Literature Survey

Enhancing the Privacy of Users in eID schemes through Cryptography

Author: Kris Shrishak

Supervisors: Asst. Prof. Dr. Zekeriya Erkin
Remco Schaar MSc.

Cyber Security Group
Department of Intelligent Systems

January 2016
Contents

List of Figures iii
List of Abbreviations iv

1 Introduction 1

2 Preliminaries 4
  2.1 Homomorphic Encryption 4
     2.1.1 Paillier cryptosystem 4
     2.1.2 ElGamal cryptosystem 5
  2.2 Block chain 6
     2.2.1 Transaction 6
     2.2.2 Proof-Of-Work (POW) 7
     2.2.3 Merkle Hash Tree 8
  2.3 Discussion 9

3 Existing eID systems 10
  3.1 Belgian eID system 10
     3.1.1 Authentication 11
     3.1.2 Revocation 12
     3.1.3 Privacy Analysis 12
  3.2 GOV.UK Verify 12
     3.2.1 Authentication 12
     3.2.2 Privacy Analysis 13
  3.3 German eID system 14
     3.3.1 Authentication 15
     3.3.2 Revocation 16
     3.3.3 Privacy Analysis 16
  3.4 Dutch eID system using polymorphic pseudonyms 17
     3.4.1 Authentication 18
     3.4.2 Privacy Analysis 19
  3.5 I Reveal My Attributes (IRMA) 20
     3.5.1 Authentication 21
     3.5.2 Revocation 22
     3.5.3 Privacy Analysis 22
  3.6 Discussion 23
4 Privacy Enhancing Solutions

4.1 Somewhat Homomorphic Encryption (SHE) ........................................ 24
  4.1.1 SHE using Ideal lattices ............................................................ 25
  4.1.2 SHE using Integers ................................................................. 26
  4.1.3 SHE based on Ring-Learning with Errors ................................. 28

4.2 Zerocash ....................................................................................... 29

4.3 Enigma ......................................................................................... 31

4.4 Discussion ..................................................................................... 32

5 Discussion and Future Work .......................................................... 33

Bibliography ....................................................................................... 34
List of Figures

2.1 A transaction chain [27]. ................................................. 7
2.2 Block chain [27]. .......................................................... 7
2.3 Merkle Hash tree of the transactions [27]. ......................... 8
2.4 Block chain with Merkle root [27]. ................................... 8

3.1 Authentication in Belgian eID system. .............................. 11
3.2 Authentication in GOV.UK Verify. ................................. 13
3.3 Authentication in German eID system. ............................ 16
3.4 Authentication in the Dutch eID system. ......................... 19
3.5 A credential with four attributes. ................................. 21
3.6 Authentication in IRMA. ............................................... 22

4.1 Zerocash [85]. .............................................................. 30
4.2 Enigma [87]. ............................................................... 31
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>eID</td>
<td>electronic IDentity</td>
</tr>
<tr>
<td>LoA</td>
<td>Level of Assurance</td>
</tr>
<tr>
<td>PII</td>
<td>Personally Identifiable Information</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>PET</td>
<td>Privacy Enhancing Technology</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest Shamir Adleman</td>
</tr>
<tr>
<td>DDH</td>
<td>Decisional Diffie-Hellman</td>
</tr>
<tr>
<td>POW</td>
<td>Proof-Of-Work</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider</td>
</tr>
<tr>
<td>IDP</td>
<td>IDentity Provider</td>
</tr>
<tr>
<td>IRMA</td>
<td>I Reveal My Attributes</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>ATP</td>
<td>ATtribute Provider</td>
</tr>
<tr>
<td>MS</td>
<td>Matching Service</td>
</tr>
<tr>
<td>CVC</td>
<td>Card Verifiable Certificate</td>
</tr>
<tr>
<td>PACE</td>
<td>Password Authenticated Connection Establishment</td>
</tr>
<tr>
<td>EAC</td>
<td>Extended Access Control</td>
</tr>
<tr>
<td>TA</td>
<td>Terminal Authentication</td>
</tr>
<tr>
<td>CA</td>
<td>Chip Authentication</td>
</tr>
<tr>
<td>PP</td>
<td>Pseudonym Provider</td>
</tr>
<tr>
<td>KMA</td>
<td>Key Management Authority</td>
</tr>
<tr>
<td>CIPEI</td>
<td>Central Information Point e-ID Investigations</td>
</tr>
<tr>
<td>ABC</td>
<td>Attributes-based Credential</td>
</tr>
<tr>
<td>FHE</td>
<td>Fully Homomorphic Encryption</td>
</tr>
<tr>
<td>SHE</td>
<td>Somewhat Homomorphic Encryption</td>
</tr>
<tr>
<td>LWE</td>
<td>Learning With Errors</td>
</tr>
<tr>
<td>MPC</td>
<td>Multi Party Computation</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Long before the Internet came into existence, Governments have had public authentica-
tion schemes by issuing Identity documents to identify a person or verify aspects of a
person’s personal identity. These documents were trusted not only by the governments
but also by businesses that required reliable authentication of the users [1]. But those
were days when transactions - exchange of information, goods, services and money - in-
volved face to face interactions. Today the number of transactions performed online by
users is increasing rapidly. A number of these transactions, including those involving the
services of the government, require the user to be able to authenticate and to electron-
ically sign documents and has forced Governments to move from paper-based identity
documents to Electronic identities (eIDs). This transition also allows Governments to
improve the security and reduce frauds.

A substantial or high level of assurance (LoA) of the identity of the user is required
for transactions that involve Personally Identifiable Information (PII) [2]. But service
providers such as webshops, tax authorities, etc. require the user to create an account at
their website where the authentication mechanism is usually password-based. Password-
based authentication has multiple requirements such as minimum length of the password,
the presence of alphabets, numbers and special characters in the password, to be accepted
for authentication. Complex passwords are difficult for the users to remember. Users of-
ten employ the same password on multiple sites or write them on paper or stored them on
the computer; thus exposing the passwords to others. An alternate approach, as imple-
mented by a large number of European banks, is to use smart cards along with a Personal
Identification Number (PIN) to authenticate users [3]. Here the user is inconvenieneced
by the need to carry multiple cards and the readers wherever they go if they want to
avail the service. The usage of different tokens by each service provider to authenticate
and identify users has lead to the situation where the governments of most European
countries have either developed an eID scheme or are in the process of developing one so
that users can use a single identification scheme and authenticate themselves to multiple
service providers. Governments envision that eID schemes will be as successful as the
earlier paper-based versions where the credentials issued by the government is used by
businesses as well as public services.

There are transactions for which the identity of the user is not required. For instance,
when one wants to buy liquor, it maybe necessary for the customer to prove that his/her
age is above 18 years but there is no need for the shopkeeper to know the date of birth.
Conventional ID cards also mention the address and place of birth among other PII.
This kind of data leakage should be avoided and data sharing should be minimized to
the essentials to prevent misuse [4]. The user should be able to complete transactions anonymously, without revealing their identity [5]. Most users use multiple services online and would desire that the service providers are not able to link their activities if they collude. It is essential that designers of eID systems focus on privacy, specifically informational privacy of users [6]. Informational privacy should be interpreted such that when a user mentions ‘my information’, the ‘my’

“is not the same as ‘my’ in ‘my car’ but rather the same as ‘my’ in ‘my body’ or ‘my feelings’; it expresses a sense of constitutive belonging, not of external ownership, a sense in which my body, my feelings, and my information are part of me but are not my possessions [7].”

In this study, existing and proposed eID systems are discussed and information privacy is investigated in terms of the following properties [8]:

1. Anonymity: Users may use a service without disclosing their identity.
2. Pseudonimity: Users may utilise a service by using pseudonyms.
3. Data minimization: Only the required information about the user must be shared in order to prevent misuse.
4. Unlinkability: User activity at multiple service providers should not be linkable.
5. Unobservability: Users should be able to use services without being observed by others.
6. Transparency: User data should be obtained only when necessary and after user consent.

Cryptography is extensively used to secure data while in storage, in transit as well as provide authentication of users in eID systems [9]. In addition to security, it can also improve the privacy of users. One method to enhance the privacy of users is to prevent unauthorized parties from accessing user data. Homomorphic encryption provides a way for parties to process encrypted user data without decrypting it [10]. This way only the user and the service provider can access the user data while intermediate parties are able to fulfil their role without accessing user data. Another method would be to decentralise the role played by central parties in eID systems through blockchain, which is a public ledger that is shared by all parties participating in the network [11]. This way user data will not be stored in a centralized database and a privacy hotspot can be eliminated. Though these methods improve privacy of users, additional challenges in the design of the system are encountered. Policies and legislation may also be employed to improve privacy but they are not discussed in this report.

But the question remains, why should businesses and governments bother about user privacy? Businesses should incorporate Privacy Enhancing Technologies (PETs) for the following reasons:

• *Simplifies compliance with data protection legislation*: PETs help in avoiding possible fines enforced by data protection authorities in the case of data breach [2].
• *Avoid Reputation damage*: Large amount of private data has been obtained by hackers through data breaches[12][13]. Data breaches cost companies in millions
and customers may sometimes drag them to court for years. Companies also lose customers due to data breaches and consequentially loss of trust [16].

- Gain Competitive advantage: Businesses can gain privacy aware customers if they offer services which care for user privacy [17].

Governments should incorporate PETs for the following reasons:

- PETs can provide an environment of greater trust for the government with the citizens. In the past government officials have been known to snoop on confidential data [18]. In addition government bodies are also vulnerable to hacking and data breaches [19]. If the citizens have greater trust and confidence in the government then more schemes and initiatives can be proposed and passed easily.

- Privacy provisions will allow the government to collect user data and share it with required parties transparently and thus employing transparency in the manner in which the government is run.

- Identity theft is a horrific experience for victims [20]. Being accused of crimes that someone else has committed is certainly not a good experience. Media coverage of such stories could negatively affect the results of the following elections for the ruling party as the citizens may not trust the government to provide a safe haven.

- The future is uncertain. Databases with identifiable information about the citizens could be used to attack certain sections of the society as was the case during the Holocaust when the population registry in Amsterdam was used to track down Jews [21].

This report focuses on privacy-related challenges in eID systems and is organized as follows: Chapter 2 provides the background information on Homomorphic encryption and Block chain. Chapter 3 provides an overview of some of the existing and proposed eID systems. Chapter 4 discusses few protocols which can be used to solve privacy-related challenges in eID systems and finally in Chapter 5 open issues are discussed and conclusions are drawn.
Chapter 2

Preliminaries

This chapter provides a brief description of homomorphic encryption followed by two partially homomorphic cryptosystems. Then blockchain is described succinctly. The chapter is concluded with discussions.

2.1 Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows processing of ciphertexts and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintexts [10]. It can be defined as follows:

Let \( M \) be the set of plaintexts and let \( C \) be the set of ciphertexts. An encryption scheme is homomorphic if for any given key \( k \) the encryption \( E \) satisfies

\[
\forall m_1, m_2 \in M, \quad E_k(m_1) \odot E_k(m_2) = E_k(m_1 \diamond m_2),
\]

for some operators \( \diamond \) in \( M \) and \( \odot \) in \( C \).

If \((M, \odot)\) and \((C, \odot)\) are two groups, then the Equation 2.1 represents group homomorphism. If \( \diamond \) is an addition operator, then the scheme is said to be additively homomorphic and if \( \diamond \) is a multiplication operator, then multiplicatively homomorphic.

Unpadded-RSA [22] is the first public-key encryption scheme with a homomorphic property. However, for semantic security [23], messages are padded with random bits before encryption. The padding results in RSA losing the homomorphic property. In the last three decades, many public-key encryption schemes with homomorphic properties such as Goldwasser–Micali [23], Paillier [24] and ElGamal [25] have been proposed. Paillier and ElGamal are described in this chapter.

2.1.1 Paillier cryptosystem

Paillier cryptosystem [24] is a probabilistic public-key cryptosystem proposed by Pascal Paillier and whose security is based on the decisional composite residuosity assumption. This mathematical assumption states that it is hard to compute the \( n^{th} \) residue classes. The cryptosystem is composed of key generation, encryption and decryption algorithms.

Key Generation

To generate the keys,
• Choose two large prime numbers $p$ and $q$ randomly and independently of each other, such that $\gcd(pq, (p - 1)(q - 1)) = 1$.
• Compute $n = pq$ and $\lambda = \phi(n) = (p - 1)(q - 1)$.
• Set $g = n + 1$ and $\mu = \lambda^{-1} \mod n$.

Public key $pk = (n, g)$ and private key $sk = (\lambda, \mu)$.

Encryption

If $m \in \mathbb{Z}_n$ is the message, then to encrypt $m$,

• Select a random $r$ where $r \in \mathbb{Z}_{n^2}^*$.
• Compute ciphertext $E(m) = c = g^m \cdot r^n \mod n^2$.

Decryption

To decrypt the ciphertext $c \in \mathbb{Z}_{n^2}^*$, compute the plaintext message as

$$m = L(c^\lambda \mod n^2) \cdot \mu \mod n,$$

where $L(u) = \left(\frac{u - 1}{n}\right)$.

Additive homomorphism

Let $E(m_1)$ and $E(m_2)$ be two ciphertexts, then their product gives

$$E(m_1) \cdot E(m_2) = (g^{m_1} \cdot r_1^n) \cdot (g^{m_2} \cdot r_2^n) = g^{m_1 + m_2} \cdot (r_1 \cdot r_2)^n$$

On decryption of the Equation 2.2, $m_1 + m_2$ is obtained. So Paillier cryptosystem is additively homomorphic such that given encryptions of $m_1$ and $m_2$, the encryption of $m_1 + m_2$ can be obtained without knowing the secret key. However, the encryption of $m_1 \cdot m_2$ cannot be obtained without knowing $m_1$ or $m_2$ first.

2.1.2 ElGamal cryptosystem

ElGamal cryptosystem is a public key cryptosystem proposed by Taher ElGamal in [25] and whose security relies on the hardness of the decisional Diffie-Hellman (DDH) problem [26]. If $G$ is a cyclic group of order $q$ with generator $g$, then the DDH assumption states that, given $g^a$ and $g^b$ for uniformly and independently chosen $a, b \in \mathbb{Z}_q$, the value $g^{ab}$ “looks like” a random element in $G$.

Key generation

To generate the keys,

• Let $g$ be a generator of group $G$ chosen randomly from $\mathbb{Z}_q$.
• Generate the secret key $s \in \mathbb{Z}_q$, and the public key $h = g^s$.

Public key $pk = (G, q, g, h)$ and private key $sk = (s)$. 

Chapter 2. Preliminaries

5
Encryption

Let $m$ be the message mapped onto $m' \in G$, then encryption is performed as follows:

- Select a random $r \in \mathbb{Z}_q$.
- Compute $c_1 = g^r$.
- Compute $c_2 = m \cdot h^r$.
- The ciphertext $E(m) = c = (c_1, c_2) = (g^r, m \cdot h^r)$.

Decryption

The ciphertext is decrypted and the plaintext is obtained by computing $m = c_2 \cdot (c_1^s)^{-1}$.

Multiplicative homomorphism

Let $E(m_1)$ and $E(m_2)$ be two ciphertexts, then their product gives

$$E(m_1) \cdot E(m_2) = (g^{r_1}, m_1 \cdot h^{r_1}) \cdot (g^{r_2}, m_2 \cdot h^{r_2})$$
$$= g^{r_1+r_2}, (m_1 \cdot m_2) \cdot h^{r_1+r_2}$$
$$= E(m_1 \cdot m_2).$$

$m_1 \cdot m_2$ is obtained on decrypting Equation 2.3. Given encryptions of $m_1$ and $m_2$, the encryption of $m_1 \cdot m_2$ can be obtained without knowing the secret key. Thus ElGamal cryptosystem is multiplicatively homomorphic. However, the encryption of $m_1 + m_2$ cannot be obtained without knowing both $m_1$ and $m_2$.

2.2 Block chain

Block chain was introduced by Satoshi Nakamoto as a timestamp server as part of the Bitcoin protocol [27]. A block chain is a public ledger shared by all nodes participating in a system based on the Bitcoin protocol [11]. A full copy of a block chain contains every transaction ever executed. Every block contains a hash of the previous block such that a chain of blocks is created from the first block of the chain, also known as genesis block, to the current block. This way the blocks are arranged in chronological order. It is also computationally infeasible to modify a block as every block that follows must also be regenerated. These properties prevent double-spending of bitcoins [11]. The components that make up the block chain are further explained in this section.

2.2.1 Transaction

A transaction as seen in Fig. 2.1 is a transfer of Bitcoin value which is broadcast to the network and collected into blocks. A transaction typically references the previous transaction by concatenating it with the public key of the new owner and hashing it. The current owner digitally signs this hash and sends it along with the public key of the new owner. A payee can verify the signatures to verify the chain of ownership.
2.2.2 Proof-Of-Work (POW)

The blocks in the block chain are created by full-nodes in the network, which are called miners. A miner receives transactions from other nodes in the Bitcoin network. Every transaction is verified on arrival so that an attacker cannot maliciously transmit transactions to double spend a bitcoin. But the transactions are received non-deterministically which causes blocks to differ from miner to miner. Thus there needs to be a method to obtain consensus on the order of transactions among the nodes. Bitcoin uses a Proof-Of-Work (POW) system based on Hashcash [28].

As part of the POW system, a nonce is added to every block (Fig. 2.2). The nonce is just a number but it is only accepted if the hash of the whole block begins with a certain number of zeros. Miners have to find the correct nonce for their block. As an incentive, the miner who finds the block receives a transaction fee in addition to miner’s fee to expend on CPU time and electricity. It is possible that multiple miners find a valid nonce at approximately the same time and notify parts of the network of their newly found block. To solve this inconsistency, Bitcoin nodes save both branches and continue using the longest branch. At some point one branch will become predominant in the network as more nodes will dedicate computing power to extend this branch and the smaller branch is then abandoned.
The Bitcoin protocol states that the average time taken to find a block should be 10 minutes. Every two weeks, the number of zeros needed in the beginning of the hash of the transaction is adjusted to compensate for the fluctuating speed of the network to find a block. The difficulty of finding a block is directly proportional to the number of zeros.

**Figure 2.3:** Merkle Hash tree of the transactions [27].

**Figure 2.4:** Block chain with Merkle root [27].

### 2.2.3 Merkle Hash Tree

When a miner receives one or more new transactions, they are collected into the transaction data part of a block. Copies of each transaction is hashed, and the hashes are then paired and hashed, paired again and hashed again until a single hash remains, the merkle root of a merkle tree (Fig. 2.3) [29]. The merkle root is stored in the block header (Fig. 2.4) instead of the transactions (Fig. 2.2) in order to prevent the block chain from bloating. Transaction verification can be performed using Merkle Hash trees without checking the transaction itself.
2.3 Discussion

Homomorphic encryption provides a way for parties to process encrypted data. But partially homomorphic cryptosystems have drawbacks. These schemes are not well suited for every use and their characteristics need to be considered before applying. Paillier cryptosystem is not homomorphic under multiplication while ElGamal cryptosystem is not homomorphic under addition. Thus they can be used under restricted conditions only. If a cryptosystem is both additively and multiplicatively homomorphic, then arbitrary computation can be performed on encrypted data [30].

Block chain allows to eliminate the necessity of a central party. But it introduces additional issues. A public ledger provides everyone in the network access to the data on the block chain. This means that instead of one central party having access to all the data, now all parties in the network have access. So the block chain as it is cannot be used to store private data and hence does not address the problem of privacy but merely shifts the problem. Even though Bitcoin protocol allows the usage of multiple pseudonyms or public keys, it is possible to link the activity of users [31]. Thus methods to use the decentralized ecosystem in a privacy friendly manner needs to be further investigated.
Chapter 3

Existing eID systems

eID systems intend to provide unique and reliable identification and authentication of the users. The eID systems discussed in this chapter were designed for use by both public and private services. The parties involved in these systems differ widely but commonly involve the following:

- **User** – wants to authenticate her/himself to the Service Provider to access a resource.
- **Service Provider** (SP) - provides a service, such as online shopping or government tax services, and makes transaction decisions based upon the acceptance of a user’s authenticated credentials and attributes.
- **Identity Provider** (IDP) - verifies the user’s identity or credentials and facilitates the user to authenticate her/himself to the SP. It improves the overall usability since the user does not need to remember multiple authentication credentials.

In this chapter, some of the existing eID systems - the Belgian eID system, GOV.UK Verify, German eID system, Dutch eID system using polymorphic pseudonyms and I Reveal My Attributes (IRMA) - are described.

3.1 Belgian eID system

The Belgian eID system is a nation-wide Public Key Infrastructure (PKI) which requires each citizen to present her/himself at the municipality for strong user authentication during the issuing phase. Thus it can be inferred that the Belgian government has taken up the role of IDP. The user is issued a smart card and is required to buy a card reader for online use. The objective of the Belgian eID card has been to fulfil four functions, namely, citizen identification, authentication, digital signature and access control [9]. These objectives are accomplished using three RSA key pairs and storing the private keys on the eID card for the following purposes [32]:

1. To authenticate the citizen.
2. Non-repudiation.
3. To identify the e-ID card towards the Belgian government.

The first key pair is accompanied by a X.509v3 [33] authentication certificate while the second key pair is accompanied by a X.509v3 qualified certificate that binds the non-repudiation key to the card holder. The private key of the third key pair is used when
the card communicates with the National Register (RRN) for mutual authentication and to update card holder details, such as the address. In addition to the two card holder certificates, three government-specific certificates - the Belgium Root Certificate Authority (CA) certificate, the Citizen CA certificate, and the RRN certificate - are also stored in the eID card.

The eID card has the name, title, nationality, place and date of birth, gender, and a photo of its holder printed on it in addition to a hand written signature of its holder and of the civil servant who issued the card. All this information is also stored on the chip in an Identity file which is signed by the RRN. The chip also contains an address file which is kept independently as the address of its holder may change within the validity period of the card. The RRN signs the address file together with the identity file to guarantee the link between these two files. The corresponding signature is stored as the address file’s signature.

3.1.1 Authentication

When a user visits a SP, say a webshop, which requires her/him to authenticate her/himself, the following sequence of events take place (Fig. 3.1):

1. The user requests a resource from SP.
2. The SP sends a random challenge to the user’s browser.
3. The user confirms she/he wants to log in on the web site by presenting her PIN to her eID card and authorizes the signature generation.
4. The browser sends the hashed challenge to the user’s eID card to sign it.
5. The browser retrieves the signature and user’s certificate from the eID card.
6. The web server receives user’s signature and certificate.
Chapter 3. Existing eID systems

7. SP grants or denies the user access to the requested resource.

3.1.2 Revocation

In Belgium it is a legal obligation for all citizens to carry an identity card. The loss of such a card must be reported promptly. Then the corresponding certificate is suspended for up to 7 days. If the citizen finds her eID card back before this 7-day period ends, then the card can be unsuspended. Otherwise it is revoked. Revocation is performed using Certificate Revocation Lists (CRL) which are maintained by each CA.

3.1.3 Privacy Analysis

The main concern with the Belgium eID card is privacy. Privacy was not a design criteria and no aspect of privacy mentioned in Chapter 1 has been addressed. The user’s identity is revealed for all transactions and she/he does not have control over the data that is shared with the SPs while it remains possible for colluding SPs to link user activities.

3.2 GOV.UK Verify

GOV.UK Verify is the eID system of the United Kingdom [34]. It is a brokered identification infrastructure where an online central hub mediates user authentications between IDPs and SPs. The role of the hub is to ensure interoperable identification and authentication as well as provide privacy benefits by hiding the IDP from the SP. The eID system has been designed considering nine Identity Assurance Principles - user control, transparency, multiplicity, data minimization, data quality, service user access and portability, certification, dispute resolution and exceptional circumstances [35]. The role of IDP will be taken up by private sector organisations, such as Mydex [36], and Digidentity [37], that are certified by the government as meeting relevant security and service standards [38]. In addition to IDPs and SPs, the system also includes (1) a hub, which has been mentioned earlier (2) Attribute Providers (ATP), which are responsible for establishing attributes and (3) Matching Service (MS), which helps validate assertions from IDPs [39].

3.2.1 Authentication

The SP identifies and authenticates a user based on the ability of the user to authenticate to an IDP. The SP identifies the user through the user pseudonym and the personal attributes associated with the pseudonym. The IDP derives a pseudonym for each user at a hub. The pseudonym ($u$) is supposed to be pseudorandom and is used for all transactions of a user. SP learns a pseudonym $v$ from the hub which is associated with the pair $(u, SP)$. Some attributes, such as the address, is validated by the IDP while in some complex scenarios an ATP is required. For example, the user may need to provide proof of insurance in a hospital in addition to the basic attributes.

The online authentication process is described as follows (Fig. 3.2):

1. The user requests a resource from the SP.
2. The SP sends an authentication request to the hub along with the choice of MS.
3. The hub then displays the list of IDPs for the user to choose.
4. The user is redirected to the IDP.
5. The user is authenticated at the IDP.
6. The IDP sends a signed assertion with the user pseudonym \( u \) and attributes to the hub.
7. In case additional attributes are required, then the hub requests them from the respective ATP for the associated \( u \).
8. The hub relays the assertion from IDP to the MS as earlier indicated by the SP. The MS validates the signature of the IDP and derives a new user pseudonym \( v \) that is equal for all SPs that choose this MS. The MS also verifies that \( v \) and attributes match to a local user account. Finally, the MS re-signs a new assertion and sends it to the hub.
9. The hub re-signs the assertion and sends it to the SP.
10. The SP responds to the user granting or denying access to the resource requested.

![Figure 3.2: Authentication in GOV.UK Verify.](image)

### 3.2.2 Privacy Analysis

In spite of claiming that privacy is one of the design criteria for the eID scheme, GOV.UK Verify has multiple privacy issues which are discussed below[39].

- The hub, which has full visibility of the user pseudonym (\( u \)) defined by the IDP, can link interactions of the same user across different SPs and has visibility over the PII of citizens even though [35] mentions that “No relationships between parties or records should be established without the consent of the user”.
• A malicious hub can undetectably impersonate users at any SP without user au-
\hspace{1em}thentication.
• A malicious hub can send the assertion to more than one MSs and obtain multiple
\hspace{1em}user identifiers.
• The MS has the task of matching \( u \) and the attributes into a local account. Thus,
\hspace{1em}the MS can link the user and the SPs that choose the same MS.
• Colluding SPs using the same MS can link the user as the pseudonym (\( v \)) is the
\hspace{1em}same.
• The attributes which contain PII are visible to the hub and the MS, even though
\hspace{1em}the goal of the transaction is to connect the user to the SP.

Thus it can be inferred that GOV.UK Verify actually degrades the privacy of citizens
\hspace{1em}instead of enhancing it. The hub can be used for undetectable mass surveillance.

### 3.3 German eID system

The German eID system uses a direct authentication eID infrastructure. The design
\hspace{1em}goals of this system were data minimization, data security and transparency. In order to
\hspace{1em}use this system, the user needs an eID card, a reader and the Ausweisapp software while
\hspace{1em}the SP needs an authorization certificate, an eID-Server that handles authentication
\hspace{1em}by communicating with the card. The objective of the German eID card has been to
\hspace{1em}fulfil three function, namely, mutual electronic proof of identity of user and SPs, digital
\hspace{1em}signature and border control. In the following mutual identification is discussed further.

The eID card has the name, title, nationality, place and date of birth, colour of eyes,
\hspace{1em}height, residential address and a photo of its holder printed on it in addition to a hand
\hspace{1em}written signature of its holder and the document number. The eID card also contains
\hspace{1em}a chip which stores the information printed on the card as well as the fingerprints of
\hspace{1em}the holder, if the holder wishes. The document number and the fingerprints can be
\hspace{1em}read offline only by authorities who have machines certified by the Federal Office for
\hspace{1em}Information Security (BSI).

The German eID system makes use of protocols, to perform mutual identification,
\hspace{1em}which are also used in EU passports. These protocols need to be executed in the order
\hspace{1em}described as follows [40]:

- **Password Authenticated Connection Establishment (PACE):** PACE is a
\hspace{1em}password authenticated Diffie-Hellman key agreement protocol [41] that provides
\hspace{1em}secure communication and explicit password-based authentication of the eID card
\hspace{1em}and the terminal.

- **The Extended Access Control protocol (EAC):** EAC provides secure key est-
\hspace{1em}ablishment between a chip card and a terminal, using a PKI. It serves the purpose
\hspace{1em}of limiting access to the sensitive data stored on the chip card. It consists of two
\hspace{1em}phases:

  - **Terminal Authentication Version 2 (TA2):** TA2 provides a challenge-
\hspace{1em}response-based proof of the authenticity and the access rights of the SP ter-
\hspace{1em}minal to the data on the eID card chip.
– **Chip Authentication Version 2 (CA2):** CA provides a proof of authenticity of the chip and sets up a secure messaging channel between the chip and the terminal authenticated by the preceding TA2. The terminal’s ephemeral public key computed by the chip during CA2 is compared to the ephemeral public key generated by the terminal during TA2. If it matches, then both chip and the terminal move ahead to agreeing on a shared secret key. To provide pseudonymity, the same key is used for a sufficiently large group of chips.

– **Chip Authentication Version 3 (CA3):** This version of the Chip Authentication combined with Restricted Identification provides authentication of the sector-specific identifier towards the terminal and also provides pseudonymity to the users without the need to use the same keys on several chips. It is based on the combination of an ephemeral key agreement with a Pseudonymous Signature.

• **Restricted Identification:** Restricted Identification is a static Diffie-Hellman key agreement protocol that generates a sector-specific identifier for each card, enabling the pseudonymous identification of the card-holder.

• **Pseudonymous Signatures:** Pseudonymous Signatures provide an efficient way to authenticate/sign data pseudonymously as the public key for verification of signatures is sector-specific. The same sector set-up as for Restricted Identification is used.

To guarantee the authenticity of users and SPs, the German eID system makes use of two PKIs:

• **Country Signing Certificate Authority (CSCA),** operated by the BSI, generates the CSCA root certificates which in turn serve as the source for the private keys of the document signing certificates of the ID card manufacturer. The root certificate can be used to verify the authenticity of the eID card.

• **Country Verifying Certificate Authority (CVCA),** also operated by BSI, generates the CVCA root certificates whose private keys are used to sign the certificate of the document verifiers (DVs), who are responsible for issuing the certificates authorizing the SPs for reading the eID and also define the individual read rights. Thus the root certificate is used to protect the biometric data stored on the eID document and to verify the authorization and the access rights of the SPs during TA2.

### 3.3.1 Authentication

The online authentication process is described as follows (Fig. 3.3):

1. The user, for instance, wants to buy a product from a webshop, for which the SP requires the user to authenticate with her/his eID card.

2. The request is passed on to the eID Service.

3. The eID Service first authenticates the SP and sends the SP’s authorization certificate to the user. The eID Service accesses the data stored on the chip after the
user enters the eID PIN to express consent. This PIN is used locally to execute the PACE protocol.

4. The eID-Client application (Ausweisapp) establishes a TLS channel to the eID-Server and displays the list of attributes requested by the SP to the user on her/his browser. If necessary, the citizen can restrict access to these attributes. Another secure messaging channel is setup on the top of this TLS by the session keys generated during EAC protocols and the eID Service transmits the user data over this channel to the SP.

5. On successful authentication, the SP grants access for the service to the user.

![Figure 3.3: Authentication in German eID system.](image)

### 3.3.2 Revocation

The German eID card offers pseudonymous authentication by using SP specific ID which do not have a global recognition. But this makes revocation challenging [42]. So a SP specific revocation list is derived from the global revocation list. When a card is stolen or lost, the user can initiate revocation by informing the police or the eID helpline. Then the SP specific revocation ID is accessed and revoked by the revocation Service.

### 3.3.3 Privacy Analysis

The German eID system is the most privacy friendly system discussed so far in this document. It provides:

1. Pseudonymity: The user’s real name is not revealed to the SPs. Instead SPs receive a pseudonym for the user. Each SP receives a different pseudonym for the same user.

2. Data minimization: Excess data sharing is avoided. For instance, to verify if the age of the user is above 18, only a yes/no is sent instead of the age.

3. Transparency: eID card releases data only with user consent to an authenticated and authorized service.
Chapter 3. Existing eID systems

4. Unobservability: The user data is sent through a channel which is protected against eavesdropping and tampering.

However the scheme has the following issues:

- The security of eID authentication relies on the tamper-resistance of the smart card chips. If an attacker manages to extract the CA2 key from any eID card, then this attacker would be able to forge arbitrary identities. There would be no way for eID servers to recognize spoofed cards. Revoking the compromised CA2 key would render all affected cards useless for eID purposes, requiring their replacement [1]. The chips produced within a period of three months get the same key pair for the chip authentication [43]. In the first year, since the introduction of German eID system, 8 million eID cards were issued. This means that if one chip is compromised, approximately 2 million cards would need to be revoked [44].

- The user data that is transmitted after the selective disclosure by the user does not contain any signature in order to verify if it is indeed the data that was originally issued by BSI and sent by a legitimate eID cardholder [1]. Only the context of the EAC protocols run and the secure channel thus established assure the eID-Server of the authenticity of the eID data. Though this is a feature of the German eID system, it is also a limitation as outside this context, there is no way to verify the origin of eID data and thus a compromised card cannot be detected.

- The eID-Server is recommended to be implemented by the SP but can also be implemented by third parties. eID server implemented by third parties create a privacy risk as it serves more than one SP and can learn the visiting patterns of users as the attributes that are revealed by the users are identifying. Until 2012, eID servers implemented by only two third parties were being used by the SPs [45].

3.4 Dutch eID system using polymorphic pseudonyms

The Dutch eID system is a federated eID infrastructure which can be used by public and private services. The objective of the Dutch eID system is as follows [46]:

- To provide a more secure electronic service for public and private organizations than existing systems [47][48].
- To provide a fall-back option to public and private organisations should the primary login facility fail.
- To reduce the number of tokens that users need to remember (username-passwords) or have (smartcards or mobile tokens) such that one token of user’s choice can be used for online authentication.
- To fulfil the need for eID tokens which can be recognized at European level.

Some variations of the Dutch eID system have been proposed during the planning stage. In the following, the version with polymorphic pseudonyms [49] will be discussed as it is the most privacy friendly version among those proposed. The role of IDP, in the system, will be taken up by private sector organizations. In addition to IDPs and SPs, the system also includes the following parties:
• **A Pseudonym Provider (PP):** Generates polymorphic pseudonym when the IDP sends a unique identifier, known as *U-id*, which is both specific for the IDP and the user.

• **Brokers:** Mediate user authentication between the various eID parties, essentially IDPs and the many SPs. Each SP can choose the Broker it wants to work with.

• **Attribute Providers (ATP) and Authorization Providers:** Supplement attributes and provide authorization when certain critical user data is requested by SP.

• **A Key Management Authority (KMA):** Generates and manages the cryptographic keys used by the parties to create, transform and use the pseudonyms within the scheme.

• **Central Information Point e-ID Investigations (CIPEI):** Investigates criminal abuse of the e-ID infrastructure pseudonymity features. CIPEI can handle two kinds of requests:
  - De-pseudonymization: Identity of the person behind the pseudonym is requested by the law enforcement by providing the SP domain along with the pseudonym.
  - Pseudonymization: Pseudonym used by a person at a SP is requested by the law enforcement agency by providing the *U-id* of a user and a reference to a SP.

The pseudonymization in this system, which involves three levels of pseudonyms, namely, (1) polymorphic pseudonyms (2) encrypted pseudonyms and (3) pseudonyms, has been designed to fulfill the following requirements:

• SP gets a user specific pseudonym from the IDP, but the pseudonym is independent of the IDP.

• Each SP gets a different pseudonym for the same user.

The enrolment of users in the Dutch eID system is performed by the IDPs. The IDP verifies the user’s identity and generates a *U-id* which is then mapped to an Elliptic Curve and sent to the PP. The PP generates a polymorphic pseudonym using the IDP’s public key received from the KMA, *U-id* and ElGamal encryption (See 2.1.2) before sending it to the IDP. The revocation can be performed through a centralized authority or at the IDPs through an offline method. The rules for revocation have been established for the eID scheme though the method has not been decided yet.

### 3.4.1 Authentication

The online authentication process is described as follows (Fig. 3.4):

1. The user requests a resource from SP.
2. The SP sends an authentication request to the broker along with the list of attributes to be obtained from ATP.
3. The Broker displays the list of IDPs for the user to choose.\footnote{It is assumed that the user possesses an authentication token of at least one IDP in the list.}

---

1. It is assumed that the user possesses an authentication token of at least one IDP in the list.
4. On selection, the user is redirected to the IDP.
5. The user is authenticated at the IDP and the IDP deduces the $U\text{-}id$.
6. The IDP transforms the polymorphic pseudonym into an encrypted pseudonym using a service provider specific ID. This transformation is performed by the IDP without the knowledge of the pseudonym using the homomorphic property of El-Gamal encryption. The IDP sends the encrypted pseudonym to the Broker.
7. If the SP had requested for additional attributes, the Broker obtains them from the ATP.
8. The SP sends the attributes along with the encrypted pseudonym to the SP. If extra attributes are not requested, just the encrypted pseudonym is sent.
9. The SP uses its own ElGamal private key to decrypt the encrypted pseudonym and obtain the pseudonym for the user. If the SP is satisfied by the authentication, it provides the user with access to the resource requested.

The Dutch eID system uses other Key Derivation Function (KDF) keys such as pseudonymization key, domain transformation key and encryption transformation key in order to form the pseudonym. Details on the formation of the pseudonym can be found in [49].

### 3.4.2 Privacy Analysis

The Dutch eID system with polymorphic pseudonyms has the following privacy properties:

1. Pseudonimity: The user’s real name is not revealed to the SPs.
2. Data minimization: The SP requests for the attributes of a user that it requires.
3. Transparency: The SP receives the requested attributes after user gives consent.

4. Unlinkability: Each SP gets a different pseudonym for the same user thus preventing the possibility of colluding SPs linking the user’s activities.

5. Unobservability: The encrypted pseudonym is randomized such that the Broker cannot identify the data of the same user sent on multiple occasions.

In spite of efforts made to provide privacy to the users, this system has its negatives which are explained as follows:

- The IDP is a privacy hotspot. The IDP knows which SPs a user visits and how often. This information might be considered very sensitive in some cases. The IDP does not need to know this information in order to perform its role. An alternative proposed in [49] is to store the polymorphic pseudonyms in a smart card, thus restricting the user’s choice of tokens. This alternative uses the Chip Authentication Version 3 protocol as in the German eID system.

- In the case of storing polymorphic pseudonyms in the smart card, the data is still sent through the IDP and the system relies on the IDP to not store the data for future use.

- The existence of law enforcement access introduces the risk of misuse. Unlike GOV.UK Verify (Section 3.2), the Dutch eID system provides a structure for the law enforcement access. For CIPEI to fulfil its role, it requires the cooperation of KMA. If CIPEI as well as KMA are run by the same organization, then there is a privacy risk due to the possibility of collusion.

- The Dutch eID system is complex and it is possible that in the future SPs might outsource the decryption of encrypted pseudonyms to a third party. In that case the third party can learn the user’s visiting patterns.

- Finally, the system does not mention anything about the transition from the existing authentication procedure to the proposed scheme. If the SPs are allowed keep their databases, then pseudonymity will be a farce as all that the SP will need to do is to add a pseudonym to an existing customer. The user then has no privacy but instead merely a new identifier.

3.5 I Reveal My Attributes (IRMA)

The systems discussed so far, except the German eID system, use the approach of identifying entities by a unique number. This approach is convenient for bookkeeping but it is not privacy friendly. Unique identifiers can be used to trace the user and her/his activities both online and offline. But most use cases do not require identification but only authentication and/or authorisation. For example, when a person boards a train, the system only needs to know whether or not this person is allowed to do so and not who the person is [50].

An alternate approach is possible by using attribute-based credentials (ABC) where the SP is provided with the required attributes instead of identity information. The credential issuer or the IDP issues credentials to the user and vouches for the validity of the attributes contained in the credential. After issuing the credentials, the IDP cannot
recognise the data credential as it is signed using blind signatures [51]. This eliminates
the possibility of the issuer tracking the card owner. The user can use the credentials to
prove the possession of an attribute to the SP. Two important technologies that make use
of an ABC approach are Microsoft’s U-Prove [52][53] and IBM’s Identity Mixer (Idemix)
[54]. Idemix is built upon the concept of Camenisch-Lysyankaya signature scheme and
its protocols [55].

![Figure 3.5: A credential with four attributes.](image)

Unlike the other eID systems discussed in this document, the IRMA [56] project is not
developed by any government. It has been developed by Radboud University of Nijmegen
in the Netherlands as a partial implementation of Idemix that demonstrates the appli-
cability of ABCs on smart cards [50]. The implementation includes privacy enhancing
features of ABCs such as selective disclosure of attributes using zero-knowledge protocols
[57][58][59][60]. The main idea behind the IRMA card is that the information stored on
the card can be read digitally from the card only if the cardholder gives consent to read
a specified set of attributes. Attributes can either be identifying or non-identifying. For
example, the attributes such as name, address, Social Security Number are identifying
attributes which uniquely identify a person while attributes such as student, age-above-
18 are non-identifying attributes as they can belong to a large group of people. A set
of attributes related to a particular activity is grouped into a cryptographic container
known as Credential. Fig. 3.5 shows a typical credential in IRMA with four attributes
and a secret key of the user signed by the issuer. The issuer’s signature provides au-
thenticity and integrity and the presence of the user’s secret key makes the credential
non-transferable.

Like all eID systems, IRMA has IDPs and SPs. In addition it requires a Scheme
Authority (SA), which is the highest authority in the IRMA system as it governs the
identity management scheme and is responsible to maintain the trust and value of the
scheme [44]. It also initializes the smart cards before they are issued to users by the
IDPs.

### 3.5.1 Authentication

The online authentication process is described as follows (Fig. 3.6):

1. The user requests a resource from the SP.
2. The SP requests for a set of attributes. The user can check if the requested at-
tributes are reasonable and absolutely necessary before cooperating to start the
IRMA procedure of proving the presence of those attributes.
3. If the user does not have those attributes in its credentials, it can request the IDP to issue credentials with the required attributes. The IDP initiates a commitment phase in which the user’s card generates a commitment using its private key. Then the IDP verifies the validity of the commitment to authenticate the user.

4. The IDP issues the credentials to the user by signing the attributes using Camenisch Lysyanskaya (CL) signature scheme [61].

5. The user provides the necessary attributes to the SP through Selective disclosure and Zero-knowledge proof.

Steps 3 and 4 are required for credential issuance only. If the user already holds the credentials, these two steps need not be performed.

### 3.5.2 Revocation

The current implementation does not have a revocation mechanism. Instead the credentials come with an expiry date and need to be re-issued. If a card is lost or stolen, then the system hopes that the data is not misused till the expiry of credentials. This is not ideal if IRMA is to be extended into a large scale project. A revocation scheme for IRMA has been recently proposed which involves a semi-trusted party, a Revocation Authority (RA), which is responsible for revoking the credentials and keeps track of the revocation values of revoked credentials [62]. In this approach, time is split into epochs and the RA chooses generators per-epoch and per-verifier.

### 3.5.3 Privacy Analysis

IRMA has the following privacy properties:

1. Issuer unlinkability: Issuance involves creating a blind signature which conceals the resulting credential from the IDP. Thus any information gathered by the IDP during issuing cannot be used to link a verification of the credential to its issuance as the IDP is not involved when the user provides proof of possession of a credential.

2. Multi-show unlinkability: The user is guaranteed that when a credential is verified multiple times, these sessions cannot be linked by the SP.
3. Transparency and Data-minimization: By selective disclosure, the user can choose to reveal only a selection of the attributes contained in a credential.

4. Anonymity: The user does not need to reveal his/her identity and instead presents his/her credentials.

IRMA is certainly the most privacy friendly system discussed in this document. There still remains few issues that need to be addressed:

- If a card is lost or stolen, the lack of revocation procedure in the current implementation allows the possibility of misuse of credentials till it expires.
- Using the revocation procedure introduced in [62] the above issue can be addressed. But it introduces a new problem. The procedure uses a revocation value which is encoded by the credential such that the credential can be identified when it is revoked. Thus weakening the unlinkability argument.

### 3.6 Discussion

Five eID systems have been discussed in this chapter, among which the Belgian eID system is the least privacy-friendly and IRMA is the most privacy friendly. All systems, except the Belgian eID system (Section 3.1), have made attempts to safeguard the user’s privacy but have issues that need to be addressed. The German eID system (Section 3.3) and IRMA (Section 3.5) use a direct authentication infrastructure where the IDP is not involved in the user authentication process. The IDP issues credentials which are stored on the user’s smartcard and the user authenticates itself to the SP when required. Both these schemes also require user consent before data is sent to the SP. The German eID largely relies on the security of the smartcard. An attacker has a massive incentive to spend resources on a single card as the private key is shared with possibly more than a million other cards. Both systems suffer when it comes to revocation.

GOV.UK Verify (Section 3.2) and the Dutch eID system (Section 3.4) are federated systems which use a mediator to prevent SP and IDP to collude. The latter provides unobservability while in the former the hub can be used for surveillance. In addition, GOV.UK Verify introduces a MS which handles domain-specific user data. Both systems suffer from privacy hotspots in the form of IDP which stores user information. Both systems rely on the security of the IDP if multiple tokens are to be used. It is possible to improve these systems by using homomorphic encryption which will allow the IDP to process user data in encrypted form. Another possibility is to decentralize the role played by IDP using block chain. These possibilities will be explored in Chapter 4.
Chapter 4

Privacy Enhancing Solutions

In the previous chapter a number of privacy issues in some of the existing/proposed eID systems were identified. In this chapter, possible solutions to address the privacy issues are discussed. First, the IDP in federated eID systems may perform computation on encrypted data using homomorphic encryption. Second, the issue of privacy hotspots may be eliminated by decentralizing the role of IDP using block chain. But as seen in Chapter 2, block chain has other privacy issues which need to be addressed in order to use its decentralized infrastructure for private data.

First, a brief description of three Somewhat Homomorphic cryptosystems is provided. Second, Zerocash, which is an extension of Bitcoin protocol, is described followed by the description of Enigma, a decentralized computation platform.

4.1 Somewhat Homomorphic Encryption (SHE)

Can we do arbitrary computations on data while it remains encrypted, without ever decrypting it? Rivest, Adleman and Dertouzos [30] posed this question in 1978. Fully Homomorphic encryption (FHE), that permits arbitrarily complex computation on encrypted data, is the answer to it. But it took researchers more than three decades to come up with a FHE scheme. In Chapter 2, schemes which have either additive homomorphic property or multiplicative homomorphic property were discussed. The first scheme that went beyond simple additive or multiplicative homomorphisms was proposed by Boneh, Goh and Nissim [63]. Their scheme supports arbitrary number of additions and a single multiplication.

The first FHE proposed by Gentry [64] included three major stages - Somewhat Homomorphic Encryption (SHE), Squashing and Bootstrapping. SHE is limited to evaluating low-degree polynomials over encrypted data because after a certain amount of computations too much error accumulates such that the decryption leads to a wrong value. Squashing modifies the decryption circuit of the original SHE scheme to make it bootstrappable while bootstrapping refreshes the ciphertext by homomorphically applying the decryption procedure and obtaining a new ciphertext that encrypts the same value as before but has lower noise.

Though Gentry’s scheme shows that FHE is theoretically possible, it is computationally expensive and not practical. Many variants of the scheme [65][66][67] that have been proposed to improve the efficiency are not impractical either. But the SHE proposed in these schemes are computationally cheaper and can be used in practical applications. In this Section, first, the SHE using ideal lattices proposed by Gentry, then the SHE
using integers and finally, the SHE based on Learning with Errors (LWE) are described in 4.1.1, 4.1.2 and 4.1.3 respectively.

4.1.1 SHE using Ideal lattices

The SHE proposed by Gentry [64][68] is based on ideal lattices. Ideal lattices were chosen as the complexity of the decryption algorithms in lattice based encryption schemes are very low compared to schemes which rely on exponentiation such as RSA and ElGamal. Before explaining the SHE scheme, ideal lattices are described briefly.

Ideal lattices

A lattice \( \mathcal{L} \) is a discrete subgroup of \( \mathbb{R}^m \), represented by all the integer linear combinations of \( n \) linearly independent vectors \( \{ \vec{b}_1, \ldots, \vec{b}_n \} \in \mathbb{R}^n \) where \( m \geq n \). The sequence of vectors \( B = \{ \vec{b}_1, \ldots, \vec{b}_n \} \) is called a lattice basis. A lattice may have many different bases.

Let \( R = \mathbb{Z}[x]/f(x) \) be the ring of integer polynomials modulo some monic polynomial \( f(x) \in \mathbb{Z}[x] \) of degree \( n \). An ideal \( I \) is a subset of a ring \( R \) and has the following properties:

- Additively Closed: \( i_1, i_2 \in I \rightarrow i_1 + i_2 \in I \)
- Closed under multiplication with \( R \): \( i \in I, r \in R \rightarrow i \cdot r \in I \)

Since \( R \), which is isomorphic to \( \mathbb{Z}^n \), is an additive group and ideals in \( R \) are by definition subgroups, they correspond to lattices. Lattices of this form are called ideal lattices.

Key Generation

To generate the keys,

- Take as input a ring \( R \) and basis \( B_I \) of \( I \).
- Generate two bases \( B_{pk}^J \) and \( B_{sk}^J \) using an algorithm \texttt{IdealGen}(\( R, B_I \)) of some ideal \( J \), such that \( I + J = R \), that is, \( I \) and \( J \) are relatively prime. A good basis which has nearly orthogonal and short vectors is chosen as \( B_{sk}^J \) while the Hermite Normal Form (HNF) of \( B_{pk}^J \) is used as \( B_{pk}^J \). The HNF of a lattice is an upper triangular basis that is efficiently computable from any other basis of the lattice [69].

Public key \( pk = (R, B_I, B_{pk}^J, \texttt{Samp}) \) and private key \( sk = B_{sk}^J \).

Encryption

- Let \( m \) be the message where \( m \in \{0, 1\} \) as implemented in [70] and \( \vec{m} = (m, 0, \ldots, 0) \).
- Sample a short vector from the coset \( \vec{m} + I \) using the algorithm \texttt{Samp} such that \( \vec{c} \leftarrow \texttt{Samp}(\vec{m}, B_I, R, B_{pk}^J) \).
- The ciphertext \( \vec{c} = \vec{c} \mod B_{pk}^J \)

The encryption is analogous to noise addition, that is the message is masked by noise. Geometrically the ciphertext is close to a \( J \)-point and the encrypted message \( m \) is mapped to a lattice point by a correctable error vector.
Decryption
The decryption is analogous to error correction such that the message is recovered as the closest lattice point such that the plaintext message is:

\[ \vec{m} = (\vec{c} \mod B_{sk}^J) \mod B_I \]

where \( \vec{c} \mod B_{sk}^J = \vec{c} - B_{sk}^J \cdot \lfloor (B_{sk}^J)^{-1} \cdot \vec{c} \rfloor \)

Evaluate
In order to check if the homomorphic property for this scheme holds, let \( \vec{c} = \vec{v} + \vec{c}' \) and \( \vec{c}' = \vec{m} + 2\vec{r} \) where \( \vec{v} \) is a vector in the lattice \( L \) and \( \vec{r} \) is a short integer vector. Then,

\[
\begin{align*}
\vec{c}_1 + \vec{c}_2 &= (m_1 + v_1 + 2r_1) + (m_2 + v_2 + 2r_2) \\
&= (m_1 + m_2) + (v_1 + v_2) + 2(r_1 + r_2) \quad (4.1)
\end{align*}
\]

and

\[
\begin{align*}
\vec{c}_1 \times \vec{c}_2 &= (m_1 + v_1 + 2r_1) \times (m_2 + v_2 + 2r_2) \\
&= (m_1 \times m_2) + v_1 \times (m_2 + v_2 + 2r_2) + v_2 \times (m_1 + 2r_1) + \\
&\quad 2(r_1 \times r_2 + m_1 \times r_2 + r_1 \times m_2) \quad (4.2)
\end{align*}
\]

As \((v_1, v_2)\) are in the lattice \( L \), the scheme is homomorphic if \( \vec{c}' + \vec{c}'' \) and \( \vec{c}'_1 \times \vec{c}'_2 \) are small enough such that taking \((\mod 2)\) will give \((m_1 \oplus m_2, 0, \ldots, 0)\) and \((m_1 \times m_2, 0, \ldots, 0)\) respectively.

4.1.2 SHE using Integers

The SHE proposed in [65], based on modular arithmetic, uses addition and multiplication over the integers rather than working with ideal lattices over a polynomial ring. The security of the scheme is based on the hardness of finding the approximate integer greatest common divisors (GCD), that is, find an integer \( p \) when a list of integers that are near-multiples of \( p \) are given [71]. In spite of being a simple scheme, it is inefficient due to the need for large parameters to fulfil the security requirements.

Parameters
Let \( \lambda \) be the security parameter, \( \gamma \) be the bit-length of integers in the public key, \( \eta \) be the bit-length of the secret key, \( \rho \) be the bit-length of the noise (the distance between the public key elements and the nearest multiples of the secret key) and \( \tau \) be the number of integers in the public key. For a scheme with complexity \( \tilde{O}(\lambda^{10}) \), the parameter sizes suggested by the authors were \( \rho = \lambda, \rho' = 2\lambda, \eta = \tilde{O}(\lambda^2), \gamma = \tilde{O}(\lambda^3) \) and \( \tau = \eta + \lambda \), where \( \rho' \) is the secondary noise parameter.
Chapter 4. Privacy Enhancing Solutions

Key Generation

To generate the keys,

- Select an odd integer $p$ of bit-length $\eta$ to be the secret key.
- Sample $x_i$ from $D_{\eta,\rho}(p)$ for $i = 0, \cdots, \tau$, where
  \[
  D_{\eta,\rho}(p) = \{ \text{choose } q \leftarrow \mathbb{Z} \cap [0, 2\eta/p), r \leftarrow \mathbb{Z} \cap (-2^{\rho'}, 2^{\rho'}) : \text{output } x = pq + r \}\]
- Relabel so that $x_0$ is the largest. Restart unless $x_0$ is odd and $r_p(x_0)$ is even, where
  \[
  r_p(x_0) = x_0 - \lfloor x_0/p \rfloor \cdot p.
  \]

Public key $pk = (x_0, \cdots, x_\tau)$ and private key $sk = p$.

Encryption

If $m \in \{0, 1\}$ is the message, then to encrypt $m$,

- Select a random subset $S \subset \{1, 2, \cdots, \tau\}$ and a random integer $r \in (-2^{\rho'}, 2^{\rho'})$
- The ciphertext is $c = [m + 2r + 2 \sum_{i \in S} x_i] \mod x_0 \in \mathbb{Z}$.

Encryption can be viewed as adding $m$ to a random subset sum of “encryptions of zeros” as the noise is even.

Decryption

The message can be obtained as $m = (c \mod p) \mod 2$ which is essentially $m = (c \mod 2) \oplus (\lfloor c/p \rfloor \mod 2)$ as $p$ is odd.

Evaluate

Given a binary circuit $C$ with $t$ inputs and $t$ ciphertexts $c_i$, on applying the integer addition and multiplication gates of $C$ on the ciphertexts, the resulting output integer is obtained. It must be noted that the noise grows as addition and multiplication are performed on the ciphertext. The noise grows exponentially on multiplication and at a certain point decryption becomes hopeless.

If $d$ is the degree of the multivariate polynomial $f$ evaluated by an integer circuit $C'$, then the noise grows exponentially with degree while the ciphertext size grows linearly. The encryption scheme will provide the correct decryption as long as the noise term is smaller than $p/2$. In terms of the degree of the polynomial,

\[
d \leq \eta - 4 - \log_2 |f| \quad \frac{\rho'}{\rho'} + 2
\]

(4.3)
4.1.3 SHE based on Ring-Learning with Errors

The SHE based on Learning with errors (LWE) relies on the hardness of standard problems in lattices [66] while the SHE based on ring-LWE relies on ideal lattices [72]. The LWE problem introduced in [73] are known to be as hard as the worst-case lattice problems, which are believed to be exponentially hard even against quantum computers. The LWE problem asks if a secret $\vec{s} \in \mathbb{Z}_n^q$ can be recovered given a sequence of ‘approximate’ random linear equations on $\vec{s}$, each correct up to some small additive error (say, $\pm 1$). In this document the SHE based on ring-LWE will be described. Before explaining the SHE scheme, the LWE problem as well as the ring-LWE are formally defined.

**LWE Problem**

The LWE problem can be defined as follows: for a size parameter $n \geq 1$, a modulus $q \geq 2$, and an ‘error’ probability distribution $\chi$ on $\mathbb{Z}_q$, let $A_{s,\chi}$ be the probability distribution obtained by choosing a vector $\vec{a} \in \mathbb{Z}_q^n$ uniformly at random and a noise term $e \in \chi$ and outputting $(\vec{a}, \langle \vec{a}, \vec{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where additions are performed in $\mathbb{Z}_q$. An algorithm solves LWE with modulus $q$ and error distribution $\chi$ if, for any $\vec{s} \in \mathbb{Z}_n^q$, given an arbitrary number of independent samples from $A_{s,\chi}$ it outputs $\vec{s}$ with high probability. LWE can be viewed as a random bounded distance decoding problem on lattices. A special case $q = 2$ corresponds to the well-known learning parity with noise (LPN) problem, which is a central problem in coding theory (decoding from random linear codes).

**Ring-LWE problem**

Cryptosystem based on LWE has a disadvantage of very large public and private key sizes. To improve the efficiency of LWE, Lyubashevsky, Peikert and Regev [74] defined ring-LWE, which is the LWE problem in a wide class of rings. The ring-LWE Problem is to find $s$, given polynomially many $(a_i, b_i) \in R \times R$ with $b_i = a_i \cdot s + e_i$ where the $a_i$’s are uniformly random in $R$, $s$ is a secret ring element in $R$, and the $e_i$’s are drawn from a random distribution over $R$.

**Key generation**

To generate the keys,

- Select a secret $s$ from the error distribution $\chi$ over the ring $R_q = \mathbb{Z}_q[x]/(f(x))$ and define a secret vector $\vec{s} = (1, s, s^2, \ldots, s^D) \in R_q^{D+1}$, where $q$ is a prime number and $D$ is the maximal ciphertext length allowed. The polynomial $f(x)$ is of degree $n = 2^{\log(\lambda)-1}$, where $\lambda$ is the security parameter.
- Select $a \in R_q$ and $e \in \chi$ randomly. Also select $t \in \mathbb{Z}_q^*$. 
- Calculate $b = as + te$.

Public key $pk = (a, b)$ and private key $sk = s$. In addition, $n, f, q$ and $\chi$ are made public.
Chapter 4. Privacy Enhancing Solutions

Encryption

- Let \( m \) be the message in the message space \( R_t = \mathbb{Z}_t[x]/\langle f(x) \rangle \).
- Compute \( c_0 = b + m \in R_t \) and \( c_1 = -a \).
- The ciphertext \( \vec{c} = (c_0, c_1) \).

While the encryption algorithm only generates ciphertexts \( \vec{c} \in R^2_t \), homomorphic operations might add more elements to the ciphertext.

Decryption

- The decryption circuit can handle \( D \) ciphertexts at most. The inner product of \( \vec{c} \) and \( \vec{s} \) is taken over \( R^{D+1}_q \).

\[
\langle \vec{c}, \vec{s} \rangle = \sum_{i=0}^{D} c_i s_i \in R_q
\]

where \( \vec{c} = (c_0, c_1, \ldots, c_D) \).
- Message can be obtained as \( m = \langle \vec{c}, \vec{s} \rangle \mod t \)

Evaluate

- Given two ciphertexts \( \vec{c} = (c_0, c_1, \ldots, c_d) \) and \( \vec{c}' = (c'_0, c'_1, \ldots, c'_d) \) of the same length \( d \leq D \). If the lengths are unequal, zeros are padded. Addition of the ciphertexts is \( \vec{c}_{add} = (c_0 + c'_0, c_1 + c'_1, \ldots, c_d + c'_d) \in R^{D+1}_q \). The length of the ciphertext does not increase.
- Given two ciphertexts \( \vec{c} = (c_0, c_1, \ldots, c_d) \) and \( \vec{c}' = (c'_0, c'_1, \ldots, c'_d) \) such that \( d + d' \leq D \), the product of the ciphertexts is

\[
\left( \sum_{i=0}^{d} c_i v^i \right) \cdot \left( \sum_{i=0}^{d'} c'_i v^i \right) \equiv \left( \sum_{i=0}^{d+d'} \hat{c}_i v^i \right)
\]

where \( v \) is a symbolic variable and the output ciphertext is \( \vec{c}_{mult} = (\hat{c}_0, \hat{c}_1, \ldots, \hat{c}_{d+d'}) \)

If the two ciphertexts \( \vec{c} \) and \( \vec{c}' \) are the encryption of \( m \in R_t \) and \( m' \in R_t \), then \( \vec{c}_{add} \) and \( \vec{c}_{mult} \) decrypt to \( m + m' \) and \( m \cdot m' \) respectively.

4.2 Zerocash

Block chain, described in Chapter 2, was introduced in [27] to prevent double spending through consensus. An ideal digital transaction has three characteristics - decentralized, secure and private. Block chain is decentralized and secure but it does not provide privacy to the users [31][75][76]. On the other hand, E-cash [51][77] uses blind signatures to make transactions untraceable and secure but it is not decentralized. Another alternative for untraceable transactions is the use of laundries which mix different users’ funds and shuffle them [78]. Here the users need to trust the laundry not to record how the mixing was done and to return the funds after mixing. This system relies on a central party, is not completely private and not secure enough as was seen from the Mt.Gox incident [79].
Zerocoin [80] extends Bitcoin by adding a decentralized laundry. Zerocoin, as such, is a digital commitment to a random serial number which has a value once added to the block chain. To spend a zerocoin, the sender reveals the serial number by non-interactive Zero-Knowledge proof that the serial number is from a zerocoin in the block chain. The serial number is marked as spent and the recipient receives a random bitcoin from the escrow pool. But Zerocoin has limitations. It requires an explicit laundry process which has a poor performance. The algorithm adds cost to the Bitcoin network and reveals the payment destination.

To counter the limitations, Zerocoin has been further extended to Zerocash [81]. Zerocash utilizes succinct non-interactive zero-knowledge arguments of knowledge (zk-SNARKs) [82][83][84]. zk-SNARKS used in this system requires an initial setup of a public key (about 1GB) by a trusted party using a random trapdoor which must be destroyed. This public key is required to spend transactions. If the trapdoor is not destroyed, the trusted party could forge transactions. The Zerocash can be described in a simplified manner through two main processes - minting and spending. To perform minting and spending, Zerocash requires the following:

- **Coin Commitment** \( c = \text{Comm}(v, \rho, a_{pk}) \)
  where \( \rho \) is a random number used to generate a serial number, \( a_{pk} = H(a_{sk}, 0) \) is the public key corresponding to the secret key address \( a_{sk} \) which controls the coin and \( H() \) is a cryptographic hash function.
- **Serial Number** \( sn = \text{PRF}(a_{sk}, \rho) \)
  where \( \text{PRF} \) is a Pseudo-Random Function.
- The miner maintains the Merkle Tree (2.2.3) of all previous commitments.

![Figure 4.1: Zerocash [85].](image)

Minting is the process of converting the bitcoin of value \( v \) to a zerocash coin of value \( c \). For spending, the sender uses the following protocol (Fig. 4.1):

- Create a coin of value \( c \) using \( a_{pk} \) of the recipient.
• Send coin secrets \((v, \rho, r', r'')\) to the recipient out of band or using the recipient’s public key. Note that \(sn\) and \(a_{sk}\) are not known to the sender. The recipient can check that the sender has not double spent by using \(sn\) and zk-SNARK.

Zerocash enables users to pay one another directly without revealing the origin, the destination, and the amount. It allows transactions of variable denomination and is more efficient than Zerocoin.

### 4.3 Enigma

Enigma is a peer-to-peer network, enabling different parties to jointly store and run computations on data while keeping the data private [86]. It uses an on-chain, the block chain, and an off-chain called Enigma, a private chain of nodes. The data is split between the different nodes such that no single node has access to the entire data. Data is sufficiently randomized across the nodes and replicated to ensure high availability. The off-chain network solves the following issues:

• *Storage:* Enigma has a decentralized offchain distributed hash-table (DHT) that is accessible through the blockchain, which stores references to the data but not the data itself.

• *Private-enforcing Computation:* Private data can be processed without leaking the data to the nodes.

• *Heavy Processing:* The off-chain can perform complex calculations and allow the block chain to scale as the transactions are not replicated by every node.

![Figure 4.2: Enigma [87].](image-url)
The computational model of Enigma is based on Secure Multi Party Computation (MPC), which allows computation in a distributed way, without a trusted third party. Data from different nodes is used for computation without leaking information to other nodes and the result is guaranteed through a verifiable secret-sharing scheme [88]. The blockchain is utilized as the controller of the network, manages access control, identities and serves as a tamper-proof log of events.

Consider the following to understand how this infrastructure could be used (Fig. 4.2)[87]: A user installs an application on his/her phone and when he/she signs up for the first time, a new shared (user, service) identity is generated and sent, along with the associated permissions, to the block chain. The data collected from the phone is encrypted using a shared encryption key and sent to the block chain, which subsequently routes it to an off-chain, while retaining only a pointer to the data on the public ledger.

4.4 Discussion

Somewhat homomorphic encryption provides a way for computing a reasonable number of operations on encrypted data without being computationally infeasible. The first SHE proposed using ideal lattices discussed in 4.1.1, is computationally expensive and relies on non-standard security assumptions. The SHE using integers in 4.1.2 is simple as it uses integers. Its security is based on the approximate-GCD assumption. In spite of the simplicity, it is inefficient due to the need for large key sizes $O(\lambda^5)$. The SHE based on ring-LWE in 4.1.3 was discussed. It is efficient and has smaller ciphertexts compared to other schemes.

In Section 4.2, Zerocash, an extension to the Bitcoin protocol, was discussed. It provides a way to improve the privacy of users on the block chain by anonymizing the sender, receiver as well as the data. It has been designed for crypto-currencies but it may be possible to adopt it for the purpose of identity management as well. Finally, Enigma, a decentralized computation platform which allows storage and computation of private data, has been discussed in 4.3. It uses off-chains in addition to the block chain. A similar approach has been used in [89] but it requires a minimally trusted manager.
Chapter 5

Discussion and Future Work

In this report the necessity for the development of eID systems and the need to safeguard the privacy of users in these systems has been stated. Then, an overview of the existing eID systems - Belgian eID system, GOV.UK Verify, German eID system, Dutch eID system using polymorphic pseudonyms and IRMA - along with the authentication mechanism has been provided. The privacy of users in direct and federated eID systems in terms of anonymity, pseudonimity, data-minimization, unlikability, unobservability and transparency has been analysed.

In order to address the privacy issues encountered in the existing eID systems, two possible solutions - homomorphic encryption and block chain - have been put forth. Homomorphic encryption will allow parties to process data without being able to read it while block chain can decentralize the role played by central parties such as IDP. Partially homomorphic cryptosystems do not allow arbitrary computation on encrypted data while fully homomorphic cryptosystems allow arbitrary computation on encrypted data but are practically infeasible. Somewhat homomorphic cryptosystems provide a trade-off between computation and the practicality. It allows considerable number of computations to be performed efficiently.

Block chain provides a decentralized infrastructure which provides integrity but is not suitable to handle private data. In order to utilize the decentralized infrastructure for private data processing, variation of block chain - Zerocash and Enigma - have been briefly discussed.

Further, the focus of the research will be on improving the privacy of users at the IDP in a federated eID system through cryptographic solutions. Henceforth in the research it is assumed that the user is in possession of a cryptographic token, such as a smart card, from an IDP. From this literature survey, the following research questions have been formulated:

- Can somewhat homomorphic encryption provide a scalable and efficient solution to improve the privacy of users at the IDP by preventing the IDP from knowing which user it is authenticating to an SP; which SP’s service has been requested by a user; hiding both the SP and user from the IDP?
- How feasible and scalable is it to decentralize the role of IDP using block chain and what topology of private block chain used to store private data minimizes the latency?
Bibliography


