Identity-Based Encryption

Group 15

November 20, 2007

Contents

Abstract

Public-Key Certification methods were introduced to provide users with confidence in the authenticity of the public keys they were using, while mitigating the threat of man-in-the-middle attacks. Public-Key Infrastructures (PKIs) were designed as mechanisms to manage certificates but they turned out to be heavy to deploy and cumbersome to use. In order to bypass problems associated with conventional PKIs, Shamir introduced in 1984 the concept of identity-based cryptography where a public key can be any binary string identifying its owner non-ambiguously. The motivation of this kind of scheme was to simplify key management and remove the need of public-key certificates as much as possible. Several practical solutions for identity based signatures (IBS) have been devised since 1984 but finding a practical identity based encryption scheme (IBE) remained an open challenge until 2001 when Cocks and Boneh-Franklin independently proposed schemes using quadratic residues and bilinear maps respectively. Since then, a lot of work has been done to extend the viability and functionality of Boneh-Franklin's scheme. In this report, we survey Cocks' and Boneh-Franklin's schemes in detail and also review other schemes based on Boneh-Franklin which try to demonstrate a stronger security model. We conclude the report with some applications of IBE and the open problems for the practicability of IBE.

Keywords: Identity-based Encryption, Private Key Generator, Quadratic Residues, Bilinear Maps, Random Oracle

Chapter 1

Introduction

Identity-Based Encryption as an idea was first proposed in a seminal paper by Adi Shamir in 1985. The paper introduced a novel idea of a cryptographic scheme, which enabled any pair of users to communicate securely and to verify each other's signatures without the need to exchange private or public keys, without keeping key directories and without using the services of a third party [Sha85]. Since then numerous schemes have been proposed which use the idea of identity-based encryption.

1.1 Motivation

Identity-Based Encryption (IBE) is defined as a public-key encryption scheme where any valid string, which uniquely identifies a user, is the public key of the user.

The original motivation for identity-based cryptography was to simplify certificate management and thus eliminate the need for Certification Authorities. In traditional Public-Key Infrastructure (PKI), a public-key certificate is required to bind the key to its user. However, certificates are not required in IBE because each user has a unique identity to which they are intrinsically bound. Instead, IBE requires a trusted central authority called a Private-Key Generator (PKG) for generation and distribution of private keys to registered users.

Thus, IBE removes several difficulties associated with traditional PKI such as certificate lookup, lifecycle management, Certificate Revocation Lists and cross-certification issues giving a greatly-simplified public-key encryption and signature scheme.

The differences between Traditional PKI and IBE are summarized in Table 1.1.

1.2 Working of IBE

A simplistic model of the way in which IBE functions is shown in Figure 1.1.

Table 1.1: Differences between traditional PKI and IBE

Features	Certificate based PKI	ID based PKI
Private key generation	By user or CA	By PKG
Key certification	Yes	No
Key distribution	Requires an integrity protected channel for distributing a new public key from a user to his CA	Requires an integrity and privacy protected channel for distributing a new private key from the TA to its owner
Public key retrieval	From public directory or key owner	On-the-fly based on owner's identifier
Escrow facility	No (Unless Public key generation is by CA)	Yes

Secure communication takes place in three steps as follows:

1. **Step 1:** Alice encrypts the email using Bob's e-mail address, `bob@b.com`, as the public key.
2. **Step 2:** When Bob receives the message, he contacts the key server. The key server contacts a directory or other external authentication source to authenticate Bob's identity and establish any other policy elements.
3. **Step 3:** After authenticating Bob, the key server then returns his private key, with which Bob can decrypt the message. This private key can be used to decrypt all future messages received by Bob.

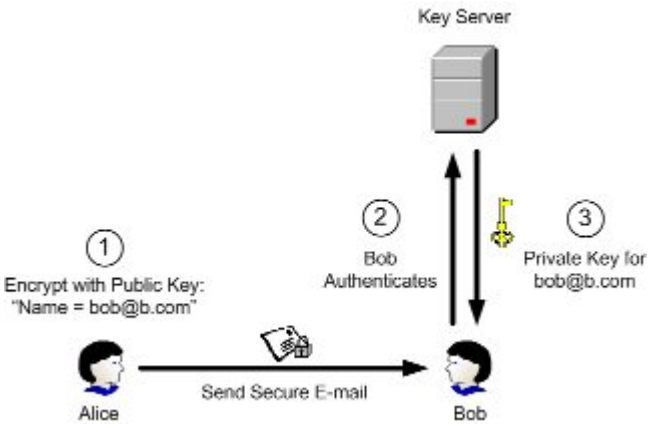

Figure 1.1: Identity-based Encryption

1.3 A Generic IBE scheme

After Shamir first proposed the idea of IBE in 1985, several researchers including Shamir himself, published schemes for Identity-Based Signature (IBS) schemes through the 90's. However, a viable scheme for IBE remained an open problem until 2001 when the first papers were independently published by Cocks [Coc01] and Boneh and Franklin [BF01]. Both these schemes and several other schemes which improve upon them are essentially made up of four algorithms as follows [BF01]:

- **Setup:** generates system parameters $params$ which are made public to all users in the system and a master-key sk_{PKG} which is known only to the PKG.
- **Extract:** takes as input $params$, sk_{PKG} and any arbitrary ID and returns a private key sk_{ID} corresponding to that ID.
- **Encrypt:** takes as input $params$, the ID of the receiver and a plaintext message M and returns a ciphertext C.
- **Decrypt:** takes as input $params$, the private key sk_{ID} issued by the PKG and the ciphertext C and returns the plaintext message M.

We will study Cocks' scheme in detail (Chapter 2) followed by Boneh-Franklin's scheme (Chapter 3).

Chapter 2

Cocks' Method of Quadratic Residues

Cocks' identity-based cryptosystem uses quadratic residues modulo a large composite integer [Coc01]. The security of this scheme is related to the difficulty of solving the quadratic residuosity problem [Coc01]. Understanding this scheme requires some background in number theory, which we describe in section 2.1. We explain the actual scheme in Section 2.2 and then discuss its various practical and security aspects in later sections.

2.1 Mathematical Background

2.1.1 Quadratic Residues

Definition 1 *[MvOV97] Let $a \in \mathbb{Z}_p^*$. Then a is said to be a quadratic residue modulo p if there exists an $x \in \mathbb{Z}_p^*$ such that $x^2 \equiv a \pmod{p}$, else it is a quadratic non-residue modulo p. Then, Q_p denotes the set of all quadratic residues modulo p and $\overline{Q_p}$ denotes the set of all quadratic non-residues modulo p.*

2.1.2 Legendre Symbol

Definition 2 *[MvOV97] Let p is an odd prime and a be an integer. The Legendre symbol $\left(\frac{a}{p}\right)$ is defined to be,*

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p|a, \\ 1, & \text{if } a \in Q_p, \\ -1, & \text{if } a \in \overline{Q_p} \end{cases}$$

2.1.3 Jacobi Symbol

Definition 3 *[MvOV97] Let $n \geq 3$ be odd with prime factorization $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$. Then the Jacobi symbol $\left(\frac{a}{n}\right)$ is defined to be,*

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}$$

Some of the properties of the Jacobi symbol are instrumental in the design of Cocks' algorithm. These are enumerated next.

Properties of Jacobi Symbol

1. $\left(\frac{a}{n}\right)$=0,1 or -1. Moreover, $\left(\frac{a}{n}\right) = 0$ iff gcd(a,n)$\neq 1$.
2. $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$. Hence if $a \in \mathbb{Z}_n^*$, then $\left(\frac{a^2}{n}\right) = 1$.
3. $\left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right)\left(\frac{a}{n}\right)$.
4. $\left(\frac{1}{n}\right) = 1$.
5. $\left(\frac{-1}{n}\right) = (-1)^{(n-1)/2}$. Hence $\left(\frac{-1}{n}\right) = -1$ if $n \equiv 3 \pmod 4$.

2.2 Cocks' IBE algorithm

Cocks' IBE algorithm can also be broken down into the four distinct phases described in 1.3 [Kim]. These are:

Setup:

The PKG chooses two primes $p, q \equiv 3 \pmod 4$. Here p and q are private parameters which are known only to PKG. Next, it computes $n = p \cdot q$ and makes the value of n public to all users. The system also makes use of a universally available secure hash function, $H : \{0,1\}^* \rightarrow J_n$ which maps any arbitrary string to J_n, the set of integers in $\mathbb{Z}_n^*$ whose Jacobi symbol is 1 [Coc01].

Extract:

Whenever, a user, say Alice, contacts the PKG for her private key, the PKG extracts this key from knowledge of the user's identity and its privately-known parameters p and q.

Knowing Alice's identity, say ID_A, the PKG first computes $H(ID_A) = a$ such that the Jacobi symbol $\left(\frac{a}{n}\right)$ is $+1$. From the properties of the Jacobi symbol and given an odd n and an $a = 2^e a_1$ (where the exponent e is some integer), it is possible to efficiently calculate the Jacobi symbol without knowing the prime factorization $n = p \cdot q$ [MvOV97]. This algorithm has a running time of $O((lg\ n)^2)$ bit operations. Also, since the hash function is public, this process of finding an $H(ID_A) = a$ which satisfies $\left(\frac{a}{n}\right) = +1$ can be replicated by any user [Coc01].

However, unlike the Legendre symbol, the Jacobi symbol $\left(\frac{a}{n}\right)$ does not reveal whether or not a is a quadratic residue modulo n [MvOV97]. This requires further deductions as described next.

By definition from 2.1.3, $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$. Since $\left(\frac{a}{n}\right) = +1$, there are only two cases possible:

- $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = +1$
  Thus a is a quadratic residue modulo both p and q. This means that a is also a quadratic residue modulo n.

- $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$
  Now $\left(\frac{-a}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{-1}{p}\right) = (-1)(-1) = +1$
  $\Rightarrow -a \in Q_p$. Similarly, $-a \in Q_q$
  This means that $-a$ is also a quadratic residue modulo n.

Thus, either a or $-a$ is a quadratic residue modulo n. However, deduction of this fact requires knowledge of the private parameters p and q.

Next, the PKG computes the private key of Alice. One way to do this is to calculate $r = a^{\frac{(n+5)-(p+q)}{8}} \pmod{n}$ [Coc01]. Such an r will satisfy either $r^2 \equiv a \pmod{n}$ or $r^2 \equiv -a \pmod{n}$ depending on which of a or $-a$ is a square modulo n. An algorithm is described in [MvOV97] to calculate square roots modulo a composite n, with an expected running time of $O((lg\ n)^3)$ bit operations.

Encrypt:

Suppose another user, say Bob, wants to send an encrypted message to Alice. The encryption is done bit-by-bit.

Each bit x_i of m-bit plaintext message string $M = \{x_1 \cdots x_m\}$ is encoded as either +1 or -1 [Coc01]. Bob first chooses values t_1, t_2 at random modulo n, such that $t_1 \neq t_2$ and $\left(\frac{t_1}{n}\right) = \left(\frac{t_2}{n}\right) = x_i$. The ciphertext $\langle s_{i,1}, s_{i,2} \rangle$ corresponding to the plaintext bit $\langle x_i \rangle$ is then computed as $s_{i,1} \equiv (t_1 + a/t_1)\ (mod\ n)$ and $s_{i,2} \equiv (t_2 - a/t_2)\ (mod\ n)$ [Gag03].

Here, we observe that the ciphertext for one message bit consists of two data elements. This doubling of ciphertext data occurs because, even though Bob can calculate $H(ID_A) = a$, he is not sure whether Alice has a received a root of a or $-a$ [Coc01]. Hence, if Bob makes the assumption that $r^2 \equiv a \pmod{n}$, and sends only $\langle s_{i,1} \rangle$ the decryption will fail with a probability of $\frac{1}{2}$.

Decrypt:

Let's assume that Alice received a private-key r satisfying $r^2 \equiv a \pmod{n}$. Then Alice recovers the plaintext bit $\langle x_i \rangle$ by computing $(s_{i,1} + 2r) \pmod{n}$ which can be simplified as:

$$\begin{aligned}
(s_{i,1} + 2r) \pmod{n} &= (t_1 + a/t_1 + 2r) \pmod{n} \\
&= (t_1 + r^2/t_1 + 2r) \pmod{n} \\
&= (t_1 + r)(1 + r/t_1) \pmod{n} \\
&= t_1(1 + r/t_1)(1 + r/t_1) \pmod{n} \\
&= t_1(1 + r/t_1)^2 \pmod{n}
\end{aligned}$$

Then the Jacobi symbol $\left(\frac{s_{i,1}+2r}{n}\right)$ evaluates to:

$$\left(\frac{s_{i,1}+2r}{n}\right) = \left(\frac{t_1(1+r/t_1)^2}{n}\right) = \left(\frac{t_1}{n}\right)\left(\frac{(1+r/t_1)}{n}\right)^2 = \left(\frac{t_1}{n}\right) = x_i$$

Since, the Jacobi symbol can only take values +1, -1 or 0, the decryption will fail if and only if $\left(\frac{(1+r/t_1)}{n}\right) = 0$. From **Property 1** in 2.1.3, this is possible if and only if $gcd((1+r/t_1), n) \neq 1 \Rightarrow \left(1+\frac{r}{t}\right) \notin \mathbb{Z}_n^*$, i.e. there is at least one prime factor common to both $\left(1+\frac{r}{t}\right)$ and n. However, for fairly large primes p and q, the probability of such an event happening is quite low.

For better understanding, a numerical example is given in the Appendix.

2.3 Practical Aspects

Instead of encrypting and decrypting entire messages with this scheme, a more practical approach is to generate a transport key and use it to encrypt the data using symmetric encryption techniques. IBE can be employed to ensure secure transmission of the transport key.

The most computationally demanding part of the encryption process involving an m-bit transport key is computing m Jacobi symbols and m divisions modulo n. The decryption process involves computation of m Jacobi symbols. For typical key sizes, the scheme is computationally not too expensive [Coc01].

We observe from the algorithm that a single bit of the message gets mapped into two elements of the group $\mathbb{Z}_n^*$. This causes message inflation by a factor of $2log_2 n$ [Gag03]. This requires much more bandwidth than transmission of the plaintext message itself and may not be acceptable in some applications.

2.4 Security Analysis

The security of this scheme is based upon the hardness of the **quadratic residuosity problem** which can be stated as follows:

Definition 4 *[MvOV97] Given an odd composite integer n and $a \in J_n$ (where J_n =set of all $a \in \mathbb{Z}_n^*$ having Jacobi symbol +1), decide whether or not a is a quadratic residue modulo n.*

If n is a prime, then it is easy to decide whether $a \in \mathbb{Z}_n^*$ is a quadratic residue modulo n, since by definition, $a \in Q_n$ if and only if $\left(\frac{a}{n}\right) = +1$. However, if n is an odd composite integer it is difficult to decide whether it is a quadratic residue modulo n unless the factorization of n is known [MvOV97]. We can only guess the correct answer with a probability of $\frac{1}{2}$. To prove how the security of this scheme relates to the intractability of the quadratic residuosity problem, we consider a passive attack against the generation of each bit of the transport key. We show that if the attack is successful, it directly implies that the attacker has managed to crack the quadratic residuosity problem.

We assume that the hash function H is a random oracle. This assumption is required to prove that the IBE algorithm provides semantic security [Gol04]. Assume that there exists a procedure F that can recover $\langle x_i \rangle$ from $\langle s_{i,1}, s_{i,2} \rangle$ without knowing either r or factors of n. We also assume that this recovery procedure can only take the hashed identity a as the input and cannot make separate use of the input to the hash function H. Then, dropping the subscript notation and assuming, without loss of generality, only one ciphertext $\langle s \rangle$, the breaking process is the mapping:

$$F(n, a, s) \leftarrow x = \left(\frac{t}{n}\right)$$

valid whenever $s = (t + a/t) \pmod{n}$ for some t.

Consider the value of F evaluated for an a such that $\left(\frac{a}{n}\right) = +1$ but a is not a square modulo n. In this case, we can find three other values $t1$, $t2$, $t3$ which map to the same s [Gol04]. These are,

$$\begin{aligned} t1 &\equiv t \pmod{p} & t1 &\equiv a/t \pmod{q} \\ t2 &\equiv a/t \pmod{p} & t2 &\equiv t \pmod{q} \\ t3 &\equiv a/t \pmod{p} & t3 &\equiv a/t \pmod{q} \end{aligned}$$

But as $\left(\frac{a}{p}\right) = \left(\frac{a}{p}\right) = -1$, then $\left(\frac{t1}{n}\right) = \left(\frac{t2}{n}\right) = -\left(\frac{t}{n}\right) = -\left(\frac{t3}{n}\right)$ [Coc01]. Since, all the four values are equiprobable, the attacker can make a correct guess only with a probability of $\frac{1}{2}$.

Now, consider the second case where a is a square modulo n. In this case, since the value of t is unique, the attacker can correctly guess the value of x with a probability of $\frac{1}{2} + \epsilon$.

Thus, uncertainty is only associated with the first case where a is not a square modulo n. Hence, if the attacker makes a wrong guess, he is sure that a is not a square modulo n. This means that the attacker has managed to devises an algorithm which returns the correct solution of the quadratic residue problem with a probability greater than $\frac{1}{2}$. This goes against the intractability assumption of the quadratic residuosity problem. Hence, the algorithm is semantically secure.

In practice, we must also ensure that the generation of t for successive encryptions is done randomly, otherwise attacks are possible if the attacker can find some correlation.

The scheme as described above is still vulnerable to adaptive chosen ciphertext attacks. However, Cocks outlines an approach to defend against such attacks by adding redundancy to the transport key establishment data. However, no formal security proof for the scheme is available. Also, it is very easy to delete, add or modify bits in the encrypted message, so additional integrity protection must be employed to confirm the validity of the message.

Chapter 3

IBE Schemes using Bilinear Maps

Boneh-Franklin's scheme is built from bilinear maps and specifically uses the Weil pairing on elliptic curves as an example of such a map [BF01]. The scheme is secure as long as a variant of the Computational Diffie-Hellman problem called the Bilinear Diffie-Hellman assumption is hard.

We start the chapter with a brief introduction to bilinear maps. Then we present the Bilinear Diffie-Hellman assumption which used to prove the security of Boneh-Franklin's scheme. We explain Boneh-Franklin's scheme in detail along with the assumptions made in order to demonstrate its security. In later sections, we present improvements suggested on this scheme which demonstrate security even while some of the assumptions are relaxed and we also describe schemes which extend the functionality of Boneh-Franklin's scheme.

3.1 Introduction to Bilinear Maps

A bilinear map is a function which is linear in both its arguments. Here we consider maps that establish relationship between cryptographic groups. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic groups of order q, where q is some large prime. A *bilinear map* is a function $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ such that for all $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}$

$$e(au, bv) = e(bu, av) = e(u, v)^{ab}$$

These maps are sometimes called as pairings because they associate a pair of elements from group $\mathbb{G}_1$ to an element or a pair of elements in $\mathbb{G}_2$. Note that these maps can be degenerate, i.e. maps all pairs $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$. From the point of view of cryptography, we are interested only in admissible bilinear maps.

A *bilinear map* is said to be admissible if it satisfies the following properties:

1. **Bilinear:** We say that a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is *bilinear* if $\hat{e}(au, bv) = \hat{e}(bu, av) = \hat{e}(u, v)^{uv}$ for all $u, v \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$.

2. **Non-degenerate:** The map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$, i.e. $\hat{e}(u, v) \neq 1 \; ; \forall \; u, v \in \mathbb{G}_1$.

3. **Computable:** There is an efficient algorithm to compute $\hat{e}(u, v)$ for any $u, v \in \mathbb{G}_1$.

The *admissible bilinear map* is denoted by $\hat{e}$. In practice, $\mathbb{G}_1$ is implemented using the set of all points on certain elliptic curve which form an additive group and $\mathbb{G}_2$ is a multiplicative group of large prime order.

3.2 The Bilinear Diffie-Hellman assumption

Computational Diffie-Hellman Assumption (CDH): [Wat05]

Consider a cyclic group $\mathbb{G}$ of order q formed by the set of all points on an elliptic curve. The CDH assumption states that given $\langle g, ag, bg \rangle$ for a randomly chosen generator g and random $a, b \in \{0, 1, \cdots, q-1\}$, it is *computationally intractable* to compute the value $\langle abg \rangle$. The assumption can be well explained with Figure 3.1.

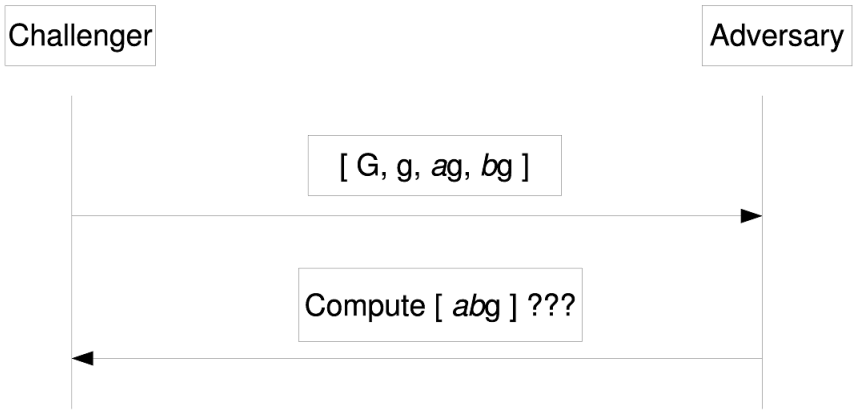

Figure 3.1: Computational Diffie-Hellman Assumption

The challenger challenges the adversary by giving $\langle \mathbb{G}, g, ag, bg \rangle$ parameters. By CDH assumption, it is is computationally infeasible for the adversary to compute $\langle abg \rangle$.

Bilinear Diffie-Hellman Assumption: [Wat05]

Let $\mathbb{G}_1$ be an additive group of prime order p formed by the set of all points on an elliptic curve and $\mathbb{G}_2$ be a multiplicative group of prime order q. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map and let P be a generator of $\mathbb{G}_1$. Then the Bilinear Diffie-Hellman (BDH) Assumption in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ can be stated as follows: Given $\langle P, aP, bP, cP \rangle$ for some $a, b, c \in \mathbb{Z}_q^*$, it is *computationally infeasible* to compute $W = \hat{e}(P, P)^{abc} \in \mathbb{G}_2$ if the CDH problem is intractable. The assumption can be well explained with Figure 3.2.

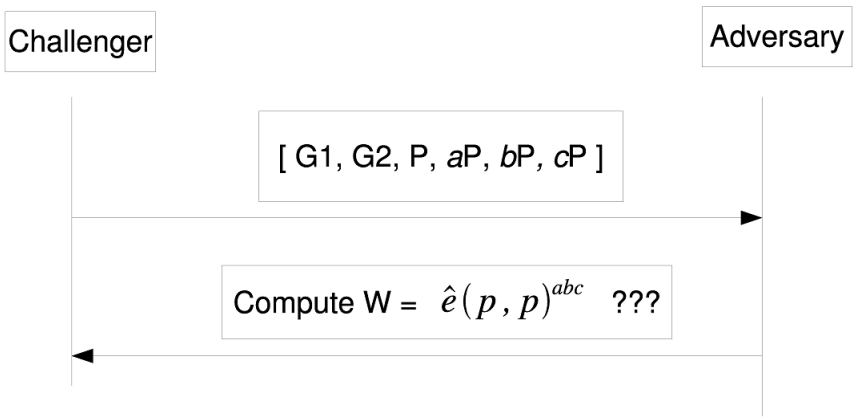

Figure 3.2: Bilinear Diffie-Hellman Assumption

The challenger challenges the adversary by giving $\langle \mathbb{G}_1, \mathbb{G}_2, P, aP, bP, cP \rangle$ parameters. By BDH assumption, it is is computationally infeasible for the adversary to calculate $W = \hat{e}(P, P)^{abc} \in \mathbb{G}_2$. We observe that given $\langle P, aP \rangle$, we can compute $\hat{e}(P, aP) = \hat{e}(P, P)^a$. We can also compute $\hat{e}(P, P)$ which is the generator of the multiplicative group $\mathbb{G}_2$. However, now we face the familiar discrete-log problem of computing a given $\hat{e}(P, P)$ and $\hat{e}(P, P)^a$.

3.3 Boneh-Franklin's IBE scheme using Bilinear Maps

The first complete and efficient IBE scheme was proposed by Boneh and Franklin [BF01] using a bilinear map called Weil pairing over elliptic curves. The construction of this bilinear map is beyond the scope of this report. Interested readers are referred to [BF01] for more details. We present a brief introduction of the pairing technique used.

The elliptic curve group (the set of point collection on elliptic curves) is used as $\mathbb{G}_1$ and the multiplicative group of a finite field is used as $\mathbb{G}_2$. The bilinear map transforms a pair of elements in group $\mathbb{G}_1$ and sends it to an element in group $\mathbb{G}_2$ in a way that satisfies **Bilinearity**. That is, it should be linear in each entry of the pair.

Assume that P and Q are two elements (e.g. points on elliptic curves) of an additive group $\mathbb{G}_1$. Let $\hat{e}(P, Q)$ be the element of a multiplicative group $\mathbb{G}_2$ which is a result of the pairing applied to P and Q. Then the pairing must have the following property:

$$\hat{e}(rP, Q) = \hat{e}(P, Q)^r = \hat{e}(P, rQ)$$

where r is an integer and rP denotes the element generated by r times of additions on P, e.g. $2P = P + P$, $3P = P + P + P$ and so on.

The proposed scheme consists of four randomized algorithms, viz. Setup, Extract, Encrypt and Decrypt. In the next section, we review each in detail.

3.3.1 Basic IBE Scheme [BF01]

1. **Setup:** Let E be an elliptic curve with coefficients in F_q, where q is some large prime. Let $\mathbb{G}_1$ be an additive group of order q formed by the set of all points on the curve E. Let P be a generator of $\mathbb{G}_1$. P is obviously a point on the curve E. Let $\mathbb{G}_2$ be a multiplicative group of order q. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map. Let l be some large prime which divides $(q^k - 1)$, here k is a ***SECURITY PARAMETER*** and $k \in \mathbb{Z}^+$.
   Choose a random $s \in [1, l] \in \mathbb{Z}_q^*$ as the ***MASTER SECRET***, compute $P_{pub} = sP$ as the public key of the PKG. Here PKG acts as Trusted Authority(TA) and s is known only to PKG.
   Choose crypto-hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \mathbb{G}_2^* \rightarrow \{0,1\}^n$ for some n
   $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2 \rangle$ are all public parameters.

2. **Extract:** For a given string $ID \in \{0,1\}^*$ of an authenticated user, compute $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$. Set private key of the user having identity ID to be $d_{ID} = sQ_{ID}$, so effectively each registered user has $\langle Q_{ID}, d_{ID} \rangle$ as public and private key pairs.

3. **Encrypt:** The transmitter selects a plaintext M and applies the pairing as shown in Figure 3.3 to generate a $TxPair = \hat{e}(sP, rQ_{ID}) \in \mathbb{G}_2$.

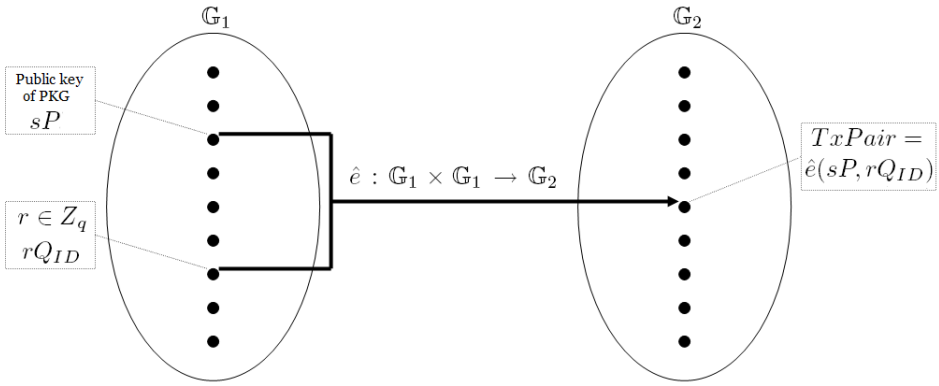

Figure 3.3: Pairing applied at the sender

   The ciphertext $C = \langle U, V \rangle = \langle\ rP,\ M \oplus H_2(TxPair)\ \rangle$ is sent to the receiver. Thus to encrypt a message M, sender uses the bilinear map to combine the identity of the receiver and public key of PKG and a random short term private key into a session key which masks the message.

4. **Decrypt:** The receiver after receiving the ciphertext $C = \langle U, V \rangle = \langle\ rP,\ M \oplus H_2(TxPair)\ \rangle$, applies the pairing as shown in Figure 3.4 to get $RxPair = \hat{e}(rP, sQ_{ID}) \in \mathbb{G}_2$.

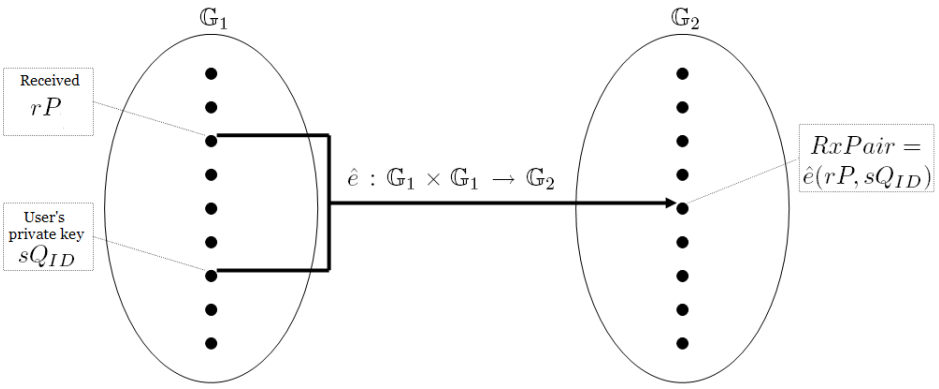

Figure 3.4: Pairing applied at the receiver

The receiver recovers the plaintext message M as follows:

$$\begin{aligned} M &= V \oplus H_2(RxPair) \\ &= M \oplus H_2(TxPair) \oplus H_2(RxPair) \\ &= M \end{aligned}$$

According to bilinearity property $TxPair = RxPair$. Thus the receiver successfully recreates the same session key and the short-term public key sent with the ciphertext for retrieval of the plaintext.

The security of proposed scheme is based on efficiently computable bilinear maps which are shown to be secure under a random oracle model. Random oracles are described in the Section 3.4.

However, this basic scheme provides security only against a chosen-plaintext attack. In [BF01], Boneh and Franklin further show that applying the Fujisaki-Okamoto generic transformation allows turning this basic scheme into a chosen-ciphertext secure one in an extended security model.

3.4 Random Oracle Model

A *Random Oracle* is an oracle, i.e. a theoretical black-box that responds to every query with a truly random response chosen uniformly from its output domain, except that when it receives any specific query it responds exactly the same way [BBM00].

The model can be better explained by Figure 3.5. Here the oracle is modeled as a mathematical function which gives Response(i) which is unique to Query(i). It never gives Response(j) for Quarry(i), where $i \neq j$.

Random oracles are typically used when no implementable solution exists. In the scheme proposed in [BF01], the two cryptographic hash functions H_1 and H_2 are modeled as random oracles for proving semantic security. The scheme

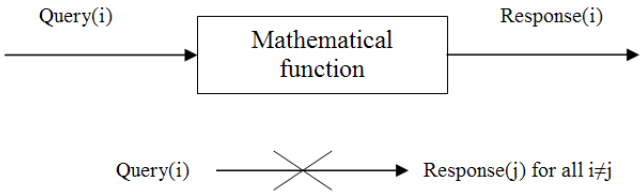

Figure 3.5: Random Oracle

is also proven to be secure against chosen ciphertext attack under the random oracle assumption. In the next section, we briefly explain an IBE scheme based on bilinear maps which is proven to be secure under the standard security model.

3.5 Efficient IBE without Random Oracles [Wat05]

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order p. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map. Let g be a generator in $\mathbb{G}_1$. If identities are allowed to be of arbitrary length, a collision resistant cryptographic hash function is used to reduce an identity to n bits, i.e. $H : \{0,1\}^* \rightarrow \{0,1\}^n$. The scheme contains four algorithms as explained below:

1. **Setup:** Choose $\alpha \in \mathbb{Z}_p$ at random. Compute $g_1 = g^\alpha$. Choose $g_2 \in \mathbb{G}_1$, compute **MASTER SECRET** $= g_2^\alpha$.
   Let $u^{'} \in \mathbb{G}_1$ be a random value and $U = (u_i)$ be a random n-length vector with every u_i chosen at random from $\mathbb{G}_1$. The parameters $\langle g, g_1, g_2, u^{'}, U \rangle$ are made public.

2. **Extract:** Consider a v bit string. Let v_i denote the i^{th} bit. Let $\Upsilon \subseteq \{1, 2, ..., n\}$ be the set of all i for which $v_i = 1$. Choose a random $r \in \mathbb{Z}_p$. Compute
   $$d_v = (g_2^\alpha (u^{'} \prod_{i \in \Upsilon} u_i), g^r) = (d_1, d_2)$$

3. **Encrypt:** In order to send a plaintext $M \in \mathbb{G}_1$, the sender chooses a random value $t \in \mathbb{Z}_p$ and computes the ciphertext C as
   $$\begin{aligned} C &= [C_1, C_2, C_3] \\ &= [\hat{e}(g_1, g_2)^t M,\ g^t,\ (u^{'} \prod_{i \in \Upsilon} u_i)^t] \end{aligned}$$

4. **Decrypt:** Once the receiver receives the ciphertext $C = [C_1, C_2, C_3]$, it decrypts the plaintext message M in the following manner.
   $$\begin{aligned} M &= C_1 \times \frac{\hat{e}(d_2, C_3)}{\hat{e}(d_1, C_2)} \\ &= \hat{e}(g_1, g_2)^t M \times \frac{\hat{e}(g^r, (u^{'} \prod_{i \in \Upsilon} u_i)^t)}{\hat{e}(g_2^\alpha (u^{'} \prod_{i \in \Upsilon} u_i)^r, g^t)} \\ &= \hat{e}(g_1, g_2)^t M \times \frac{\hat{e}(g, (u^{'} \prod_{i \in \Upsilon} u_i)^{rt})}{\hat{e}(g_1, g_2)^t . \hat{e}((u^{'} \prod_{i \in \Upsilon} u_i)^{rt}, g)} \\ &= M \end{aligned}$$

3.5.1 Efficiency of the Scheme:

If the value of $\hat{e}(g_1, g_2)$ is cached, then encryption requires $\frac{n}{2}$ *to* n group operations in $\mathbb{G}_1$, two exponentiations in $\mathbb{G}_1$, one exponentiation in $\mathbb{G}_2$ and one group operation in $\mathbb{G}_2$. For decryption it requires, two bilinear map computations, one group operation in $\mathbb{G}_2$ and one inversion in $\mathbb{G}_2$.

This scheme is computationally more expensive than [BF01] but provides security under the standard security model. However, this method requires a large amount of public data to be stored, which is undesirable for example, in smart card applications. Hence a more memory efficient scheme bearing the same amount of computational complexity is explained in the next section.

3.6 Secure and Practical IBE [Nac07]

This scheme is a variant of Water's IBE scheme with a much smaller size of system parameters. This scheme divides the system parameters size by a factor l at the cost of reducing security by l bits. The construction yields a fully secure practical IBE scheme.

The basic drawback of Water's scheme is that the public parameters contain $n+4$ group elements, where n is the size of the bit string representing identities. Since n can be the output of a hash function, the value of n must be at least 160. If we assume size of a group element to be 1024 bits, then each participant must store 164 kilobytes of public parameters. Here we will only present the variations as compared to Water's scheme and how they help to reduce the required memory size.

In Water's scheme, to encrypt a message for an identity $v = (v_1, ..., v_{n'}) \in \{0,1\}^{n'}$, we compute the product

$$u' . \prod_{v_i=1} u_i$$

where $U = (u_1, ..., u_{n'})$ is n'-dimensional public vectors.

Here we encode identities as n-dimensional vectors $v = (v_1, ..., v_n)$, where each v_i is an l-bit integer and $n.l = n'$ and compute the modified product

$$u' . \prod_{v_i=1}^{n} u_i^{v_i}$$

where $U = (u_1, ..., u_{n'})$ is an n-dimensional public vector. Therefore the size of the public vector U is reduced by a factor of $\frac{n'}{n} = l$.

3.6.1 Efficiency of the Scheme:

If the value of $\hat{e}(g_1, g_2)$ is precomputed then encryption requires one multi-exponentiation in $\mathbb{G}_2$ (n exponents of size l) and three exponentiations in $\mathbb{G}_1$.

Decryption requires two bilinear map computations, one group operation in $\mathbb{G}_2$ and one inversion in $\mathbb{G}_2$. Hence encryption and decryption are almost as efficient as in Water's scheme.

3.7 Hierarchical IBE scheme

A shortcoming of the Boneh-Franklin identity based encryption scheme is that in a large network, the PKGs key generation task rapidly becomes a bottleneck when many private keys have to be computed and secure channels have to be established to transmit them to their legitimate owner [LQ].

The burden on a single PKG could be reduced by having a hierarchy of PKGs where each PKG computes private keys only to the entities immediately below itself in the hierarchy. Such an IBE scheme is called Hierarchical Identity-Based Encryption (HIBE). Unlike normal IBE scheme where a string represents an identity, a tuple of strings now represents an identity in HIBE where each string contains his parents in the hierarchy [BNSNS04]. For example, in Figure 3.6 $\langle ID_1, ID_2 \rangle$ is the parent of $\langle ID_1, ID_2, ID_3 \rangle$.

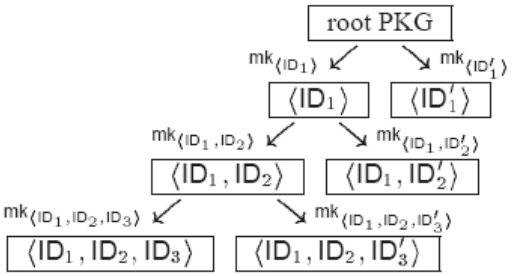

Figure 3.6: Hierachal of PKG's

Lower-level PKGs (i.e. PKGs other than the Root PKG located at the top of the hierarchy) generate private keys for their children by using some information coming from their ancestors together with a private information that is only known to them. Each of them then adds some information to the secret parameters of their children.

Gentry and Silverberg proposed a scheme that extends Boneh-Franklin scheme to obtain an HIBE scalable to an arbitrary number of levels [Gag03]. This scheme is identical to Boneh-Franklin scheme for the special case of one level of PKG, i.e. a single PKG and it also demonstrates security against adaptive chosen ciphertext attack in the random oracle model.

Chapter 4

Applications of IBE

4.1 Applications

We already mentioned that the original motivation for identity-based encryption was to simplify certificate management in Section 1.1. We now present other applications which are suited for IBE.

- **Revocation of public keys:**
  Public-key certificates are not valid beyond the expiration date. Once, the certificate expires, users have to contact the CA for a new valid certificate. However, in IBE, to ensure that private keys become invalid after a certain date, the PKG may use $receiver - identity \| current - date$ as the public key while generating the corresponding private key. Here $current - date$ can be the day, week, month or year depending on the frequency at which we want the users to renew their private key.This ensures that the private key will become invalid after that period.

- **Managing user credentials:**
  By encrypting the messages using the public-key $receiver-identity \| current-date \| clearance-level$ the receiver will be able to decrypt the message only if he has the required clearance. Thus, the PKG can be used to grant user credentials. To revoke a credential, the PKG simply stops providing the private key in the next time period.

- **Delegations of decryption keys:**
  Suppose a manager has several assistants each responsible for a different task. Then the manager can act as the PKG and give his assistants the private keys corresponding to their responsibilities (so the public key in this case would be $receiver - identity \| duty$). Each assistant can decrypt the messages whose subject fall within its responsibilities, but cannot decrypt messages intended for other assistants. The manager can decrypt all the messages using his master-key.

- **Forward secure encryption schemes:** [Gag03]
  Some of the schemes presented in this paper can also be used as building blocks to construct forward-secure encryption schemes and key-insulated

cryptosystems. In a forward-secure encryption scheme, the receivers private key evolves at each time period so that if the private key of a time period is compromised, all the messages encrypted in previous time periods are still secure. In a key-insulated encryption scheme, the secret key is divided into two parts, both evolving at every time interval, which must be combined to obtain the private decryption key. Future secret keys are compromised only if both parts are exposed in the same time period.

4.2 IBE in the Industry

Voltage Security (`http://www.voltage.com`) has secured several patents for their identity-based encryption products. The technology is called Voltage IBE (VIBE) and they have an entire suite of security products for secure mail (Voltage SecureMail), secure instant messaging and peer-to-peer communication (Voltage SecureIM) and secure file transmission and reception (Voltage SecureFile). A whitepaper (`http://www.voltage.com/pdf/VoltagePlatformTechOverview.pdf`) reveals that techniques for key management and distributions are quite similar to those described in 4.1.

Recently, Proofpoint, an email security products vendor, has added VIBE to their Secure Messaging appliance. Smartcard vendor Gemalto (formerly Gemplus) was the first to develop Smart IBE, a prototype to integrate identity-based encryption into smart cards.

Chapter 5

Observations and Conclusion

In this report, we have demonstrated that IBE definitely has some unassailable advantages over traditional PKI. To review its major characteristic, there is a strong binding between an identity and a key in IBE which especially benefits systems where there is a strong binding between user and the identifier of the communication end point, for example an email address, an IP address or a mobile phone number. Thus, IBE is a natural choice for applications such as secure email, secure SMS, mobile-commerce etc.

At the same time there are several open problems which need to be resolved before IBE can become the de facto standard for public-key cryptosystems.

The most inherent shortcoming of IBE is the Key Escrow Problem. This is because the PKG issues private keys for user using its master secret key. As a result, the PKG is able to decrypt or sign any messages. In terms of encryption, this property might be useful in some situations where user's privacy can possibly be limited. However, in terms of signature, this key escrow property is not desirable at all since non-repudiation is one of the essential requirement of digital signature schemes. It is interesting to note however that key escrow is somewhat restricted in HIBE: only a users father knows his/her complete private key (because of lower-level secret information) and the other PKGs are unable to decrypt ciphertexts intended for him/her.

Also in the algorithms discussed in this report, the crucial information is the PKGs master key. All the systems privacy is compromised if that master key is ever stolen by an attacker. In order to avoid having a single point of failure and remove the built-in key escrow, Boneh and Franklin showed in [BF01] that it was possible to split the PKG into several partial PKGs in such a way that these partial PKGs jointly generate a discrete logarithm key pair in a threshold fashion and each of them eventually holds a share of the master key. Users then have to visit a minimum of t-out-of-n honest PKGs to obtain a share of their private decryption key. These shares can then be recombined into a full decryption key using Lagrange interpolation.

However this approach has the drawback that different PKGs have to accept the same role of independently checking the users identity before delivering him/her a partial private key. This might be a burden in some situations and it hampers any centralized key issuing policy among the PKGs.

As for the algorithm themselves, Cocks scheme although not computationally-intensive requires much more bandwidth for the ciphertext than the input plaintext. Also, it never really caught on with researchers and has almost certainly been relegated to the sidelines in favour of Boneh-Franklin's scheme. The latter's original scheme demonstrated security only under the random oracle model and Boneh-Boyen's proposal which extended the security to the standard model was too computationally-intensive. Waters suggested a more efficient method, whose memory requirements were fine-tuned in a later paper by Naccache. However, all schemes based on pairing techniques are still quite expensive computationally. So ideally we would like to construct IBE schemes which are not based on the pairing but make more efficient use of bandwidth than Cocks' scheme. We do not know yet whether such schemes exist.

Bibliography

[BBM00] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. *Advances in Cryptology–Eurocrypt*, 2000, 2000.

[BF01] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *Advances in Cryptology-Crypto 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, 2001.

[BNSNS04] J. Baek, J. Newmarch, R. Safavi-Naini, and W. Susilo. A Survey of Identity-Based Cryptography. *AUUG 2004 Who Are You Identification and Authentication Issues in Computing*, pages 95–102, 2004.

[Coc01] C. Cocks. An identity based encryption scheme based on quadratic residues. *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 26–28, 2001.

[Gag03] M. Gagne. Identity-Based Encryption: A Survey. *RSA Laboratories Cryptobytes*, 6(1):10–19, 2003.

[Gol04] S. Goldwasser. 6.876 Course Lecture Notes scribed by Weis S., Massachussets Institute of Technology, 2004. `http://groups.csail.mit.edu/cis/crypto/classes/6.876/scribe/scribe-lec03.pdf`.

[Kim] J. Kim. A Survey of Identity-Based Encryption. `http://www.math.uiuc.edu/~duursma/Math595CR/KimJ.pdf`.

[LQ] B. Libert and J. Quisquater. What is possible with identity based cryptography for PKIs and what still must be improved. *Lecture notes in computer science*, pages 57–70.

[MvOV97] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[Nac07] D. Naccache. Secure and practical identity-based encryption. *Information Security, IET*, 1(2):59–64, 2007.

[Sha85] A. Shamir. Identity-based cryptosystems and signature schemes. *Proceedings of CRYPTO 84 on Advances in cryptology table of contents*, pages 47–53, 1985.

[Wat05] B. Waters. Efficient IdentityBased Encryption Without Random Oracles. *Advances in Cryptology-Eurocrypt 2005: 24th Annual International Conference on the Theory And Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, 2005.

APPENDIX

A Numerical Example of Cocks' IBE scheme

Setup: Let $p = 7$ and $q = 11$ such that $p, q \equiv 3 \pmod 4$

Therefore, $n = p \cdot q = 77$ and $\|\mathbb{Z}_n^*\| = 60$

$\mathbb{Z}_n^* = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 23, 24, 25, 26, 27, 29, 30, 31, 32, 34, 36, 37, 38, 39, 40, 41, 43, 45, 46, 47, 48, 50, 51, 52, 53, 54, 57, 58, 59, 60, 61, 62, 64, 65, 67, 68, 69, 71, 72, 73, 74, 75, 76$

$J_n = \left\{i \in \mathbb{Z}_n^* : \left(\frac{i}{n}\right) = +1\right\} = \{1, 4, 6, 9, 10, 13, 15, 16, 17, 19, 23, 24, 25, 36, 37, 40, 41, 52, 53, 54, 58, 60, 61, 62, 64, 67, 68, 71, 73, 76$

Extract: Consider an arbitrary ID such that $H(ID) = 4$.
PKG computes the private key for this ID as
$r = a^{\frac{(n+5)-(p+q)}{8}} \pmod n = 4^{\frac{(77+5)-(4+7)}{8}} \pmod n = 4^8 \pmod{77} = 9 \in \mathbb{Z}_n^*$
Here, $r^2 = 9^2 \equiv 4 \pmod n$
Consider plaintext message string $M = \{1, 0\}$ encoded as $\{+1, -1\}$

First bit, $x_1 = +1$

Encryption: Choose $t = 10$ since $\left(\frac{10}{77}\right) = +1$
Compute $s_1 \equiv (t + a/t) \pmod n = (10 + 4 \cdot 10^{-1} \pmod{77} = (10 + 4 \cdot 54 \pmod{77} = 72$
Send ciphertext $s_1 = 72$

Decryption: Compute $(s + 2r) \pmod n = (72 + 2 \cdot 9) \pmod{77} = 13$
Calculate Jacobi symbol $\left(\frac{s+2r}{n}\right) = \left(\frac{13}{77}\right) = +1 = x_1$
First bit decrypted!

Second bit, $x_2 = -1$

Encryption: Choose $t = 20$ since $\left(\frac{20}{77}\right) = -1$
Compute $s_2 \equiv (t + a/t) \pmod n = (20 + 4 \cdot 20^{-1} \pmod{77} = (20 + 4 \cdot 27 \pmod{77} = 51$
Send ciphertext $s_2 = 51$

Decryption: Compute $(s + 2r) \pmod n = (51 + 2 \cdot 9) \pmod{77} = 69$
Calculate Jacobi symbol $\left(\frac{s+2r}{n}\right) = \left(\frac{69}{77}\right) = -1 = x_2$
Second bit decrypted!

Failure condition: Consider the encryption-decryption of the second bit $x_2 = -1$. Choose $t = 12$ since $\left(\frac{12}{77}\right) = -1$ Compute $s_1 \equiv (t + a/t) \pmod{n} = (12 + 4 \cdot 12^{-1} \pmod{77} = (12 + 4 \cdot 45 \pmod{77} = 38$
Send ciphertext $s_2 = 38$

Decryption: Compute $(s + 2r) \pmod{n} = (38 + 2 \cdot 9) \pmod{77} = 56$
Calculate Jacobi symbol $\left(\frac{s+2r}{n}\right) = \left(\frac{56}{77}\right) = 0 \neq x_2$
Decryption failed for the second bit!