A Survey on ID-Based Cryptographic Primitives

M. Choudary Gorantla, Raju Gangishetti and Ashutosh Saxena

Institute for Development and Research in Banking Technology
Road No. 1, Castle Hills, Masab Tank, Hyderabad - 500057
Andhra Pradesh, INDIA.
{gmchoudary, graju}@mtech.idrbt.ac.in, asaxena@idrbt.ac.in

Abstract

ID-based cryptosystem has been, for a few years, the most active area of research and currently is of great interest to the cryptographic society. In this work we survey three fundamental ID-based cryptographic primitives *Digital Signature*, *Encryption* and *Key Agreement*, which are based on the mathematical concepts Integer Factorization, Quadratic Residues and Bilinear Pairings. We review several schemes along with their efficiency and security considerations. The survey helps in understanding the research work carried out in the area of ID-based cryptosystems from the year 1984 to 2004.

1 Introduction

The advent of E-Commerce demands for a secure communication of digital information. It has been proven for years that this can be achieved by cryptography. A set of cryptographic primitives used to provide information security services is generally referred to as a cryptosystem. The basic security services a cryptosystem should provide are Confidentiality, Integrity, Authentication, and Non-repudiation [50]. *Confidentiality* is keeping information secret from all other than those who are authorized to see it. *Integrity* is ensuring that the information has not been altered by unauthorized or unknown entities. *Authentication* is the assurance that the communicating party is the one that it claims to be. The corroboration of the identity of an entity is called *Entity Authentication* and corroborating the source of the information is called *Message Authentication*. *Non-repudiation* is preventing the denial of previous commitments or actions.

Confidentiality can be achieved by a cryptographic primitive called *Encryption*. It is defined as a function which maps an intelligible plaintext to an unintelligible ciphertext. *Digital Signature* is a fundamental cryptographic primitives which provides authentication, integrity and non-repudiation. The purpose of a digital signature is to provide a means for an entity to bind its identity to a piece of information. The process of signing entails transforming the message and some secret information held by the entity into a tag called digital signature. Besides encryption and digital signature, *Key Agreement* is another fundamental cryptographic primitive for establishing a secure communication. It is a process of computing a shared secret contributed by two or more entities such that no single entity can predetermine the resulting value. An authenticated key agreement is attained by combining the key agreement protocol with digital signatures. This avoids man-in-the-middle attack[51]. Symmetric key cryptosystems enable efficient encryption and some data integrity applications. Whereas asymmetric or Public Key Cryptosystems (PKC) enable efficient signature (particularly non-repudiation) and key management (which includes key agreement)[50].

In a traditional PKC, the association between a user's identity and his public key is obtained through a digital certificate issued by a Certifying Authority (CA). The CA checks the credentials of a user before issuing a certificate to him. If Alice wants to send a signed message to Bob, first she obtains a digital certificate for her public key from a CA. Alice then signs a message using her private key and sends the signed message along with her certificate to Bob. Bob first verifies the validity of the certificate by checking the certificate revocation list published by the CA, then he verifies the signature using public key in the certificate. If many CAs are involved between Alice and Bob the entire certificate path has to be verified. Hence, the process of certificate management requires high computational and storage efforts [38].

To simplify the certificate management process, Shamir [66] introduced the concept of ID-based cryptosystem in 1984. In such cryptosystems the public key of a user is derived from his identity information and his private key is generated by a trusted third party called Private Key Generator (PKG). The advantage of ID-based cryptosystems is that it simplifies the key management process which is a heavy burden in the traditional certificate based cryptosystems. In these cryptosystems Alice can send an encrypted message to Bob by using Bob's identity information even before Bob obtains his private key from the PKG. In the case of signature Bob can verify Alice's signature just by using her identity information. In general, an identity based cryptosystem has the following properties:

– user's public key is his identity (or derived from identity).

– no requirement of public key directories

– message encryption and signature verification processes require only receivers' and signers' identity respectively along with some system parameters (`params`)[1].

These properties make ID-based cryptosystems advantageous over the traditional PKCs, as key distribution is far simplified. It needs a directory only for authenticated public system parameters of the PKG, which is clearly less burdensome than maintaining a public key directory for total users. However, they suffer from an inherent drawback of key escrow i.e. PKG knows the users' private keys. They also require a secure channel for key issuance between PKG and user. The ID-based cryptosystems require the users to authenticate themselves to their PKG in the same way as they would authenticate themselves to a CA in traditional PKC.

Shamir[66], in his path breaking work, proposed an ID-based signature (IBS) scheme based on integer factorization problem. Later, satisfactory and practical solutions for IBS schemes were proposed in [27, 28]. In [37] Guillou and Quisquater proposed a "paradoxical" IBS using their interactive zero-knowledge protocol in [36, 37]. An IBS scheme using pairings was first proposed by Sakai, Ohgishi and Kasahara in [64], however they did not present the security analysis in their work. Paterson [55] proposed an IBS scheme based on pairings with brief security arguments but without rigorous proof. A provably secure IBS was proposed by Hess in [39], which is secure against existential forgery under adaptively chosen message and fixed ID attacks. In 2003, Cha-Cheon [18] proposed an IBS scheme based Gap Diffie-Hellman groups. They provided a definition of security for IBS schemes called security against existential forgery under adaptively chosen message and ID attacks and proved their scheme secure. An IBS scheme that enables secure batch verification was later proposed by Cheon, Kim and Yoon in [19]. This scheme is an adaptation of the signature scheme in [18]. An IBS scheme based Weil pairing and Quadratic Residues, which is equivalent to [18], was independently proposed by Yi in [73]. Chen, Zhang and Kim [21] proposed an IBS scheme without trusted PKG, eliminating the inherent *Key Escrow* problem.

From Identification to IBS. Fiat and Shamir [28] proposed a method of transforming identification schemes into efficient signature schemes. In Eurocrypt 2002, Abdalla, An, Bellare and Namprempre [1] proposed minimal conditions on the identification schemes to ensure security of the signature schemes in the random oracle model. They showed that a signature scheme is secure against chosen message attack in the random oracle model if and only if the underlying identification scheme is secure against impersonation under passive attacks. Dodis, Katz, Xu and Yung [24] defined a class of standard signature(SS) schemes that they call trapdoor, and then presented a random oracle using transform that returns any secure trapdoor SS scheme to secure IBS scheme. In Eurocrypt 2004, Bellare, Namprempre and Neven [6] presented a framework to provide security proofs for a large family of IBS schemes by considering the security against passive, active and concurrent attacks of underlying 'convertible' identification schemes. In their framework, they made use of 1). Fiat-Shamir transform [28] which turns a standard identification (SI) scheme [7] to SS scheme, 2). a transform that turns a convertible SI scheme into an identity based identification scheme and 3). another transform that turns an SS scheme to an IBS scheme. Using these transforms, they also devised new identity based signature schemes from earlier works [29, 53, 54] in the literature which describe only SI schemes.

Although there were many practical solutions proposed for ID-based signature schemes, the first practical ID-based encryption scheme was due to Boneh and Franklin [13] in 2001. Their encryption scheme is

[1] `params` typically include the public key of PKG and setup parameters calculated and published by PKG, which is a one time process.

indistinguishably secure against adaptively chosen ciphertext attacks i.e. IND-ID-CCA secure. In the same year Cocks [22] proposed another ID-based encryption scheme based on quadratic residues. However, there is no formal security proof given for the scheme and it is very inefficient in terms of bandwidth requirements.

The concept of hierarchical ID-based encryption scheme was first introduced by Horwitz and Lynn in [40]. It greatly reduces the workload on master servers (PKGs) and introduces key escrow at several levels. A secure and practical solution for hierarchical identity based encryption was later proposed by Gentry and Silverberg in [32]. A simple ID-based cryptography with mediated RSA was proposed by Ding and Tsudik in [23]. Sakai and Kasahara [63] proposed efficient method for a class of ID-based cryptosystems and ID-based cryptosystems with signatures and having multiple centers. An authenticated identity based encryption scheme that provides non-repudiation was proposed by Lynn in [47]. Canetti, Halevi and Katz [16] introduced a weaker model of security called "Selective-ID" model and proposed an ID-based encryption scheme without random oracles in that model. The same authors in [17] showed that any selective-ID, chosen plaintext ID-based encryption gives a chosen ciphertext secure public key system. Boneh and Boyen [10] proposed efficient ID-based encryption schemes that are secure in selective-ID model. The same authors again proposed another ID-based encryption scheme that is fully secure without random oracles in [11]. Recently, an efficient version of [11] was proposed by Waters in [70]. An ID-based encryption scheme with *Keyword Search* was proposed by Boneh et.al in [12]. A fuzzy ID-based encryption scheme, which allows for the encryption of data using biometric input as public key was proposed by Sahai and Waters in [62].

An ID-based authenticated key agreement protocol based on Weil pairing that makes use of the ideas of [13], [41] and [51] was proposed by Smart [68]. Scott [65] proposed another ID-based authenticated key agreement protocol based on Tate pairing. Chen and Kudla proposed an ID-based authenticated key agreement protocol that is efficient than [68]. They also are the first to suggest the concept of authenticated key agreement between members of separate domains i.e. key agreement between users under different PKGs. Shim [67] discussed a weakness in Smart's scheme [68] and proposed an ID-based authenticated key agreement protocol, which he claimed efficient and secure. However, Sun and Hsieh [69] showed that Shim's scheme is insecure against man-in-the-middle attacks. Later, McCullagh and Barreto [48] proposed efficient key agreement protocol with security proof in Bellare and Rogaway model [8], which can be instantiated in escrow and escrowless mode without imposing extra computational effort. But, Xie [71] pointed out a flaw and showed that an adversary can launch key compromise attack on this scheme. Choo [44] also demonstrated that McCullagh and Barreto's scheme and its 'fix' variant are not secure. Recently, Xie [72] proposed an ID-based authenticated key agreement scheme, secure in Bellare-Rogaway model [8], which is similar in construction to [48].

In this work we survey three fundamental ID-based cryptographic primitives *Encryption*, *Signature* and *Key Agreement* schemes. We review the schemes along with their efficiency and security considerations. The rest of the work is organized as follows: Section 2 gives the mathematical concepts and security models for the cryptographic primitives. Section 3 reviews ID-based signature schemes, Section 4 reviews ID-based encryption schemes and Section 5 gives ID-based authenticated key agreement protocols. We conclude our work in Section 6.

2 Background Concepts

In this section, we briefly present the background concepts which help in realizing the ID-based cryptosystems. This covers mathematical problems on *Integer Factorization*, *Quadratic Residues*, *Discrete Logarithm*, and *Bilinear Pairings* including *Diffie-Hellman Problem*.

2.1 Integer Factorization Problem

Definition 1: The Integer Factorization Problem (IFP) is defined as, given a positive integer n, find its factorization; i.e., write $n = q_1^{e_1} q_2^{e_2} ... q_k^{e_k}$ where the q_i are pairwise distinct primes and each $e_i \geq 1$.

2.2 Quadratic Residuosity Problem

Quadratic Residues. Let $a \in Z_n^*$ is said to be a quadratic residue modulo n, or a square modulo n, if there exists an $x \in Z_n^*$ such that $x^2 \equiv a \pmod{n}$. If no such x exists, then a is called a quadratic non-residue modulo n. The set of all quadratic residues modulo n is denoted by Q_n and the set of all non-residues is denoted by $\overline{Q}_n$.
The probability of any integer a to be a quadratic residue modulo n is approximately 1/2.
Legendre Symbol. Let q be an odd prime and a an integer. The Legendre symbol $\left(\frac{a}{q}\right)$ is defined to be

$$\left(\frac{a}{q}\right) = \begin{cases} 0 & \text{if } q|a; \\ 1 & \text{if } a \in Q_q; \\ -1 & \text{if } a \in \overline{Q}_q. \end{cases}$$

Jacobi Symbol. Let $n \geq 3$ be odd, with prime factorization $n = q_1^{e_1} q_2^{e_2} ... q_k^{e_k}$. Then the Jacobi symbol $\left(\frac{a}{n}\right)$ is defined as

$$\left(\frac{a}{n}\right) = \left(\frac{a}{q_1}\right)^{e_1} \left(\frac{a}{q_2}\right)^{e_2} ... \left(\frac{a}{q_k}\right)^{e_k}$$

If n is prime, then the Jacobi symbol is just the Legendre symbol.
Definition 2: The Quadratic Residuosity Problem (QRP) is defined as, given an odd composite integer n and $a \in J_n$, decide whether or not a is a quadratic residue modulo n. Here J_n is the set of all $a \in Z_n^*$ having the Jacobi symbol 1 for an odd $n \geq 3$.

Note that QRP reduces to IFP in polynomial time. If the factorization of n is unknown, then there is no efficient procedure known for solving QRP. It is believed that the QRP is as difficult as IFP, although no proof is known [50].

2.3 Discrete Logarithm Problem

Definition 3: Given a prime q, a generator $g \in Z_q^*$ and an element $b \in Z_q^*$, find an integer x, $0 \leq x \leq q-2$, such that $g^x \equiv b \pmod{q}$.
The DLP can be generalized to any cyclic group of finite order and is treated as computationally hard.

2.4 Diffie-Hellman Problem

Definition 4: The Diffie-Hellman Problem (DHP) is, given a prime q, a generator $g \in Z_q^*$, and elements $g^a \bmod q$ and $g^b \bmod q$ find $g^{ab} \bmod q$.
The DHP can be generalized to cyclic groups. DHP is also teated as computationally hard and reduces to DLP in polynomial time.

2.5 Bilinear Pairings

The bilinear pairings namely Weil pairing and Tate pairing of algebraic curves were used in cryptography for the MOV attack [49] and FR attack [30] respectively. These attacks reduce the DLP on some elliptic or hyperelliptic curves to the DLP in a finite field. Thus, their existence was thought to be a bad thing in cryptography. However, the situation has changed after Joux [41] gave a simple tripartite Diffie-Hellman protocol based on Weil pairing on supersingular curves. After this many elegant cryptographic schemes [13, 14, 39] have been devised exploiting the properties of these bilinear pairings.

Here we briefly give properties of a cryptographic bilinear map which is a modified Weil pairing [13]. Note that, unless specified, the notations in this subsection are the same for all the pairing based schemes in this work.

A cryptographic bilinear pairing is defined as $e : G_1 \times G_1 \rightarrow G_2$ where G_1 is an additive cyclic group of prime order q, G_2 is a multiplicative cyclic group of the same order and P is an arbitrary generator of G_1. An admissible bilinear pairing has the following properties:
Bilinear: $e(aR, bS) = e(R, S)^{ab}$ $\forall R, S \in G_1$ and $a, b \in Z_q^*$. This can be restated as $\forall R, S, T \in G_1$, $e(R + S, T) = e(R, T)e(S, T)$ and $e(R, S + T) = e(R, S)e(R, T)$.

Non-degenerate: There exists $R, S \in G_1$ such that $e(R,S) \neq I_{G_2}$ where I_{G_2} denotes the identity element of the group G_2.
Computable: There exists an efficient algorithm to compute $e(R,S)\ \forall R,S \in G_1$.
We refer to [13] for more comprehensive description on how these groups, pairings and other parameters are defined. Now, we present some mathematical problems in the domain of pairings, which form basis of security for some of the schemes given in this work.

2.5.1 Computational Diffie-Hellman Problem (CDHP)

Instance: (P, aP, bP) for some $a, b \in Z_q^*$.
Output: abP.

The success probability of any probabilistic, polynomial-time, 0/1-valued algorithm $\mathcal{A}$ in solving CDHP in G_1 is defined to be:

$$\mathtt{Succ}_{\mathcal{A},G_1}^{\mathtt{CDH}} = \Pr\left[\mathcal{A}(P, aP, bP, abP) = 1 : a, b \in Z_q^*\right]$$

`CDH Assumption:` For every probabilistic, polynomial-time, 0/1 - valued algorithm $\mathcal{A}$, $\mathtt{Succ}_{\mathcal{A},G_1}^{\mathtt{CDH}}$ is negligible.

2.5.2 Decisional Diffie-Hellman Problem(DDHP)

Instance: (P, aP, bP, cP) for some $a, b, c \in Z_q^*$.
Output: `yes` if $c = ab \bmod q$ and output `no` otherwise.

The DDHP in G_1 is easy as it can be solved in polynomial time by verifying $e(aP, bP) = e(P, cP)$. This is the well known MOV reduction [49]: The DLP in G_1 is no harder than the DLP in G_2.
The advantage of any probabilistic, polynomial-time, 0/1-valued algorithm $\mathcal{A}$ in solving DDHP in G_1 is defined to be:

$$\mathtt{Adv}_{\mathcal{A},G_2}^{\mathtt{DDH}} = |\Pr\left[\mathcal{A}(P, aP, bP, cP) = 1\right] - \Pr\left[\mathcal{A}(P, aP, bP, abP) = 1\right] : a, b, c \in_R Z_q^*|$$

`DDH Assumption:` For every probabilistic, polynomial-time, 0/1 - valued algorithm $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A},G_2}^{\mathtt{DDH}}$ is negligible.

`Gap Diifie-Hellman (GDH) Group:` A prime order group G_1 is a GDH group if there exists an efficient polynomial-time algorithm which solves the DDHP in G_1 and there is no probabilistic polynomial-time algorithm which solves CDHP with non-negligible probability of success. The domains of bilinear pairings provide examples of GDH groups. The MOV reduction provides a method to solve DDHP in G_1, whereas there is no known efficient algorithm for CDHP in G_1.

2.5.3 Weak Diffie-Hellman Problem(WDHP)

Instance: (P, S, aP) for some $S \in G_1$ and $a \in Z_q^*$.
Output: aS.
WDHP is no harder than CDHP.

2.5.4 Bilinear Diffie-Hellman Problem (BDHP)

Instance: (P, aP, bP, cP) for some $a, b, c \in Z_q^*$.
Output: $e(P, P)^{abc}$.

2.5.5 Decisional Bilinear Diffie-Hellman Problem (DBDHP)

Instance: (P, aP, bP, cP, r) for some $a, b, c \in_R Z_q^*$ and $r \in_R G_2$.
Output: `yes` if $r = e(P, P)^{abc}$ and output `no` otherwise.

`Decisional Modified BDHP:`
Instance: (P, aP, bP, cP, r) for some $a, b, c \in_R Z_q^*$ and $r \in_R G_2$.
Output: `yes` if $r = e(P, P)^{ab/c}$ and output `no` otherwise.

2.5.6 k-Decision Bilinear Diffie-Hellman Inversion (k-BDHI)

Instance: $\left(P, xP, x^2P, \ldots, x^kP\right)$ for some $x \in Z_q^*$.
Output: $e(P,P)^{1/x}$.

The 1-BDHI assumption is polynomially equivalent to the standard BDH assumption. Whereas it is not known if the k-BDHI assumption, for $k > 1$, is polynomially equivalent to BDH.

2.6 Security Models

Security against existential forgery under adaptively chosen message attack [34] is the standard security model for any signature scheme. For ID-based signature schemes *security against existential forgery under adaptively chosen message and ID* attack is standard security notions, which is a generalization of the standard chosen message security notion.

Indistinguishability of encryptions against adaptively chosen ciphertext attack (IND-CCA) [5, 25, 60] is the standard notion of security for public key encryption schemes. A strengthened model of IND-CCA called IND-ID-CCA is the standard notion of security for ID-based encryption schemes.

The model proposed by Bellare and Rogaway in [8] defines provable security for entity authentication and key distribution goals. Later, Blake-Wilson,Johnson and Menezes [15] provided new definitions of security for authenticated key agreement in the public key setting.

Here, we briefly review these security models.

2.6.1 Security Model for ID-based Signature Schemes

Security against existential forgery under adaptively chosen message and ID attack for an ID-based signature scheme which consists of **Setup**, **Extract**, **Sign** and **Verify**algorithms is defined through the following game between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

- $\mathcal{C}$ runs **Setup** algorithm of the scheme. The resulting system parameters are given to $\mathcal{A}$. $\mathcal{C}$ keeps the master-key as a secret with itself.

- $\mathcal{A}$ issues the following queries as he wants:
  (a). *Hash function query.* $\mathcal{C}$ computes the value of the hash for the requested input and sends the value to $\mathcal{A}$.
  (b). *Extract Query.* Given an identity ID, $\mathcal{C}$ returns a private key corresponding to ID which is obtained by running the **Extract** algorithm.
  (c). *Sign Query.* Given an identity ID and a message m, $\mathcal{C}$ returns a signature which is obtained by running **Sign** algorithm.

- $\mathcal{A}$ outputs (ID, m, σ), where ID is an identity, m is a message and σ is a signature such that ID and (ID, m) are not equal to the inputs of any *Extract* and *Sign* queries respectively. $\mathcal{A}$ wins the game if σ is a valid signature of m for ID.

We say that an ID-based signature scheme is secure against existential forgery under adaptively chosen message and ID attack if no polynomially bounded adversary has non-negligible advantage in this game.

A variant of the above game is used to define the security against adaptively chosen message and fixed ID attacks. In this game the identity ID is first fixed. The challenger $\mathcal{C}$ gives to $\mathcal{A}$ system parameters along with ID, and in the final step the adversary $\mathcal{A}$ must output the given ID (together with a message and a signature) as its final result. We say a signature scheme is secure against adaptively chosen message and fixed ID attack if no polynomial adversary has non-negligible advantage in this variant.

2.6.2 Security Model for ID-based Encryption Schemes

The standard security model for a public key encryption scheme involves indistinguishability of encryptions against fully adaptive chosen ciphertext attack (IND-CCA) [5, 25, 60]. Boneh and Franklin [13] strengthened the IND-CCA model to deal with an adversary who possesses private keys corresponding to identities of its choice $ID_1, ID_2, ..., ID_n$ and attacks an identity ID in an ID-based system. They called it IND-ID-CCA model. The model is described through the following game between the challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup: The challenger takes a security parameter k and runs the Setup algorithm. It gives the adversary the resulting system parameters `params`. It keeps the master-key to itself.

Phase 1: The adversary issues queries $q_1, ..., q_m$ where q_i is one of:

- Extraction query $\langle ID_i \rangle$. The challenger responds by running algorithm `Extract` to generate the private key d_i corresponding to the public key $\langle ID_i \rangle$. It then sends d_i to the adversary.
- Decryption query $\langle ID_i, C_i \rangle$. The challenger responds by running algorithm `Extract` to generate the private key d_i corresponding to ID_i. It then runs algorithm `Decrypt` to decrypt the ciphertext C_i using the private key d_i. It sends the resulting plaintext to the adversary.

These queries may be asked adaptively, that is, each query q_i may depend on the replies to $q_1, ..., q_{i-1}$.

Challenge: Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts M_0, M_1 an identity ID on which it wishes to be challenged. The only constraint is that ID did not appear in any private key extraction queries in Phase 1. The challenger picks a random bit $b \in 0, 1$ and sets $C =$ Encrypt(params, ID, M_b). It sends C to the adversary.

Phase 2: The adversary issues more queries $q_{m+1}, ..., q_n$ where q_i is one of:

- Extraction query $\langle ID_i \rangle$ where $ID_i \neq ID$. Challenger responds as in Phase 1.
- Decryption query $\langle ID_i, C_i \rangle \neq \langle ID, C \rangle$. Challenger responds as in Phase 1.

Guess: Finally, the adversary outputs a guess $b' \in 0, 1$ and wins the game if $b = b'$.

The advantage for the adversary $\mathcal{A}$ is given as a function of the security parameter k as below:

$$Adv_{\mathcal{A}}(k) = |Pr[b = b'] - 1/2|$$

We say that an identity based scheme is semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA), if no polynomially bounded adversary $\mathcal{A}$ has non-negligible advantage against the challenger.

Canetti, Halevi and Katz [16] gave Selective ID model, which is slightly weaker than the model described above. In this model the adversary must commit ahead of the time to the identity that it intends to attack, whereas in the standard model given above, the adversary is allowed to choose this identity adaptively.

2.6.3 Security Model for ID-based Authenticated Key Agreement

The security of an authenticated key agreement protocols is analyzed using the model given in [8], which defines provable security for entity authentication and key distribution goals[2, 20]. This model was later extended to the public key setting in [15]. The Bellare-Rogaway model [8] is described as below.

The adversary $\mathcal{A}$ is a probabilistic machine that controls all the communications that take place between parties by interacting with a set of $\prod^i_{U_1,U_2}$ oracles ($\prod^i_{U_1,U_2}$ is defined to be the ith instantiation of a principal U_1 in a specific protocol run and U_2 is the principal with whom U_1 wishes to establish a secret key). The predefined oracle queries are described informally as follows.

- The `Send`(U_1, U_2, i, m) query allows $\mathcal{A}$ to send some message m of her choice to either the client $\prod^i_{U_1,U_2}$ at will. After receiving the query, $\prod^i_{U_1,U_2}$ will compute what the protocol specification demands and returns to $\mathcal{A}$ the response message and/or decision. If $\prod^i_{U_1,U_2}$ has either accepted with some session key or terminated, this will be made known to $\mathcal{A}$.

- The `Reveal`(U_1, U_2, i) query allows $\mathcal{A}$ to expose an old session key that has been previously accepted. $\prod^i_{U_1,U_2}$, upon receiving the query and if it has accepted and holds some session key, will send this session key back to $\mathcal{A}$.

- The $\texttt{Corrupt}(U_1, K_E)$ query allows $\mathcal{A}$ to corrupt the principal U_1 at will, and thereby learn the complete internal state of the corrupted principal. The corrupt query also gives $\mathcal{A}$ the ability to overwrite the long-lived key of the corrupted principal with any value of her choice(i.e. K_E). This query can be used to model the real world scenarios of an insider cooperating with the adversary or an insider who has been completely compromised by the adversary.

- The $\texttt{Test}(U_1, U_2, i)$ query is the only oracle query that does not correspond to any of $\mathcal{A}$'s abilities. If $\prod_{U_1,U_2}^{i}$ has accepted with some session key and is being asked a $\texttt{Test}(U_1, U_2, i)$ query, then depending on a randomly chosen bit b, $\mathcal{A}$ is given either the actual session key or a session key drawn randomly from the session key distribution.

The notion of freshness is used to identify the session keys about which $\mathcal{A}$ ought not to know anything because $\mathcal{A}$ has not revealed any oracles that have accepted the key and has not corrupted any principals knowing the key. Oracle $\prod_{A,B}^{i}$ is fresh (or it holds a fresh session key) at the end of execution, if, and only if, oracle $\prod_{A,B}^{i}$ has accepted with or without a partner oracle $\prod_{B,A}^{i}$, both oracle $\prod_{A,B}^{i}$ and its partner oracle $\prod_{B,A}^{i}$ (if such a partner oracle exists) have not been sent a $\texttt{Reveal}$ query, and the principals A and B of oracles $\prod_{A,B}^{i}$ and $\prod_{B,A}^{i}$ (if such a partner exists) have not been sent a $\texttt{Corrupt}$ query.

Security is defined using the game $\mathcal{G}$ between a malicious adversary $\mathcal{A}$ and a collection of $\prod_{U_x,U_y}^{i}$ for players $U_x, U_y \in \{U_1, U_2, ..., U_{N_p}\}$ and instances $i \in \{1, ..., N_s\}$. $\mathcal{A}$ runs the game simulation $\mathcal{G}$, whose setting is as follows:

- **Stage1:** $\mathcal{A}$ is able to send any `SendClient`, `SendServer`, `Reveal`, and `Corrupt` oracle queries at will in the game simulation $\mathcal{G}$

- **Stage 2:** At some point during $\mathcal{G}$, $\mathcal{A}$ will choose a fresh session on which to be tested and send a `Test` query to the fresh oracle associated with the test session. Note that the test session chosen must be fresh. Depending on a randomly chosen bit b, $\mathcal{A}$ is given either the actual session key or a session key drawn randomly from the session key distribution.

- **Stage 3:** $\mathcal{A}$ continues making any `SendClient`, `SendServer`, `Reveal`, and `Corrupt` oracle queries of its choice.

- **Stage 4:** Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b', which is its guess of the value of b.

Success of $\mathcal{A}$ in $\mathcal{G}$ is measured in terms of $\mathcal{A}$s advantage in distinguishing whether $\mathcal{A}$ receives the real key or a random value. $\mathcal{A}$ wins if, after asking a $\texttt{Test}(U_1, U_2, i)$ query, where $\prod_{U_1,U_2}^{i}$ is fresh and has accepted, $\mathcal{A}$'s guess bit b' equals the bit b selected during the $\texttt{Test}(U_1, U_2, i)$ query. The advantage for the adversary $\mathcal{A}$ is given as a function of the security parameter k as below:

$$\texttt{Adv}_{\mathcal{A}}(k) = 2 \times \Pr[b = b'] - 1$$

We say that a protocol is secure in Bellare-Rogaway model [8] if both validity and indistinguishability requirements are satisfied:

1. When the protocol is run between two oracles in the absence of a malicious adversary, the two oracles accept the same key.

2. For all probabilistic polynomial adversaries $\mathcal{A}$, $\texttt{Adv}_{\mathcal{A}}(k)$ is negligible.

3 ID-Based Signature Schemes

An ID-based signature scheme consists of four phases (algorithms) namely **Setup**, **Extract**, **Sign** and **Verify**. The Private Key Generator (PKG) initializes the system in the **Setup** phase by indicating the system parameters that are made publicly available. The PKG also chooses a `master-key` and keeps it secret. The `master-key` is used in the **Extract** phase to calculate private keys for the participating users in the system. A signer with an identity ID signs a message in the **Sign** phase using the private key given by the PKG corresponding to his identity ID. To verify a signature of an entity with identity ID, a verifier in an ID-based signature scheme just uses the identity ID in the **Verify** phase. Now, we briefly review some of the existing ID-based signature scheme along with their security and computational efficiency in signing and verification phases. Note that whenever we say point, it represents a point on the underlying elliptic curve on which the bilinear parings are realized.

3.1 Shamir's IBS [66]

3.1.1 Description

Setup: The PKG chooses the system parameters as follows:
1. Calculates n as a product of two large prime numbers.
2. Selects a large number e that is relatively prime to $\Phi(n)$ where Φ is Euler's totient function.
3. Selects a one way function h.

`params:` $\langle n, e, h \rangle$ `master-key:` Factorization of n.

Extract: For a user with identity ID, the PKG calculates the corresponding private key g such that $g^e = ID \bmod n$.

Sign: A user with private key g signs a message m by the following operations:
1. Chooses a random number r
2. Calculates $t = r^e \bmod n$
3. Computes $s = g.r^{h(t,m)} \bmod n$

`Signature:` $\sigma = \langle s, t \rangle \in Z_n \times Z_n$.

Verify: The signature $\sigma = \langle s, t \rangle$ of a user with identity ID is valid if and only if the following equality holds good.

$$s^e = ID.t^{h(t,m)} \bmod n$$

3.1.2 Efficiency

The signing and verification phases each requires 2 integer exponentiations, 1 integer multiplication and 1 hash operation.

3.1.3 Security

The security of this signature scheme is based on the difficulty of Integer Factorization Problem (IFP). Bellare et.al [6] proved that the scheme is secure against existential forgery under chosen message attack by proving the underlying SI scheme secure against impersonation under passive attacks, assuming one-wayness of the underlying RSA key generator.

3.2 A Paradoxical IBS [37]

3.2.1 Description

Setup: The PKG chooses the system parameters as follows:
1. Calculates n as a product of two large prime numbers p and q.
2. Selects a large number d that is relatively prime to $\Phi(n)$, where $\Phi()$ is Euler's totient function.
3. Calculates e such that $d.e = 1 \bmod n$.

`params:` $\langle n, d \rangle$ `master-key:` $\langle p, q, e \rangle$

Extract: The user A with identity ID_A sends his identity to the PKG.
1. The PKG verifies the identity and calculates a "shadow" [35] J_A of the identity ID_A, which serves as public key of the user A with identity ID_A.
2. The PKG signs J_A as $S_A = J_A^{-e} \pmod{n}$ and sends S_A to A in a secure way.
3. The user A verifies S_A as $S_A^d = J_A^{-1} \pmod{n}$ and uses it as his private key .

Sign: For a user A with identity ID_A, to sign a message M, he calculates:
1. $r \in_R Z_n$.
2. $u = r^d \pmod{n}$.
3. $b = J_A^M.u^{d^k} \pmod{n}$, by a selecting such a k which satisfies $d^{k-1} \leq M \leq d^k$.
4. $v = r.S_A^b$.
`Signature:` $\sigma = \langle b, v \rangle \in Z_n \times Z_n$
Verify: To verify a signature $\sigma = \langle b, v \rangle$ of a user A on a message M the verifier computes:
1. $u = J_A^b.v^d \pmod{n}$.
2. Accepts the signature if and only if the following equation holds

$$b = J_A^M.u^{d^k} \pmod{n}$$

3.2.2 Efficiency

Each of the signing and verifications phases requires 4 integer exponentiations and 2 integer multiplications in Z_n.

3.2.3 Security

The scheme is secure against existential forgery under chosen message attack assuming RSA is one-way. The security proof is obtained by observing that the scheme is a result of applying the trapdoor SS to identity based signature transform [24] to underlying trapdoor SS scheme already proven secure [1, 59]. Later, Bellare et.al [6] also supported the security proof for the scheme, by applying their framework.

3.3 Sakai-Ohgishi-Kasahara's IBS [64]

3.3.1 Description

Setup: The PKG chooses $s \in_R (Z/q)$ as his master secret key and computes the global public key P_{pub} as sP. It then chooses a random Map-to-Point hash function $H_1 : \{0,1\}^n \rightarrow G_1$.
`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1 \rangle$ `master-key:` $\langle s \rangle$
Extract: The PKG verifies the given identity ID, and computes the secret key for the identity as $S_{ID} = sH_1(ID)$. The component $Q_{ID} = H_1(ID)$ plays the role of the corresponding public key.
Sign: Given a private key S_{ID} and a message $M \in G_1$, choose a $r \in_R Z/q$ and calculate:
1. $S_1 = S_{ID} + rM$
2. $S_2 = rP$
`Signature:` $\sigma = \langle S_1, S_2 \rangle \in G_1 \times G_1$.
Verify: The signature $\sigma = \langle S_1, S_2 \rangle$ of an identity ID on a message M is valid if the following equation holds good.

$$e(Q_{ID}, P_{pub})e(M, S_2) = e(S_1, P)$$

3.3.2 Efficiency

It requires 2 scalar multiplications and 1 point addition in G_1 in signing phase and 3 pairing operations, 1 map-to-point hash operation in verification phase.

3.3.3 Security

Bellare et al. [6] proved that a modified version of the scheme, which is obtained by applying their transforms, is secure against existential forgery under chosen message attack. Later, Libert and Quisquater [45] presented a security reduction from the DHP to a chosen-message attacker against the modified scheme that is more efficient than any other known security reduction [6, 43] for existing identity based signatures [18, 39]. It is still unclear whether the original scheme of [64] can be proved secure against existential forgery under chosen message attack.

3.4 Paterson's IBS [55]

3.4.1 Description

Setup: The PKG chooses $s \in_R (Z/qZ)^{\times}$ as his master secret key and computes the global public key P_{pub} as sP. The PKG also selects a Map-to-point hash function $H_1 : \{0,1\}^* \rightarrow G_1$ and two hash function $H_2 : \{0,1\}^* \rightarrow Z_q$ and $H_3 : G_1 \rightarrow Z_q$.
`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1, H_2, H_3 \rangle$ `master-key:` $\langle s \rangle$.

Extract: The PKG verifies the given identity ID, and computes the secret key for the identity as $S_{ID} = tH_1(ID)$. The component $Q_{ID} = H_1(ID)$ plays the role of the corresponding public key.

Sign: To sign a message $M \in \{0,1\}^*$, a user first chooses $k \in_R Z_q^*$ and computes:
1. $R = kP$
2. $S = k^{-1}(H_2(M).P + H_3(R).D_{ID}$, where k^{-1} is the inverse of k in Z_q^*.

`Signature:` $\sigma = \langle R, S \rangle \in G_1 \times G_1$

Verify: Accept the signature (R, S) of an identity ID on a message M, if the following equation holds good:

$$e(R, S) = e(P, P)^{H_2(M)}.e(P_{pub}, Q_{ID})^{H_3(R)}$$

3.4.2 Efficiency

The signing phase requires 3 scalar multiplications and 2 point additions in G_1, 2 hash (H_2 and H_3) operations. The verification phase requires 3 pairing operations, 2 exponentiation and 1 multiplication in G_2, 2 hash (H_2 and H_3) and 1 map-to-point hash operations.

3.4.3 Security

No formal proof for the security is available.

3.5 Hess's IBS [39]

3.5.1 Description

Setup: The PKG chooses $s \in_R (Z/qZ)^{\times}$ as his master secret key and computes the global public key P_{pub} as sP. The PKG also selects a Map-to-point hash function $H_1 : \{0,1\}^* \rightarrow G_1^*$ and another cryptographic hash function $h : \{0,1\}^* \times G_2 \rightarrow (Z/qZ)^{\times}$.
`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1, h \rangle$ `master-key:` $\langle s \rangle$

Extract: Given the public identity information ID, compute the secret key for the identity as $S_{ID} = tH_1(ID)$. The component $Q_{ID} = H_1(ID)$ plays the role of the corresponding public key.

Sign: To sign a message $m \in \{0,1\}^*$ using the secret key S_{ID}, the signer chooses an arbitrary $P_1 \in G_1^*$, picks a random integer $k \in (Z/qZ)^{\times}$ and computes:
1. $r = e(P_1, P)^k$
2. $v = h(m, r)$.
3. $U = vS_{ID} + kP_1$.

`Signature:` $\sigma = \langle U, v \rangle \in G_1 \times (Z/qZ)^{\times}$.

Verify: To verify the signature $\sigma = (U, v)$ of an identity ID on a message m calculate
1. $r = e(U, P).e(Q_{ID}, -P_{pub})^v$.
2. Accept the signature if and only if $v = h(m, r)$.

3.5.2 Efficiency

The signing phase requires 1 pairing operation, 1 exponentiation in G_2, 1 point addition and 2 scalar multiplication in G_2 and 1 hash (h)operation. The verification phase requires 2 pairing operations, 1 map-to-point hash and 1 exponentiation in G_2.

3.5.3 Security

The signature scheme is secure against existential forgery under adaptive chosen message and *fixed* ID attack in the random oracle model assuming the hardness of CDHP. The proof is obtained through Pointcheval and Stern's [58, 59] forking lemma, which does not yield tight security reductions [33, 42]. Libert and Quisquater [45] stated that the scheme is also secure against *strong existential forgery under chosen-message attacks*, a strengthened model considered in [3, 9].

3.6 Cha-Cheon's IBS [18]

3.6.1 Description

Setup: The PKG chooses $s \in_R (Z/q)$ as his master secret key and computes the global public key P_{pub} as sP. It also chooses one map-to-point hash function $H_1 : \{0,1\}^* \to G_1$ and another cryptographic hash function $H_2 : \{0,1\}^* \times G_1 \to Z/q$

`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1, H_2\rangle$ `master-key:` $\langle s\rangle$

Extract: The PKG verifies the given identity ID, and computes the secret key for the identity as $S_{ID} = sH_1(ID)$. The component $Q_{ID} = H_1(ID)$ plays the role of the corresponding public key.

Sign: To sign a message $m \in \{0,1\}^*$ using the private key S_{ID}, the signer chooses an integer $r \in_R Z/q$ and calculates:
1. $U = rQ_{ID}$
2. $h = H_2(m, U)$
3. $V = (r + h)S_{ID}$

`Signature:` $\sigma = \langle U, V\rangle \in G_1 \times G_1$.

Verify: To verify a signature $\sigma = (U, V)$ of an identity ID on a message m, check whether $(P, P_{pub}, U + hQ_{ID}, V)$ is a valid Diffie-Hellman tuple. This can be accomplished by the equation below:

$$e(P, V) = e(P_{pub}, U + hQ_{ID})$$

Notice that this check could be performed because of the assumption that the group G_1 is a Gap Diffie-Hellman group.

3.6.2 Efficiency

The signing phase requires 1 map-to-point hash, 2 scalar multiplications in G_1, 1 cryptographic hash (H_2)and 1 addition in Z_q. The verification phase requires 2 pairing operations, 1 map-to-point hash, 1 scalar multiplication and 1 point addition in G_1.
The Signing phase of the signature scheme is very efficient as it requires no pairing operations.

3.6.3 Security

The scheme completely secure against existential forgery under adaptively chosen message and ID attacks in the random oracle model assuming the hardness of the CDHP. The proof is obtained through Pointcheval and Stern's [58, 59] forking lemma which does not yield tight security reductions [33, 42]. Libert and Quisquater [45] stated that the scheme is also secure against *strong existential forgery under chosen-message attacks*, a strengthened model considered in [3, 9]

`Note:` Cheon et.al [19] later proposed another ID-based signature scheme based on this scheme that enables secure batch verification. They also showed that [18] is not secure when used for batch verification.

3.7 Chen-Zhang-Kim's IBS without Trusted PKG [21]

3.7.1 Description

Setup: PKG chooses $s \in_R Z_q^*$ and sets the public key $P_{pub} = sP$ and s serves as the master secret key. It also selects one map-to-point hash function $H_1 : \{0,1\}^* \times G_1 \to G_1$ and another cryptographic hash function

$H_2 : \{0,1\}^* \times G_1 \rightarrow Z_q$.
`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1, H_2 \rangle$ `master-key:` $\langle s \rangle$

Extract:
1. A user submits his identity to the PKG and authenticates himself to the PKG.
2. The user selects an integer $r \in_R Z_q^*$ as his long term secret key and sends rP to the PKG.
3. The PKG computes $S_{ID} = sQ_{ID} = sH_1(ID||T, rP)$ and sends it to the user via a secure channel, where T is the life span of the secret key s.
4. The secret key of the user is the pair (S_{ID}, r) and ID is the public key.

Sign: To sign a message m using the secret key (S_{ID}, r) corresponding to the identity (public key) ID the following steps are performed by the signer:
1. Choose $a \in_R Z_q^*$ and compute $U = aQ_{ID}$
2. Compute $V = rH_1(m, U)$
3. Compute $h = H_2(m, U + V)$
4. Compute $W = (a + h)S_{ID}$.

`Signature:` $\sigma = \langle U, V, W, T, rP \rangle \in G_1 \times G_1 \times G_1 \times \{0,1\}^* \times G_1$.

Verify: To verify a signature $\sigma = (U, V, W, T, rP)$ of an identity ID on the message m the verifier does the following:
1. Compute $Q_{ID} = H_1(ID||T, rP)$
2. Compute $H_1(m, U)$ and $h = H_2(m, U + V)$
3. Accept the signature if and only if the following equations hold:

$$e(W, P) = e(U + hQ_{ID}, P_{pub})$$
$$e(V, P) = e(H_1(m, U), rP)$$

Tracing: This phase is executed to detect impersonation attacks done by the PKG. The PKG can impersonate a signature for an identity ID as follows:
1. The PKG chooses a random $r' \in Z_q^*$ and let $Q_{ID'} = H_2(ID||T, r'P)$.
2. He then performs the above described signing on a message m to produce $\langle U', V', W', r', P \rangle$. The signature passes the verification test.

However, the dishonesty of the PKG can be proved by the user by providing a "knowledge proof" of his secret key to an arbiter.

3.7.2 Efficiency

The signing phase requires 2 map-to-point hash, 3 scalar multiplications and 1 point addition in G_1, 1 cryptographic hash (H_2) operation and 1 addition in Z_q. The verification requires 4 pairing operations, 2 map-to-point hash, 1 scalar multiplication and 2 point additions in G_1 and 1 cryptographic hash operations.

3.7.3 Security

The scheme is secure against existential forgery under adaptively chosen message and ID attacks in the random oracle model assuming the hardness of CDHP. The scheme eliminates the inherent *Key Escrow* problem.

4 ID-Based Encryption Schemes

An ID-based encryption scheme consists of four phases namely: **Setup**, **Extract**, **Encrypt** and **Decrypt**. The functionalities of the **Setup** and **Extract** phases are same as those in an ID-based signature scheme. Any user can encrypt a message for an entity with identity ID just by using ID in the **Encrypt** phase. In the **Decrypt** phase, a receiver with an identity ID can decrypt a message encrypted using ID using a private key corresponding to ID obtained from the PKG.

4.1 Boneh-Franklin's IBE [13]

4.1.1 Description

Setup: The PKG selects the master secret key as $s \in Z_q^*$ and calculates the public key $P_{pub} = sP$. It also specifies a map-to-point hash function $H_1 : \{0,1\}^* \rightarrow G_1^*$ and another cryptographic hash function $H_2 : G_2 \rightarrow \{0,1\}^n$.
`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1, H_2 \rangle$ `master-key:` $\langle s \rangle$
Extract: Given an identity string $ID \in \{0,1\}^*$ the PKG verifies the identity and does the following
1. Computes $Q_{ID} = H_1(ID) \in G_1^*$
2. Sets the private key $S_{ID} = sQ_{ID}$
The component Q_{ID} acts as a public key corresponding to the identity ID.
Encrypt: To encrypt a message $m \in \{0,1\}^n$ for a user with the identity ID do the following:
1. Compute $Q_{ID} = H_1(ID) \in G_1^*$
2. Choose a random $r \in Z_q^*$
3. Set the cipher text to be:

$$C = \langle rP, M \oplus H_2(g_{ID}^r) \rangle \qquad \text{where} \qquad g_{ID} = e(Q_{ID}, P_{pub})$$

`Ciphertext:` $C = \langle U = rP,\ V = M \oplus H_2(g_{ID}^r) \rangle \in G_1^* \times \{0,1\}^n$
Decrypt: To decrypt a ciphertext $C = \langle U, V \rangle$ encrypted using the identity ID compute

$$V \oplus H_2(e(d_{ID}, U)) = M$$

4.1.2 Efficiency

The encryption process requires 1 pairing operation, 1 map-to-point hash operation, 1 group exponent in G_2, 1 hash (H_2) operation, 1 scalar multiplication in G_1 and 1 XOR operation. The decryption process requires 1 pairing operation, 1 hash operation (H_2) and 1 XOR operation.

4.1.3 Security

The scheme described above is **BasicIdent**, which is secure against adaptive chosen message attack. By applying the padding technique of Fujisaki-Okamoto [31], the scheme can be extended to **FullIdent**, which is secure against chosen ciphertext attack (IND-ID-CCA secure).

4.2 Cocks' IBE [22]

4.2.1 Description

The scheme makes use of Euler's criteria given below:
Euler's Criterion: Let q be an odd prime and $gcd(a,q) = 1$, then a is quadratic modulo q if and only if $a^{(q-1)/2} = 1 \pmod q$.
If $q = 3 \pmod 4$ and a is a QR modulo q where q is a prime, there is a simple formula to compute square roots $r_{\{1,2\}}$ of QR a modulo q as $r_{\{1,2\}} = \pm a^{(p+1)/4} \pmod q$.
Setup: The PKG generates a universally available modulus M, which is a product of two primes P and Q. The primes numbers P and Q are congruent to 3 mod 4 and they are held privately by the PKG. The PKG also selects a universally available secure hash functions.
`params:` $\langle M$, hash functions $\rangle$ `master-key:` The factorization of M i.e. $\langle P, Q \rangle$.
Extract: When Alice submits her identity string to the PKG, the PKG verifies the identity and does the following:
1. Applies a hash function and produces a value a mod M such that the Jacobi symbol $(\frac{a}{M})$ is $+1$. This involves multiple applications of hash function in a structured way to produce a set of candidate values for a, stopping when $(\frac{a}{M})$ is $+1$. Since $(\frac{a}{M})$ is $+1$, $(\frac{a}{P}) = (\frac{a}{Q})$ and so either a is a square modulo both P and Q, and hence is a square modulo M, or else $-a$ is a square modulo p, Q and hence M. Thus either a or $-a$

will be a quadratic residue modulo P and Q.
2. The PKG presents a root to Alice as her private key corresponding to her identity, which only he can calculate.
Note that, since the only the PKG knows the factorization of M, he can calculate the root r as

$$r = a^{\frac{M+5-(P+Q)}{8}} \bmod M$$

The value r will satisfy either $r^2 = a \bmod M$ or $r^2 = -a \bmod M$ depending upon which of a or $-a$ is a square modulo M.

Encrypt: When Bob wants to encrypt a message for Alice he generates a transport key and encrypts the message with a symmetric encryption algorithm. He sends Alice each bit of the transport key as follows:
1. Let x be a single bit of the transport key coded as $+1$ or -1.
2. Bob chooses a value t at random such that $(\frac{t}{M})$ equals x.
3. He sends $s = (t + a/t) \bmod M$ to Alice.
if Bob doesn't know which of a or $-a$ is the square for which Alice holds the root, he will have to replicate the above process, using different randomly chosen t values to send the same x bits as before, and transmitting $s = (t - a/t)) \bmod M$ each time.

Decrypt Alice can recover the bit x as follows:
1. Alice calculates the Jacobi symbol $(\frac{s+2r}{M})$ using her private key r.
2. Alice recovers the bit x by calculating $(\frac{s+2r}{M}) = (\frac{t}{M}) = x$ as $s + 2r = t(1 + r/t) * (1 + r/t) \bmod M$.
3. Alice decrypts the message once she recovers all the bits of the transport key.

4.2.2 Efficiency

The encryption phase requires calculation of 1 Jacobi Symbol, 2 additions, 2 multiplications and 2 inverses modulo M for each bit of the transport key. It also requires encryption using symmetric algorithm. The decryption phase requires calculating 1 Jacobi symbol and 1 addition modulo M for each transport key bit to extract the transport key. It then requires one symmetric decryption algorithm.

The scheme is very inefficient in terms of bandwidth requirements as each bit of the transport key requires a number of size up to M to be sent.

4.2.3 Security

The scheme is based on the hardness of QRP. The scheme described above is vulnerable to adaptive chosen ciphertext attacks. In this paper, the author has outlined an approach to defend such attacks by adding redundancy to the transport key establishment data. No formal security proof for the scheme is available.

4.3 Hierarchical IBE [32]

The concept of Hierarchical ID-based encryption was first introduced by Horwitz and Lynn in [40]. However, the first secure and practical hierarchical ID-based encryption scheme was an open question till Gentry and Silverberg [32] proposed a scheme. Here we briefly review the scheme of [32].

In an ID-based setting, having a single PKG completely eliminates online lookup. But, it is undesirable for a large network because the PKG becomes a bottleneck. Not only is private key generation computationally expensive, but also the PKG must verify proofs of identity and must establish secure channels to transmit private keys. Hierarchical ID-based encryption (HIBE) allows a root PKG to distribute the workload by delegating private key generation and identity authentication to lower-level PKGs. In a HIBE scheme, a root PKG need only generate private keys for domain-level PKGs, who in turn generate private keys for users in their domains in the next level. Authentication and private key transmission can be done locally. To encrypt a message to Bob, Alice need only obtain the public parameters of Bobs root PKG (and Bobs identifying information); there are no lower-level parameters. Another advantage of HIBE schemes is damage control: disclosure of a domain PKGs secret does not compromise the secrets of higher-level PKGs. It is noted that the previous schemes [13] and [22] do not have these properties. This scheme is derived from the IBE of Boneh and Franklin [13].

4.3.1 Description

The entities in the tree (other than the root) are the users of the tree. Let $Level_i$ be the set of entities at level i, where $Level_0 = \{RootPKG\}$.

Root Setup: The root PKG chooses an arbitrary generator $P_0 \in G_1$, picks a random $s_0 \in Z_q^*$ and sets the public key as $Q_0 = s_0 P_0$. The root PKG also specifies a map-to-point hash function $H_1 : \{0,1\}^* \to G_1$ and a cryptographic hash function $H_2 : G_2 \to \{0,1\}^n$. The global public key is $\langle P_0, Q_0 \rangle$.
`params:` $\langle G_1, G_2, e, P_0, Q_0, H_1, H_2 \rangle$ `master-key:` $\langle s_0 \rangle$

Lower-level Setup: An entity E_t at $Level_t$ picks a random $s_t \in Z_q^*$ which it keeps secret.

Extract: Let E_t be an entity at $Level_t$, with ID-tuple $(ID_1, \ldots, ID_t)$, where $(ID_1, \ldots, ID_t)$ for $1 \leq i \leq t$ is the ID-tuple of E_i's ancestor at $Level_i$. Set S_0 to be the identity element of G_1. Then the entity E_t's parent does the following:
1. computes $P_t = H_1(ID_1, \ldots, ID_t) \in G_1$.
2. sets E_t's secret key S_t to be $S_{t-1} + s_{t-1}P_t = \sum_{i=1}^{t} s_{i-1}P_i$
3. also gives E_t the values $Q_i = s_i P_0$ for $1 \leq i \leq t-1$.

Encrypt: To encrypt a message $M \in \{0,1\}^n$ with ID-tuple $(ID_1, \ldots, ID_t)$, do the following:
1. Compute $P_i = H_1(ID_1, \ldots, ID_i) \in G_1$ for $1 \leq i \leq t$
2. Choose a random $r \in Z_q^*$
3. Set the ciphertext to be

$$C = rP_0, rP_2, \ldots, rP_t, M \oplus H_2(g^r) \qquad \text{where} \qquad g = e(Q_0, P_1) \in G_2$$

`Ciphertext:` $C = \langle U_0 = rP_0, U_2 = rP_2, \ldots, U_t = rP_t, V = M \oplus H_2(g^r) \rangle \in G_1^t \times \{0,1\}^n$

Decrypt: Let $C = \langle U_0, U_2, \ldots, U_t, V \rangle$ be the ciphertext encrypted using the ID-tuple $(ID_1, \ldots, ID_t)$. To decrypt C, the entity E_t computes the plaintext as:

$$V \oplus H_2 \left(\frac{e(U_0, S_1)}{\prod_{i=2}^{t} e(Q_{i-1}, U_i)} \right)$$

4.3.2 Efficiency

For an identity at level t, the encryption process requires 1 pairing operation, t scalar multiplications in G_1, 1 map-to-point hash operation, 1 cryptographic hash (H_2) operation, 1 exponentiation in G_2 and 1 XOR operation. For an identity at level t, the decryption process requires t pairing operations, 1 cryptographic hash operation and 1 XOR operation.
The scheme is quite efficient as the bit-length of the ciphertext and the complexity of decryption grow only linearly with the level of the message recipient.

4.3.3 Security

Chosen ciphertext security of the above scheme is obtained by using the padding technique of Fujisaki-Okamoto [31] in the random oracle model under the assumption that BDH problem is hard. It is a practical, fully scalable, HIBE scheme with total collusion resistance regardless of the number of levels in the hierarchy.

4.4 Authenticated IBE [47]

It uses the same **Setup** and **Extract** algorithms as the Boneh-Franklin [13] scheme except that it requires an extra hash function.

4.4.1 Description

Setup: The PKG chooses a random generator $g \in G_1$. and picks cryptographic hash functions $H_1 : F_q \times G_2 \to \{0,1\}^n$, $H_2 : \{0,1\}^* \to G_1$, $H_3 : \{0,1\}^* \times \{0,1\}^* \to F_q$ and $H_4 : \{0,1\}^n \to \{0,1\}^n$, (for some n). Also selects a secret master key $s \in F_q$.

`params`: $\langle e, G_1, G_2, g, g^s, H_1, H_2, H_3, H_4 \rangle$ `master-key`: $\langle s \rangle$

Extract: The PKG calculates a private key for a user with identity ID_A as $d_A = H_2(ID_A)^s$.

Authenticated-Encrypt: A user A with identity ID_A encrypts a message $M \in \{0,1\}^*$ for another user B with identity ID_B using his private key d_A as below.
1. Choose a random $\sigma \in_R \{0,1\}^n$.
2. Compute $c_1 = H_3(\sigma, M)$ and $c_2 = e(d_A, H_2(ID_B))$.
3. Output the ciphertext $C = \langle \sigma \oplus H_1(c_1, c_2), E_{H_4(\sigma)}(M) \rangle$.

Note that $E_{H_4(\sigma)}(M)$ represents semantically secure symmetric encryption in [31].
`Ciphertext`: $C = \langle \sigma \oplus H_1(c_1, c_2), E_{H_4(\sigma)}(M) \rangle \in \{0,1\}^n \times \mathcal{C}$, where $\mathcal{C}$ represents the ciphertext space of the symmetric algorithm.

Authenticated-Decrypt: A user B decrypts a ciphertext $\langle U, V, W \rangle$ encrypted by another user A with identity ID_A using A's identity ID_A, his private key d_B and `params`. It is described as below.
1. Compute $c_2 = e(H_2(ID_A), d_B)$
2. $\sigma = V \oplus H_1(U, c_2)$
3. $M = D_{H_4(\sigma)}(W)$.
4. Check whether $U = H_3(\sigma, M)$.
5. If not, reject the ciphertext, otherwise output the plaintext M.

Note that $D_{H_4(\sigma)}(W)$ represents semantically secure symmetric decryption in [31].

4.4.2 Efficiency

The encryption and decryption phases each requires 1 pairing operation, 1 map-to-point hash, 3 hash (H_1, H_3 and H_4) operations, 1 XOR operation. In addition, the encryption and decryption schemes require secure symmetric encryption and decryption algorithms respectively.

Authenticated Encryption is faster than plain Encryption because there is one less exponentiation and no point multiplication. Note that both encryption and decryption algorithms benefit greatly from caching c_2, obviating the need for an expensive Weil pairing computation which makes their computation as fast as a symmetric cipher and MAC. (In original system [13] caching helped encryption but not decryption).

4.4.3 Security

This scheme provides non-repudiation as well as integrity and confidentiality . The scheme is secure against adaptive chosen ciphertext attack in the random oracle model assuming the hardness of BDHP.

4.5 Selective-ID Secure IBE without Random Oracles[10]

This scheme is based on non-standard assumption, called Decision Bilinear Diffie-Hellman Inversion (Decision BDHI). Let G_1 be a bilinear group of prime order q. The public keys $(ID) \in Z_q^*$ messages to be encrypted are elements in G_2. This system works as follows:

4.5.1 Description

`Setup`: The PKG selects a generator $g \in_R G_1^*$, elements $x, y \in_R Z_q^*$ and calculate $X = g^x$ and $Y = g^y$.
`params`: $\langle G_1, G_2, e, g, X, Y \rangle$ `master-key`: $\langle x, y \rangle$

KeyGen To create a private key for the public key ID $\in Z_q^*$
1. Pick $r \in_R Z_q$
2. Compute $K = g^{1/(ID+x+ry)} \in G_1$.
3. Output the private key $d_{ID} = (r, K)$.

Encrypt To encrypt a message $M \in G_2$ under public key ID , pick $s \in_R Z_q^*$ and output the ciphertext as

$$C = (g^{s.ID} X^s, Y^s, e(g,g)^s.M)$$

`Ciphertext`: $C = \langle U = g^{s.ID} X^s, V = Y^s, W = e(g,g)^s.M \rangle \in G_1 \times G_1 \times G_2$

Decrypt To decrypt a ciphertext $C = (U, V, W)$ using the private key $d_{ID} = (r, K)$, output the plaintext M as

$$M = W/e(UV^r, K)$$

4.5.2 Efficiency

The encryption process requires 3 exponentiations in G_1, 1 group multiplication in G_1, 1 exponentiation in G_2, 1 group multiplication in G_2 and 1 pairing operation. The decryption process requires 1 exponentiation in G_1, 1 group multiplication in G_1, 1 inversion in G_2 and 1 pairing operation. However, in the encryption phase $e(g, g)$ can be pre-computed once and cached so that encryption does not require any pairing operation.

4.5.3 Security

This scheme is selective identity, chosen plaintext secure without random oracles based on the decision q-BDHI assumption.

4.6 Secure IBE without Random Oracles [11]

In this scheme the bilinear map considered is $e : G_1 \times G_1 \to G_2$, where both G_1 and G_2 are multiplicative groups of same primes order q. $\Sigma = \{1, ..., s\}$ be an alphabet of size s and let $\{H_k : \{0,1\}^w \to \Sigma^n\}_{k \in \mathcal{K}}$ be a family of hash functions.

4.6.1 Description

Setup: The PKG selects a random generator $g \in G_1^*$ and picks a random $\alpha \in Z_q$ and sets $g_1 = g^\alpha$. It then does the following:
1. Picks a random element $g_2 \in G_1$ and construct a random $n \times s$ matrix $U = (u_{i,j} \in G_1^{n \times s})$ where each $u_{i,j}$ is uniform in G_1.
2. Chooses a random k as a hash function key.
`params`: $\langle e, G_1, G_2, g, g_1, g_2, U, k \rangle$ `master key`: $\langle g_2^\alpha \rangle$
Extract: To generate a private key for an identity $ID \in \{0,1\}^\omega$
1. Let $H_k(ID) = a_1...a_n \in \Sigma^n$.
2. Pick random $r_1, ..., r_n \in Z_q$.
3. The Private Key d_{ID} is a

$$d_{ID} = \left(g_2^\alpha . \prod_{i=1}^{n} u_{i,a_i}^{r_i}, g^{r_1}, ..., g^{r_n} \right) \in G_1^{n+1}$$

Encrypt: To encrypt a message $M \in G_2$ under the public key $ID \in \{0,1\}^w$, calculate $H_k(ID) = a_1...a_n \in \Sigma^n$, pick a random $t \in Z_q$, and calculate

$$C = \left(e(g_1, g_2)^t.M, g^t, u_{1,a_1}^t, ..., u_{n,a_n}^t\right)$$

`Ciphertext:` $C = \langle A = e(g_1, g_2)^t.M, B = g^t, C_1 = u_{1,a_1}^t, \ldots, C_n = u_{n,a_n}^t \rangle \in G_2 \times G_1^{n+1}$
Decrypt: To decrypt a ciphertext $C = (A, B, C_1, ..., C_n)$ using the private key $d_{ID} = (d_0, d_1, ..., d_n)$, calculate the plaintext as :

$$A.\frac{\prod_{j=1}^{n} e(C_j, d_j)}{e(B, d_0)} = M$$

4.6.2 Efficiency

The encryption phase requires 1 pairing operation, 1 exponentiation in in G_2, 1 multiplication in G_2 and $(n+1)$ exponentiations in G_1. The decryption phase requires $(n+1)$ pairing operations, $(n+1)$ multiplication in G_2 and 1 inversion in G_2. However, the component $e(g_1, g_2)$ can be pre-computed or can be added to the system parameters so that no pairing operations are computed in the Encryption phase.
Recently, Waters [70] proposed an efficient version of this scheme.

4.6.3 Security

The scheme is completely secure without random oracles and is based on the hardness of DBDHP.

4.7 Public Key Encryption with Keyword Search [12]

Suppose Alice wishes to read her email on a number of devices : laptop, desktop, pager, etc. Alices mail gateway is supposed to route email to the appropriate device based on the keywords in the email. Suppose Bob sends an email with keyword urgent. The gateway routes the email to Alices pager, after testing whether the email contains this keyword urgent without learning anything else about the mail. This mechanism is referred to as Public Key Encryption with Keyword Search (PEKS). To send a message M with keywords $W_1, ..., W_n$, Bob sends

$$E_{A_{pub}}(M)||PEKS(A_{pub}, W_1)||...||PEKS(A_{pub}||W_n)$$

where $E_{A_{pub}}(M)$ is the encryption of M using Alice's public key A_{pub}. The point of searchable encryption is that given $PEKS(A_{pub}, W')$ and a certain trapdoor T_W (that is given to the gateway by Alice), the gateway can test whether $W = W'$. If $W \neq W'$, the gateway learns nothing more about W'.

4.7.1 Description

This construction is based on Boneh-Franklin's IBE [13] scheme. It needs hash functions $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : G_2 \rightarrow \{0,1\}^{logp}$.
KeyGen: Choose $s \in_R Z_q^*$ and set $P_{pub} = sP$. The secret key s and the public key is P_{pub}.
`params:` $\langle G_1, G_2, e, P, H_1, H_2 \rangle$ `master-key:` $\langle s \rangle$
PEKS:
1. Given a key word W and a public key P_{pub}, choose a random $r \in Z_q^*$.
2. Compute $\langle rP, H_2(e(H_1(W), P_{pub})^r) \rangle$.

Trapdoor: Given a key word W and secret key s output $T_W = sH_1(W)$.

Test: Given a trapdoor T_W and a PEKS $S = \langle U, V \rangle$
1. Test if $V = H_2(e(T_W, U))$
2. If true output `yes` otherwise output `no`.

4.7.2 Efficiency

The **PEKS** phase requires 1 paring operation, 1 map-to-point hash function, 1 scalar multiplication in G_1, 1 cryptographic hash function and 1 exponentiation in G_2. The **trapdoor** phase requires 1 scalar multiplication in G_1. The **test** phase requires 1 pairing operation and 1 cryptographic hash function.

4.7.3 Security

The system is proven to be a noninteractive searchable encryption scheme, semantically secure against a chosen key word attack in the random oracle model. The security relies on the hardness assumption of BDH problem.

4.8 Fuzzy IBE [62]

There has been recent interest about the challenge of generating cryptographic keys from biometric inputs. The primary difficulty in generating a strong key from a biometric input is that the measured value of a biometric can change slightly upon each sampling. This effect can be explained by differences in sampling devices, environmental noise, or small changes in the human trait itself.

In a Fuzzy Identity-Based Encryption scheme a user with secret key for the identity ω is able to decrypt a ciphertext encrypted with the public key ω' if and only if ω and ω' are within a certain distance of each other as judged by some metric. Identities are represented as a set of n elements and the set overlapping between two identities is used to measure their similarity. Let the value d represent the error-tolerance in terms of minimal set overlap. When an authority is creating a private key for a user he will associate a random $d-1$ degree polynomial, $p(x)$, with each user with the restriction that each polynomial have the same valuation at point 0, that is $p(0) = y$.

Let $e : G_1 \times G_1 \rightarrow G_2$ denote the bilinear map, where G_1 and G_2 are cyclic groups of prime order q. Let $\Delta_{i,S}(x)$ be the Lagrange Coefficient defined as

$$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}.$$

The set S contains the elements of Z_q. Identities are n-element sets where the elements are members of $Z_q^* \in \mathcal{U}$. $\mathcal{U}$ is the universe of elements defined by the master key holder.

4.8.1 Description

Setup: The following steps are performed
1. Define the universe $\mathcal{U}$, of elements. $\mathcal{U}$ can be the first $|\mathcal{U}|$ elements of Z_q^*.
2. Choose $t_1, ..., t_{|\mathcal{U}|}$ uniformly at random from Z_q.
3. Choose y uniformly at random from Z_q.

`params:` $\langle T_1 = g^{t_1}, ..., T_{|\mathcal{U}|} = g^{t_{|\mathcal{U}|}}, Y = e(g,g)^y \rangle$ `master-key:` $\langle t_1, ..., t_{|\mathcal{U}|}, y \rangle$.

Extract: To generate private key for identity $\omega \in \mathcal{U}$ the following steps are taken.
1. Choose a $d-1$ degree polynomial p such that $p(0) = y$.
2. The private key is calculated as $D_i = g^{\frac{p(i)}{t_i}}$ $\forall i \in \omega$.

Encrypt: A message $m \in G_2$ is encrypted using the identity ω' as follows:
1. Choose a random value $s \in Z_q$
2. Compute $E' = mY^s$
3. Compute $\{E_i = T_i^s\}$ $\forall i \in \omega'$)

`Ciphertext:` $C = \langle \omega', E', E_i = \forall i \in \omega' \rangle \in \mathcal{U} \times G_2 \times G_1^n$

Decrypt: Suppose the cipher text C is encrypted using key corresponding to an identity ω' and we have a key corresponding to identity ω, where $\omega \bigcap \omega' \geq d$. Choose an arbitrary d-element subset S of $\omega \bigcap \omega'$. The ciphertext can be decrypted as

$$E' / \prod_{i \in S} (e(D_i, E_i))^{\Delta_{i,S}(0)}.$$

4.8.2 Efficiency

The encryption phase requires 1 exponentiation in G_2, 1 multiplication in G_2 and i exponentiations in G_1, where i is the number of elements in ω'. The decryption phase requires 1 multiplication in G_2, 1 inversion in G_2 and i number of pairing operations and exponentiations in G_2. Here i represents the number of elements in common for the identities ω and ω'.

4.8.3 Security

The scheme is secure in Selective-ID model assuming hardness of the Decisional Modified BDHP. The scheme can be extended to the chosen ciphertext model by applying the technique of using simulation sound NIZK proofs to achieve chosen ciphertext security [46, 52, 61] as described by Canetti et.al. [16]. Alternatively, Fujisaki and Okamoto [31] can be used to prove its security in the random oracle model.

5 ID-based Key Agreement Schemes

A key agreement protocol is said to provide implicit key authentication (of entity B to entity A) if A is assured that no other entity besides B can possibly ascertain the value of the secret key. A key agreement protocol that provides mutual implicit key authentication is called an authenticated key agreement protocol. The following are the desired properties for an authenticated key agreement scheme.

Known-key Security. Each run of the protocol should result in a unique secret session key. The compromise of one session key should not compromise other session keys.
Forward Secrecy. If long-term private keys of one or more of the entities are compromised, the secrecy of previously established session keys should not be affected. We say that a system has partial forward secrecy if some but not all of the entities long-term keys can be corrupted without compromising previously established session keys, and we say that a system has perfect forward secrecy if the long-term keys of all the entities involved may be corrupted without compromising any session key previously established by these entities. There is a further (perhaps stronger) notion of forward secrecy in identity-based systems, which is called *PKG forward secrecy*, which implies perfect forward secrecy. This is the idea that the PKG's long-term private key may be corrupted (and hence all users long-term private keys) without compromising the security of session keys previously established by any users.
Key-compromise Impersonation Resilience. Compromising an entity A's long-term private key will allow an adversary to impersonate A, but it should not enable the adversary to impersonate other entities to A.
Unknown Key-share Resilience. An entity A should not be able to be coerced into sharing a key with any entity C when in fact A thinks that she is sharing the key with another entity B.
Key Control. Neither entity should be able to force the session key to be a preselected value.

In this section, we briefly review some ID-based authenticated two-party key agreement protocols. An ID-based authenticated key agreement scheme can be specified by three algorithms: **Setup**, **Extract**, and **Key Agreement**.

5.1 Smart's Key Agreement [68]

The scheme is based on the Weil pairing.

5.1.1 Description

Setup: The PKG chooses a secret key $s \in 1, ..l$ where l is the order of the sub group of large prime subgroup over an elliptic curve. Then computes the public key as $P_{pub} = sP$. It also specifies a map-to-point hash function $H_1 : Z_q^* \rightarrow G_1$.
`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1 \rangle$ `master-key:` $\langle s \rangle$
Extract: For a user with identity ID the public key is given by $Q_A = H_1(ID)$ and the PKG generates the associated private key as $S_A = sQ_A$.
Key Agreement:
1. A picks $a \in Z_q^*$ at random and computes $T_A = aP$ and sends T_A to B.
2. B picks $b \in Z_q^*$ at random and computes $T_B = bP$ and sends T_B to A.
3. A computes shared secret $K_{AB} = e(a.Q_B, P_{pub})e(S_A, T_B)$
4. Similarly, B computes shared secret $K_{BA} = e(b.Q_A, P_{pub})e(S_B, T_A)$
5. If both A and B follow the protocol they will compute the same shared secret key:

$$K_{AB} = K_{BA} = e(aS_B + bS_A, P)$$

`Shared Key:` $K = kdf(K_{AB}) = kdf(K_{BA})$, where kdf is the key derivation function. kdf can be defined as a hash function $H_2 : G_2 \rightarrow \{0, 1\}^*$

5.1.2 Efficiency

The key agreement protocol requires 2 pairing operations,2 scalar multiplications in G_1 and 1 map-to-point hash operation for each party to calculate the shared secret key.

5.1.3 Security

Smart informally argues that this protocol has the security properties: mutual implicit key authentication, known key security, forward secrecy, key control, key-compromise impersonation, and unknown key-share resilience. Later, Shim [67] discussed a weakness in this scheme and showed that it does not provide full forward secrecy.

5.2 Scott's Key Agreement [65]

Scott proposed an ID-based key agreement protocol based on Tate pairing.

5.2.1 Description

Setup: The PKG chooses a prime number with $p = 3 \bmod 4$ and $p + 1$ is a product of two primes c, r. It also chooses a map-to-point hash function $H : \{0, 1\}^* \rightarrow G_1$. It then chooses a random $s \in F_q$ as its master-secret.
`params:` $\langle G_1, G_2, e, P, q, H\rangle$ `master-key:` $\langle s\rangle$
Extract: For a user A with identity ID_A, the PKG calculates his private key as $S_A = sQ_A$, where Q_A is A's public key calculated as $Q_A = H(ID_A)$. The user chooses a PIN number α_A, calculates $\alpha_A Q_A$ and $(s - \alpha_A)Q_A$ by subtracting $\alpha_A Q_A$ from S_A. The user stores the values Q_A, $(s - \alpha_A)Q_A$ and can reconstruct S_A using the stored values and the memorized PIN α_A.

Key Agreement:
1. A picks a random $a < r$, computes $T_A = e((s - \alpha_A)Q_A + \alpha_A Q_A, Q_B)^a$ and sends T_A to B.
2. B picks a random $b < r$, computes $T_B = e((s - \alpha_B)Q_B + \alpha_B Q_B, Q_A)^a$ and sends T_B to A.
3. A calculates $K_{AB} = T_B^a$ and similarly B computes $K_{BA} = T_A^b$. If both A and B follow the protocol they will compute the same shared secret key:

$$K_{AB} = K_{BA} = e(Q_A, Q_B)^{sab}$$

`Shared Key:` $K = e(Q_A, Q_B)^{sab}$

5.2.2 Efficiency

The protocol requires 1 pairing operation, 1 map-to-point hash function (H), 1 group addition in G_1 and 2 group exponentiations in G_1.

5.2.3 Security

The author informally argued that the scheme is secure against impersonation assuming the hardness of BDHP.

5.3 Chen and Kudla's Key Agreement [20]

In this work, the authors investigated some security issues related to identity based authenticated key agreement and proposed an efficient protocol, which is similar in construction to the protocol in [68].

5.3.1 Description

The **Setup** and **Extract** algorithms are same as the protocol in [68] and **Key Agreement** protocol is as follows.
Key Agreement:
1. A picks $a \in Z_q^*$ at random and computes $T_A = aQ_A$ and sends T_A to B.
2. B picks $b \in Z_q^*$ at random and computes $T_B = bQ_B$ and sends T_B to A.
3. A computes $K_{AB} = e(S_A, T_B + aQ_B)$ and similarly B computes $K_{BA} = e(T_A + bQ_A, S_B)$. If A and B follow the protocol, they will compute the same shared secret:

$$K_{AB} = K_{BA} = e(Q_A, Q_B)^{s(a+b)}$$

`Shared Key:` $K = kdf(K)$, where kdf is the key derivation function as in [68].

5.3.2 Efficiency

This protocol is efficient than the protocol in [68]. It requires 1 pairing operation, 2 scalar multiplications in G_1, 2 map-to-point hash functions and 1 group addition in G_1 for each party to calculate the shared secret key.

5.3.3 Security

The security of this protocol is analyzed using the security models given in [8, 15], assuming that the adversary makes no *reveal* queries and BDHP is hard, under random oracle model. The authors heuristically argued that this protocol has security properties: partial forward secrecy, imperfect key control, unknown key share resilience and key compromise impersonation.
`Note:` In this work, Chen and Kudla suggested a modification for removing escrow from their scheme, which can also be applied to the protocol in [68]. In the scheme without escrow, although the PKG has the ability to generate the private keys of both users, it is not able to obtain the shared session key for any particular run of the protocol. The authors also suggested another modification that allows key agreement between members of separate domains i.e. key agreement between users under different PKGs.

5.4 Shim's Key Agreement [67]

The author discussed a weakness in the scheme in [68] and proposed an efficient key agreement protocol by making modifications to the one in [68].

5.4.1 Description

The **Setup** and **Extract** algorithms are same as the above protocol [68] and **Key Agreement** algorithm is as follows.

Key Agreement:
1. A picks $a \in Z_q^*$ at random and computes $T_A = aP$ and sends T_A to B.
2. B picks $b \in Z_q^*$ at random and computes $T_B = bP$ and sends T_B to A.
3. A computes shared secret $K_{AB} = e(aP_{pub} + S_A, T_B + Q_B)$.
4. Similarly, B computes shared secret $K_{BA} = e(bP_{pub} + S_B, T_A + Q_A)$.
5. If both A and B follow the protocol they calculate the same shared secret:

$$K_{AB} = K_{BA} = e(P_{pub}, aQ_B + bQ_A + abP)e(Q_A, Q_B)^s$$

`Shared Key:` $K = kdf(K_{AB}||A||B) = kdf(K_{BA}||A||B)$, where kdf is key derivation function.

5.4.2 Efficiency

The key agreement protocol requires 1 pairing operation, 2 scalar multiplications in G_1, 2 point additions in G_1 and 1 map-to-point hash operation for each party to calculate the shared secret key. It clearly is efficient than [68].

5.4.3 Security

The authors claimed that this protocol completely satisfies the security properties Known-key security, Forward Secrecy, Forward Secrecy, Key Compromise Impersonation resilience and Unknown key-share resilience. However, Sun and Hsieh [69] showed that Shims key agreement protocol is insecure against the man-in-the-middle attack and it also does not provide key-compromise impersonation resilience.

5.5 McCullagh and Barreto's Key Agreement [48]

McCullagh and Barreto proposed an efficient ID-based authenticated key agreement protocol that can be instantiated in either escrow or escrowless mode without imposing extra computational effort. Here we describe the key agreement scheme with escrow.

5.5.1 Description

Setup: The PKG chooses a master secret key $s \in Z_q^*$ and calculates its public key as $P_{pub} = sP$. It also specifies a hash function $H_1 : \{0,1\}^* \rightarrow Z_q^*$. The system parameters and the public key are distributed to the users through authenticated channel.
`params:` $\langle G_1, G_2, e, P, P_{pub}, H_1 \rangle$ `master-key:` $\langle s \rangle$
Extract: The PKG verifies identity ID_A of a user A and calculates A's private key as $Q_A = (a+s)P$, where $a = H_1(ID_A)$. Q_A can also be computed as $aP + P_{pub}$. The PKG then calculates A's private key as $S_A = (a+s)^{-1}P$.
Key Agreement:
1. A picks $x_a \in Z_q^*$ at random and computes $T_A = x_a Q_B$ and sends T_A to B.
2. B picks $x_b \in Z_q^*$ at random and computes $T_B = x_b Q_A$ and sends T_B to A.
3. A computes $K_{AB} = e(T_B, S_A)^{x_a}$ and similarly B computes $K_{BA} = e(T_A, S_B)^{x_b}$. If A and B follow the protocol, they will compute the same shared secret key :

$$K_{AB} = K_{BA} = e(P,P)^{x_a x_b}$$

`Shared Key:` $K = e(P,P)^{x_a x_b}$

5.5.2 Efficiency

The scheme is efficient than the schemes in [20, 68]. It requires 1 pairing operation, 2 scalar multiplications in G_1, 1 exponentiation and 1 group addition in G_1 and 1 cryptographic hash operation H_1.

5.5.3 Security

Although, the scheme carries a proof of security in Bellare and Rogaway model [8], Xie [71] pointed out a flaw, where a malicious adversary is able to launch a key compromise attack successfully. Xie[71] suggested modifications for the protocol to be secure from the attack and to attain Known-Key Security, Perfect-Forward-Secrecy, Key-Compromise Impersonation, Unknown Key-Share,and Key control. Recently, Choo [44] also demonstrated that the scheme and its variant, proposed to resist the attack by Xie[71], are not secure if the adversary is allowed to reveal non-partner players who had accepted the same session key.
`Note:` In this work McCullagh and Barreto also showed that scheme described here can also be used in escrowless mode with slight changes, using conventional Tate pairing. They also described a scheme for key agreement between members of different domains, which can be used in escrow and escrowless mode. The scheme is twice as efficient as the scheme in [20] without precomputation.

6 Conclusions

To summarize here an identity based cryptosystem has the following properties:
– user's public key is his identity (or derived from identity)
– no requirement of public key directories
– message encryption and signature verification processes require only receivers' and signers' identity respectively along with some system parameters.

These properties make ID-based cryptosystems advantageous over the traditional PKCs, as key distribution is far simplified. However they suffer from the inherent drawback of key escrow i.e. PKG knows the users' private keys. They also require a secure channel for key issuance between PKG and user.

In this work we surveyed three fundamental ID-based cryptographic primitives *Digital Signature*, *Encryption* and *Key Agreement*, which are based on the mathematical concepts of Integer Factorization, Quadratic Residues and Bilinear Pairings. We reviewed several schemes along with their efficiency and security considerations. The efficiency and security concerns of several schemes are presented in a structured form so that a uniform base can be achieved for analyzing them. The survey helps in understanding the research work that has been carried out in the area of ID-based cryptosystems for the past twenty years.

References

[1] M. Abdalla, J.H. An, M. Bellare and C. Namprempre. *From identification to signatures via the Fiat-Shamir tranform: Minimizing assumptions for security and forward-security.* In Advances in Crytology-Eurocrypt'02, LNCS 2332, pp. 418-433, Springer-Verlag, 2002.

[2] S. Al-Riyami and K.G. Paterson. *Tripartite authenticated key agreement protocols from pairings.* In Proceedings of IMA Conference on Cryptography and Coding, LNCS 2898, pp. 332-359, Springer-Verlag, 2003.

[3] J.-H. An, Y. Dodis and T. Rabin. *On the security of joint signature and encryption.* In Advances in Cryptology-Eurocrypt '02, LNCS 2332, pp. 83-107, Springer-Verlag, 2002.

[4] P.S.L.M. Barreto. *The Pairing-Based Crypto Lounge.* http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html.

[5] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. *Relations among notions of security for public-key encryption schemes.* In Advances in Cryptology-Crypto'98, LNCS 1462, pp. 26-45, Springer-Verlag, 1998.

[6] M. Bellare, C. Namprempre and G. Neven. *Security Proofs for identity-based identification and signature Schemes.* http://www.cse.ucsd.edu/users/mihir/crypto-research-papers.html Extended Abstract in Advances in Crptology-Eurocrypt'04, LNCS 3027, pp. 268-286, Springer-Verlag, 2004.

[7] M. Bellare and A. Palacio. *GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attack.* In Advances in Crptology-Crypto'02, LNCS 2442, pp. 162-177, Springer-Verlag, 2002.

[8] M. Bellare and P. Rogaway. *Entity authentication and key distribution.* In Advances in Cryptology-Crypto'93, LNCS 0773, pp. 232-249, Springer- Verlag, 1994. Full version available at http://www-cse.ucsd.edu/users/mihir.

[9] D. Boneh and X. Boyen. *Short Signatures Without Random Oracles*, In Advances in Cryptology-Eurocrypt'04, LNCS 3027, pp. 56-73, Springer-Verlag, 2004.

[10] D. Boneh, X. Boyen, *Efficient Selective ID Secure Identity Based Encryption without Random Oracles*,In Advances In Cryptology-Eurocrypt'04, LNCS 3027, pp. 223-238, Springer-Verlag, 2004.

[11] D. Boneh, X. Boyen, *Secure Identity Based Encryption Without Random Oracles*, In Advances in Cryptology-Crypto'04, LNCS 3152, pp. 443-459, Springer-Verlag, 2004.

[12] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, *Public key encryption with keyword search*, In Advances in Cryptology-Eurocrypt'04, LNCS 3027, pp. 506-522, Springer-Verlag, 2004.

[13] D. Boneh. and M. Franklin. *Identity-based Encryption from the Weil pairing.* SIAM J. of Computing, 32(3):586-615, 2003. Extended abstract in Advances in Crptology-Crypto'01, LNCS 2139, pp.213-229, Springer-Verlag, 2001.

[14] D. Boneh, B. Lynn and H. Shacham. *Short Signatures from the Weil Pairing*, In Advances in Crptology-Asiacrypt'01, LNCS 2248, pp. 514-532, Springer-Verlag, 2001.

[15] S. Blake-Wilson, D. Johnson, and A. Menezes. *Key agreement protocols and their security analysis.* In Proceedings of the 6th IMA International Conference on Cryptography and Coding, LNCS 1355, pp. 30-45, Springer-Verlag, 1997.

[16] R. Canetti, S. Halevi and J Katz. *A forward-secure public-key encryption scheme.* In Advances in Crptology-Eurocrypt'03, LNCS 2656, pp. 255-271, Springer-Verlag, 2003.

[17] R. Canetti, S. Halevi and J Katz. *Chosen-ciphertext security from identity based encryption.* In Advances in Cryptology-Eurocrypt'04, LNCS 3027, pp. 207-222, Springer-Verlag, 2004.

[18] J. Cha and J.H. Cheon. *An Identity-Based Signature from Gap Diffie-Hellman Groups*. In Public Key Cryptography-PKC'03, LNCS 2567, pp.18-30, Springer-Verlag, 2003.

[19] J. H. Cheon, Y. Kim, H. J. Yoon, *A New ID-based Signature with Batch Verification*, Cryptology ePrint Archive, Report 2004/131, 2004. http://eprint.iacr.org/2004/131.

[20] L. Chen and C. Kudla. *Identity based authenticated key agreement from pairings*. Cryptology ePrint Archive, Report 2002/184, 2002. http://eprint.iacr.org/ 2002/184.

[21] X. Chen, F. Zhang, K. Kim, *A New ID-based Group Signature Scheme from Bilinear Pairings*, In Proceedings of WISA'03, LNCS 2908, pp.585-592, Springer-Verlag, 2003.

[22] C. Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, International Conference on Cryptography and Coding-Proceedings of IMA, LNCS 2260,pp. 360-363, Springer-Verlag, 2001.

[23] X. Ding and G. Tsudik. *Simple Identity-Based Cryptography with Mediated RSA*. In Proceedings of CT-RSA '03, LNCS 2612, pp.193-210, Springer, 2003.

[24] Y. Dodis, J. Katz, S. Xu and M. Yung. *Strong key-insulated signature schemes*. In Public Key Cryptography-PKC'03, LNCS 2567, pp. 130-144, Springer-Verlag, 2003.

[25] D. Dolev, C. Dwork, and M. Naor. *Non-malleable cryptography*. SIAM J. Computing, 30(2):391-437, 2000.

[26] R. Dutta, R. Barua and P. Sarkar. *Pairing Based Cryptographic Protocols: A Survey*. Cryptology ePrint Archive, Report 2004/064, 2004.

[27] U. Fiege, A. Fiat and A. Shamir. *Zero-knowledge proofs of identity*, J. Cryptology, Vol.1, pp. 77-94, Springer, 1988.

[28] A. Fiat and A. Shamir, *How to prove yourself: Practical Solutions to identification and signature problems*, In Advances in Cryptology-Crypto'86, LNCS 0263, pp. 186-194, Springer-Verlag, 1987.

[29] M. Fischlin and R. Fischlin. *The representation problem based on factoring*. In Proceedingsn of CT-RSA'02, LNCS 2271, pp. 96-113, Springer-Verlag, 2002.

[30] G. Frey and H. Ruck. *A Remark concerning m-divisibility and the discrete logarithm in the divisor class of group of curves*, Mathematics of Computation, 62:865-874, 1994.

[31] E. Fujisaki and T. Okamoto. *Secure integration of asymmetric and symmetric encryption schemes*. In Advances in Cryptology-Crypto'99, LNCS 1666, pp. 537-554, Springer-Verlag, 1999.

[32] C. Gentry and A. Silverberg, *Hierarchical ID-Based Cryptography*, In Advances in Cryptology-Asiacrypt'02, LNCS 2501 ,pp.548-566, Springer-Verlag, 2002.

[33] E.-J. Goh and S. Jarecki. *A signature scheme as secure as the diffie-hellman problem*. In Advances in Cryptology-Eurocrypt 03, LNCS 2656, pp. 401-415, Springer-Verlag, 2003.

[34] S. Goldwasser, S. Micali and R. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*. SIAM J. Computing, 17(2):281-308, 1988.

[35] L.C. Guillou and J. -J. Quisquater. *Efficient digital public-key signature with shadow*, In Advances in Cryptology-Crypto'87, LNCS 0293, pp. 223, Springer-Verlag, 1987.

[36] L. C. Guillou and J.-J. Quiaquater. *A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory*, In Advances in Cryptology-Eurocrypt'88, LNCS 0330, pp.123-128, Springer-Verlag, 1988.

[37] L.C. Guillou and J.-J. Quisquatar. *A "paradoxical" identity-based signature scheme resulting from zero-knowledge*. In Advances in Cryptology-Crypto'88 , LNCS 0403, pp. 216-231, Springer-Verlag, 1990.

[38] P. Guttman. *PKI: Its not dead, just resting*. IEEE Computer, 35(8):41-49, 2002. Extended version available at www.cs.auckland.ac.nz/ pgut001/pubs/notdead.pdf

[39] F. Hess. *Efficient Identity Based Signature Schemes Based on Pairings*. In Selected Areas in Cryptography-SAC'02, LNCS 2595, pp.310-324, Springer- Verlag, 2003.

[40] J. Horwitz and B. Lynn. *Toward Hierarchical Identity-Based Encryption*, In Advances in CryptologyEurocrypt'02, LNCS 2332, pp. 466481, Springer-Verlag, 2002.

[41] A. Joux, *A one round protocol for tripartite Diffie-Hellman*, In Proc. of ANTS, LNCS 1838, pp. 385-394, 2000.

[42] J. Katz and N. Wang. *Efficiency improvements for signature schemes with tight security reductions*, Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 155-164, 2003.

[43] K. Kurosawa and S.-H. Heng. *From Digital Signature to ID-based identification/signature.* In Public Key Cryptography-PKC'04, LNCS 2947, pp. 248-261, Springer-Verlag, 2004.

[44] K.K.R. Choo. *Revisit Of McCullagh–Barreto Two-Party ID-Based Authenticated Key Agreement Protocols*. Cryptology ePrint Archive, Report 2004/343, 2004. http://eprint.iacr.org/2004/343.

[45] B. Libert and J.-J. Quisquater. *The Exact Security of an identity based signature scheme and its applications*, Cryptology ePrint Archive, Report 2004/102, 2004. http://eprint.iacr.org/2004/102.

[46] Y. Lindell. *A simpler construction of CCA2-secure public-key encryption under general asumptions.* In Advances in Cryptology-Eurocrypt'03, LNCS 2656, 241-254, Springer-Verlag, 2003.

[47] B. Lynn. *Authenticated ID-based Encryption* Cryptology ePrint Archive, Report 2002/072, 2002. http://eprint.iacr.org/2002/072.

[48] N. McCullagh and P. S. L. M. Barreto. *A New Two-Party Identity-Based Authenticated Key Agreement*. Cryptology ePrint Archive, Report 2004/122, 2004. In Proceeding of CT-RSA 2005. http://eprint.iacr.org/2004/122.

[49] A.J. Menezes, T. Okamoto and S. A. Vanstone. *Reducing elliptic curve logarithms in a finite field*, IEEE Trans. Inf. Theory, 39(5):1639-1646, 1993

[50] A.J. Menezes, P. C. van Oorschot and S. A. Vanstone. *Hand Book of Applied Cryptography.* ISBN 0-8493-8523-7, CRC Press.

[51] A.J. Menezes, M. Qu and S. Vanstone. *Some new key agreement protocols providing mutual implicit authentication*. In Proceedings of the Second Workshop on Selected Areas in Cryptography, SAC 95, pp. 22-32, 1995.

[52] M. Naor and M. Yung. *Public-key cryptosystems provably secure against chosen ciphertext attacks.* In ACM Symposium on Theory of Computing - STOC, pp. 427437, 1990.

[53] K. Ohta and T. Okamoto. *A modification of the Fiat-Shamir scheme*. In Advances in Cryptology-Crypto'88, LNCS 0403, pp. 232-243, Springer- Verlag, 1990.

[54] H. Ong and C.-P. Schnorr. *Fast signature generation with a Fiat-Shamir-like scheme*. In Advances in Cryptology-Eurocrypt'90, LNCS 0473, pp. 432-440, Springer-Verlag, 1990.

[55] K. G. Paterson. *ID-based signatures from pairings on elliptic curves*, Cryptology ePrint Archive, Report 2002/004, 2002. http://eprint.iacr.org/2002/004.

[56] K.G. Paterson and G. Price. *A comparison between traditional Public Key Infrastructures and Identity-Based Cryptography*, Information Security Technical Report, 8(3):57-72, Elsevier Ltd, 2003.

[57] J. Pieprzyk, T. Hardjono and J. Seberry, *Fundamentals of Computer Security*, ISBN 3540431012, Springer, 2003

[58] D. Pointcheval and J. Stern, *Security Proofs for signature schemes*, In Advances in Cryptology-Eurocrypt'96, LNCS 1070, pp. 387-398, Springer-Verlag, 1996.

[59] D. Pointcheval and J. Stern. *Security Arguments for Digital Signatures and Blind Signatures*. J. Cryptology, 13(3):361-396, Springer, 2000.

[60] C. Rackoff and D. Simon. *Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attacks*. In Advances in Cryptology-Crypto'91, LNCS 576, pp. 433-444. Springer-Verlag, 1991

[61] A. Sahai. *Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security.* In Proceedings of 40th IEEE Symp. on Foundations of Computer Science, pp. 543, 1999.

[62] A. Sahai and B. Waters, *Fuzzy Identity-Based Encryption*. In Advances in Cryptology-Eurocrypt'05. LNCS 3494, pp. 457-473, Springer, 2005. Available at http://eprint.iacr.org/2004/086.

[63] R. Sakai and M. Kasahara. *ID based cryptosystems with pairing on elliptic curve.* In Symposium on Cryptography and Information SecuritySCIS03, Hamamatsu, Japan, 2003. http://eprint.iacr.org/2003/054.

[64] R. Sakai, K. Ohgishi and M. Kasahara. *Cryptosysytems based on pairing*. In Symposium on Cryptography and Information Security-SCIS'00, 2000.

[65] M. Scott. *Authenticated ID-based key exchange and remote log-in with insecure token and PIN number.* Cryptology ePrint Archive, Report 2002/164, 2002. http: //eprint.iacr.org/2002/164/

[66] A. Shamir. *Identity-based Cryptosystems and Signature Schemes*. In Advances in Cryptology-Crypto'84, LNCS 196, pp. 47-53, Springer-Verlag, 1984.

[67] K. Shim, *Effcient ID-based authenticated key agreement protocol based on the Weil pairing*, Electron. Lett., 39(8), pp. 653-654, 2003.

[68] N. P. Smart, *An ID-based authenticated key agreement protocol based on the Weil pairing*, Electron. Lett., 38(13), pp. 630-632, 2002.

[69] H.-M Sun and B.-T. Hsieh. *Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings*. Cryptology ePrint Archive, Report 2003/113, 2003. http://eprint.iacr.org/2003/113.

[70] B. R. Waters, *Efficient Identity-Based Encryption Without Random Oracles*, In Advances in Cryptology-Eurocrypt'05. LNCS 3494, pp. 114-127, Springer, 2005. Available at http://eprint.iacr.org/2004/180.

[71] G. Xie. *Cryptanalysis of Noel McCullagh and Paulo S. L. M. Barreto's two-party identity-based key agreement*. Cryptology ePrint Archive, Report 2004/308, 2004. http://eprint.iacr.org/2004/308.

[72] G. Xie. *An ID-Based Key Agreement Scheme from pairing*. Cryptology ePrint Archive, Report 2005/093, 2005. http://eprint.iacr.org/2005/093.

[73] X. Yi, *An Identity-Based Signature Scheme From the Weil Pairing*, IEEE Communication Letters, 7(2):76-78, IEEE, 2003.