>Reveal UserID for inspection.</abc:InspectionGrounds>
</abc:DisclosedAttribute>
</abc:Credential>
<abc:AttributePredicate Function="urn:oasis:names:tc:xacml:1.0
     :function:integer-equal">
<abc:Attribute CredentialAlias="#token" AttributeType="AccountID" />
<abc:ConstantValue>23784638726</abc:ConstantValue>
</abc:AttributePredicate>
<abc:AttributePredicate Function="urn:oasis:names:tc:xacml:1.0
     :function:boolean-equal">
<abc:Attribute CredentialAlias="#token" AttributeType="Withdraw" />
<abc:ConstantValue>1</abc:ConstantValue>
</abc:AttributePredicate>
<abc:AttributePredicate Function="urn:oasis:names:tc:xacml:1.0
     :function:boolean-equal">
<abc:Attribute CredentialAlias="#token" AttributeType="Transfer" />
<abc:ConstantValue>1</abc:ConstantValue>
</abc:PresentationPolicy>
</abc:PresentationPolicyAlternatives>
```

Listing A.4: IDEMIX Presentation Policy

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<abc:RevocationReferences
xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://abc4trust.eu/wp2/abcschemav1.0
     ../../../../../../../../abc4trust-xml/src/main/resources/xsd/schema.xsd">
<abc:RevocationInfoReference ReferenceType="http">
```

```
<abc:References>http://localhost:9500/revocation/updaterevocationinformation</abc:References>
</abc:RevocationInfoReference>
<abc:NonRevocationEvidenceReference ReferenceType="http">
<abc:References>http://localhost:9500/revocation/generatenonrevocationevidence</abc:References>
</abc:NonRevocationEvidenceReference>
<abc:NonRevocationEvidenceUpdateReference ReferenceType="http">
<abc:References>http://localhost:9500/revocation/generatenonrevocationevidenceupdate</abc:References>
</abc:NonRevocationEvidenceUpdateReference>
</abc:RevocationReferences>
```

Listing A.5: IDEMIX Revocation References

Bibliography

[1] Nemid.nu. One login does it all - nemid, [Accessed 18 June 2015].

[2] Sidsel Duch Langpap. Se og hør-gate: Læs lytternes vildeste sladdertips | p3 | dr, [Accessed 18 June 2015].

[3] Signicat. Digital identity & electronic signing - signicat, [Accessed 18 June 2015].

[4] Nykredit.dk. Nykredit, [Accessed 18 June 2015].

[5] Ian Avrum Goldberg. *A pseudonymous communications infrastructure for the internet.* PhD thesis, University of California at Berkeley, 2000.

[6] Gerrit Bleumer. Anonymity. In HenkC.A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 37–38. Springer US, 2011.

[7] Gerrit Bleumer. Pseudonyms. In HenkC.A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 992–994. Springer US, 2011.

[8] Kazue Sako. Verifiable encryption. In HenkC.A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 1356–1357. Springer US, 2011.

[9] Berry Schoenmakers. Zero-knowledge. In HenkC.A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 1401–1403. Springer US, 2011.

[10] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[11] Oded Goldreich. Secure multi-party computation. *Manuscript. Preliminary version*, 1998.

[12] Doerte K Rappe. *Homomorphic cryptosystems and their applications*. PhD thesis, Universität Dortmund, 2005.

[13] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology—CRYPTO 2000*, pages 255–270. Springer, 2000.

[14] Rafael Accorsi. On the relationship of privacy and secure remote logging in dynamic systems. In *Security and privacy in dynamic environments*, pages 329–339. Springer, 2006.

[15] Ivan Damgård and Mads Jurik. A length-flexible threshold cryptosystem with applications. In *Information Security and Privacy, 8th Australasian Conference, ACISP 2003, Wollongong, Australia, July 9-11, 2003, Proceedings*, pages 350–364, 2003.

[16] David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, pages 11–16. ACM, 2006.

[17] D Hardt, J Bufu, and J Hoyt. Openid attribute exchange 1.0-final. *at, Dec*, 5:11, 2007.

[18] Zurich.ibm.com. Ibm research - zurich | computer science | idemix, [Accessed 18 June 2015].

[19] Research.microsoft.com. U-prove - microsoft research, [Accessed 18 June 2015].

[20] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology—EUROCRYPT 2001*, pages 93–118. Springer, 2001.

[21] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 21–30. ACM, 2002.

[22] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Security in communication networks*, pages 268–289. Springer, 2003.

[23] Heiko Roßnagel, Jan Camenisch, Lothar Fritsch, Thomas Gross, Detlef Houdeau, Detlef Hühnlein, Anja Lehmann, and Jon Shamah. Futureid-shaping the future of electronic identity. *Datenschutz und Datensicherheit (DuD)*, 36(3):189–194, 2012.

[24] Ahmad Sabouri, Ioannis Krontiris, and Kai Rannenberg. *Attribute-based credentials for Trust (ABC4Trust)*. Springer, 2012.

[25] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Advances in Cryptology-CRYPTO 2003*, pages 126–144. Springer, 2003.

[26] Christian Paquin. U-prove technology overview v1. 2013.

[27] Christian Paquin and Greg Zaverucha. U-prove cryptographic specification v1. 1. Technical report, Microsoft Technical Report, http://connect. microsoft. com/site1188, 2011.

[28] Christian Paquin and Greg Zaverucha. U-prove collaborative issuance extension. 2014.

[29] Greg Zaverucha. U-prove id escrow extension. Technical report, TechReport MSR-TR-2013-86, 2013.

[30] Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, et al. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

[31] Patrik Bichsel, Jan Camenisch, Maria Dubovitskaya, Robert R Enderlein, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Janus Dam Nielsen, Christian Paquin, Franz-Stefan Preiss, et al. H2. 2–abc4trust architecture for developers. *ABC4Trust heartbeat H*, 2:2, 2013.

[32] Jan Camenisch, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Christian Paquin, and Kai Rannenberg. H2. 1-abc4trust architecture for developers. *Heartbeat*, 2:1, 2012.