Journal of Theoretical and Applied Information Technology
15th January 2020. Vol.98. No 01
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TOWARDS CERTFICATELESS PUBLIC KEY INFRASTRUCTURE: A PRACTICAL ALTERNATIVE OF THE TRADITIONAL PKI

[1]EIHAB B.M. BASHIER, [2]MOHAMMED A. HASSOUNA, [3]TAOUFIK BEN JABEUR

[1,3]Dept of Mathematics, CAAS, Dhofar University, P.O. Box: 2509, Salalah, Oman

[2]Faculty of Computer Studies, National Ribat University, P.O. Box: 55, Khartoum, Sudan

E-mail: [1]ebashier@du.edu.om, [2]hassounatop@gmail.com, [3]tjabeur@du.edu.om

ABSTRACT

Although the maturity and efficiency of the traditional PKI in managing the public keys of the enterprise users, it still has two central and interrelated challenges or limitations when the number of users get large: Scalability and Key management. These two challenges are of great concern to the organization's information security officials, who are working to manage public key infrastructure, especially when the organization's growth rate becomes large. The most two significant alternatives for the traditional PKI are: Identity-based Cryptography and Certificateless Cryptography. The Identity-based Cryptography (IBC) provides an easy way to manage the public keys of its users, without need to any kind of certificate and its managing overhead, where the identity of a user is its public key. The private key is provided by a trusted Key Generation Centre (KGC) after an authentication process that the user must follow. IBC has many security features, and there are many schemes in the literature that are based on this new concept. It has one major problem: The Key escrow, where all the private keys of the users are generated centrally by the KGC. Certificateless Cryptography is another important alternative for the traditional PKI. It provides solution to the key escrow problem encountered by the IBC and raises many nice security features. This paper provides a robust certificateless signature scheme, which is provably secure in the Random Oracle Model (ROM). Then, it presents a Certificateless Hierarchal Encryption scheme, which provides trust level 3, so, can solve many practical problems, based on the Certificateless Cryptography as a public key infrastructure.

Keywords: *Certificateless cryptography, Public key infrastructure, Random Oracle Model, Security services, Trust levels.*

1. INTRODUCTION

The traditional face-to-face transactions require only minimal interaction and normally do not necessitate the use of other security and integrity mechanisms. However, for e-commerce on the Internet, additional security and integrity mechanisms are necessary. Security is important when data is either confidential or commercial. However, many networks are not secure, so an eavesdropper can conveniently intercept and capture the sensitive and valuable data that are moving in an insecure channel [2].

In general, the security of data against unauthorized access can be accomplished by several methods, the first method is based on symmetric cryptography, which provides the confidentiality (*providing the secrecy and privacy of data*) and the integrity (*ensuring that data cannot be corrupted or modified, and transactions cannot be altered*) of the data (hash functions and digital signature). The second method is public key cryptography, which provides in addition to confidentiality and the integrity the authentication (*verifying that the identity of entities is provided using public key certificates and digital signature*) and non-repudiation (*ensuring that data, cannot be renounced or a transaction denied*).

These four security requirements are provided by the Public Key Infrastructure (PKI), which is the name given to the combination of hardware, software, people and policies with aim to manage digital certificates (create, issue, modify, store and remove digital certificates). A Digital Certificate associates an identity with the private-public key pair of the owner of the identity. The main role of the PKI is to provide a system for distributing and managing digital certificates, to enable users of an insecure public network (such as the Internet) to securely and privately exchange data using a public and a private cryptographic key pair that is obtained and shared through a trusted authority.

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A general purpose of the PKI raised from the simple fact that, in order to use a public key, one should have a guarantee that the public key is truly belonging to the entity that claims to own it. This guarantee of authenticity is achieved by means of a certificate, i.e., a digitally signed document binding the identity of the keyholder to its public key.

Even though, the digital certificates are considered as the best form of authentication, but they are hard to manage, especially in terms of certificate validation and revocation problems. When the certificate is to be revoked, then third parties cannot rely on that certificate unless the CA distributes certificate status information indicating whether the certificate is currently valid. Certificate revocation problem becomes harder when the number of PKI users becomes large, and this problem is termed as the scalability problem. In addition, solving this problem requires a lot of infrastructure, and the need for this infrastructure taken as a reason against widespread implementation of public-key cryptography and the PKI [9].

As stated in [2], the currently existed PKI technologies suffer the scalability and certificate management, making the authentication service inefficient, particularly with devices, which are limited in their resources. Furthermore, the implementation of PKI requires a lot of infrastructure and high transmission costs to be operated and managed in an environment such as the mobile banking.

The Identity-based Public Key Cryptography (ID-PKC) [28] came to address these two problems but could not offer true non-repudiation due to the key escrow problem [1, 8]. In ID-PKC, an entity's public key is derived directly from certain aspects of its identity, for example, an IP address belonging to a network host, or an email address associated with a user. Private keys are generated for entities by a trusted third party called a private key generator (PKG). The first fully practical and secure identity-based public key encryption scheme was presented in [5]. Since then, rapid development of ID-PKC has taken place. The ID-PKC suffers a key escrow problem that the PKG knows all users' private keys in the system and furthermore cannot offer true non-repudiation.

In 2003 Al-Riyami and Paterson [1] introduced the concept of Certificateless Public Key Cryptography (CL-PKC) to overcome the key escrow limitation of the identity-based public key cryptography (ID-PKC). In CL-PKC a trusted third party called Key Generation Center (KGC) supplies a user with a partial private key. Then, the user combines the partial private key with a secret value (that is unknown to the KGC) to obtain his full private key. In this way the KGC does not know the user's private key. Then the user combines his secret value with the KGC's public parameters to compute his public key.

The certificateless cryptography is considered a combination between PKI and identity-based cryptography [1]. It combines the best features of the PKI and ID-PKC, such as lack of certificates, no key escrow property, reasonable trust to trust authority and lightweight infrastructure [25]. It provides a solution to the non-repudiation problem, through enabling a user to generate his/her full long-term private key, where the trusted third party is unable to impersonate the user. The use of certificateless cryptography schemes have appeared in literature, this includes the uses of certificateless encryption [8], [35]; certificateless signatures [33, 37, 43] and certificateless signcryption [34], [38-39].

Almost all the CLPKC schemes found in the literature focus on algorithms of public parameters generation, public/private key generation of system's parties, encryption and decryption processes, but leaves many key problems without clear solutions. Such problems like how the system parameters are published and where, what the authentication method that can be used between the users and the KGC server, what the users shall do if the KGC updates its parameters and how they can be notified, what is the format of the elements of the CLPKC system, and so forth. Also, there are other challenges regarding trust models, such as to determining whether the traditional PKI trust models can be applied to CL-PKI, whether a PKI can be migrated to CL-PKI, and whether an existing PKI-based system can be integrated with another CL-PKI-based system. In this chapter, an integrated model of Certificateless Public Key Infrastructure (CL-PKI) is studied. It is assumed that there exists a Registration Authority (RA) which is responsible for user's registration in the system, and a Key Generation Center (KGC) that is used to generate the system parameters and master secret and publish the system parameters on the public directory (PD) and keep the master secret secure.

In this paper, we show the gaps between the traditional PKI and the CL-PKC and as an extension

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to our work in [17], we show how it is possible to fill this gap such that the currently existing CL-PKC can be promoted to an integrated certificateless public key infrastructure.

We organized the rest of this paper as follows. The basics of the certificateless cryptography will be presented in Section 2. In Section 3, we state the gap between the PKI and CL-PKC. Then in Section 4, we show what is required to fill this gap in order to obtain a certificateless PKI. In Section 5 are the security proofs for the certificateless cryptography. Finally, Section 6 concludes the chapter.

2. CERTIFICATELESS CRYPTOGRAPHY (CL-PKC)

In this section, we introduce the certificateless cryptography technique, that was originally described by Al-Riyami and Paterson in [1]. We first give the necessary mathematical background to understand the scheme.

2.1 Backgrounds

In this section, we give backgrounds about pairing in elliptic curves and its related cryptography primitives. Throughout the Chapter, G_1 denotes an additive group of prime order q and G_2 a multiplicative group of the same order. We let P denote a generator of G_1.

2.1.1 Pairing in Elliptic Curves Cryptography

A pairing is a map $e: G_1 \times G_1 \to G_2$, with the following properties:

1. The map e is bilinear: given $Q, W, Z \in G_1$, we have:
   $e(Q, W + Z) = e(Q, W) \cdot e(Q, Z)$ and $e(Q + W, Z) = e(Q, W) \cdot e(W, Z)$.

I. CONSEQUENTLY, FOR ANY $a, b \in Z_q$, WE HAVE:
$e(a \cdot Q, b \cdot W) = e(Q, W)^{ab} = e(abQ, W)..etc.$

2. The map e is non-degenerate: $e(P, P) \neq 1_{G_2}$.
3. The map e is efficiently computable.

Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. We refer to [3-7], [10-11] for a more comprehensive description of how these groups, pairings and other parameters should be selected in practice for efficiency and security.

2.1.2 Deffie-Hellman Problems in Elliptic Curves Cryptography

In this section, we introduce two Deffie-Hellman problems in the elliptic curves cryptography, namely, the bilinear and generalized bilinear Deffie-Helman problems.

Definition 1. *Bilinear Diffie-Hellman Problem (BDHP):*
Let G_1, G_2, P and e be as above. The BDHP in G_1, G_2, e is as follows: Given P, aP, bP and cP with uniformly random choices of $a, b, c \in Z_q$, compute $e(P, P)^{abc} \in G_2$. An algorithm A has advantage ε in solving the BDHP in G_1, G_2, e if: $\Pr[A(P, aP, bP, cP) = e(P, P)^{abc}] = \varepsilon$.
Here, the probability is measured over the random choices of $a, b, c \in Z_q$ and the random bits of A.

Definition 2. *Generalized Bilinear Diffie-Hellman Problem (GBDHP)*
Let G_1, G_2, P and e be as above. The GBDHP in G_1, G_2, e is as follows: Given P, aP, bP and cP with uniformly random choices of $a, b, c \in Z_q$, output a pair $(Q \in G_1^*, e(P, Q)^{abc} \in G_2)$. An algorithm A has advantage ε in solving the GBDHP in G_1, G_2, e if: $\Pr[A(P, aP, bP, cP) = e(P, Q)^{abc}] = \varepsilon$.

Notice that the BDHP is a special case of the GBDHP, in which the algorithm outputs the choice $Q = P$. While the GBDHP may appear to be in general easier to solve than the BDHP (because the solver must choose Q), there is no polynomial-time algorithm that can solve either problem, when the groups G_1, G_2 and the pairing e are appropriately selected. If the solver knows $s \in Z_q$ such that $Q = sP$, then the problems are of course equivalent. The GBDHP is related to generalized versions of the computational Diffie-Hellman problems in G_1 and G_2 in the same way that the BDHP is related to the standard computational Diffie-Hellman problem in those groups [6-7].

Definition 3. *BDH Parameter Generator:*
As in [5], a randomized algorithm $\mathcal{G}$ is a BDH parameter generator if $\mathcal{G}$:

1. takes security parameter $k \geq 1$,
2. runs in polynomial time in k, and
3. outputs the description of groups G_1 and G_2 of prime order q and a pairing $e: G_1 \times G_1 \to G_2$.

Formally, the output of the algorithm $\mathcal{G}(1^k)$ is (G_1, G_2, e).

There are other computational hardness assumptions related to the elliptic curves groups and are infeasible in polynomial time [6-7].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1. Elliptic Curve Discrete Logarithm Problem: Given $P, Q \in G_1$, find an element $a \in Z_q$ such that $Q = a \cdot P$.
2. Computation Elliptic Curve Diffie-Hellman Problem: Given $(P, aP, bP, cP) \in G_1$ where $a, b, c \in Z_q$, compute abP.

2.2 Al-Riyami and Paterson Scheme

In 2003 Al-Riyami and Paterson [1] introduced the concept of Certificateless Public Key Cryptography (CL-PKC) to overcome the key escrow limitation of the Identity-based Cryptography. In CL-PKC a trusted third party called Key Generation Center (KGC) supplies a user with partial private key, the user then combines the partial private key with a secret value (unknown to the KGC) to obtain his/her full private key. In this way the KGC does not know users' private keys. Then the user combines the same secret value with the KGC's public parameters to compute his/her public key.

Compared to Identity-based Public Key Cryptography (ID-PKC), the trust assumptions made of the trusted third party in CL-PKC are much reduced. In ID-PKC, users must trust the private key generator (PKG) will not abuse its knowledge of private keys in performing passive attacks, while in CL-PKC, users need only trust the KGC will not actively propagate false public keys [1].

In CL-PKC users can generate more than one pair of keys (private and public) for the same partial private key. To guarantee that KGC does not replace users' public keys, Al-Riyami and Paterson [1] introduced a binding technique to bind a user's public key with his/her private key. In their binding scheme, the user first fixes his/her secret value and his/her public key and supplies the KGC with his/her public key. Then the KGC redefine the identity of the user to be the user's identity concatenated with his/her public key. By this binding scheme the KGC replacement of a public key apparent, and equivalent to a CA forging a certificate in a traditional PKI and hence CL-PKC can provide trust level 3.

We give a general description to the CL-PKC algorithms as introduced by Al-Riyami and Paterson [1], which consist of five main algorithms: **Setup**, **Set-secret-Value**, **Partial-Private-Key-Extract**, **Set-Private-Key** and **Set-Public-Key** algorithms.

Let k be a security parameter given to the Setup algorithm and $\mathcal{G}$ be a Bilinear Diffie-Hellman Problem (BDH) parameter generator with input k.

1. **Setup (running by the KGC):** this algorithm runs as follows:
   (a) Run $I\mathcal{G}$ on input k to generate output $< G_1, G_2, e >$ where G_1 and G_2 are groups of some order q and $e: G_1 \to G_2$ is a pairing.
   (b) Choose an arbitrary generator $P \in G_1$.
   (c) Select a master-key s uniformly at random from Z_q^* and set $P_0 = sP$.
   (d) Choose cryptographic hash functions $H_1: \{0,1\}^* \to G_1^*$ and $H_2: G_2 \to \{0,1\}^n$, where n is the bit-length of plaintexts taken from some message space $M = \{0,1\}^n$ with a corresponding ciphertext space $C = G_1 \times \{0,1\}^n$.
   Then, the KGC publishes the system parameters $Params = < G_1,\ G_2,\ e,\ n,\ P,\ P_0,\ H_1,\ H_2 >$, while the secret master-key s is kept secure by the KGC.
2. **Set-Secret-Value (running by the user):** The inputs of this algorithm are *params* and entity m's identifier ID_m. It selects $x_m \in Z_q^*$ at random and output x_m as m's secret value. Then, entity m computes $X_m = x_m P$ and sends X_m to the KGC.
3. **Partial-Private-Key-Extract (running by the KGC):** The inputs of this algorithm are an identifier $ID_m \in \{0,1\}^*$ and X_m. The algorithm carries out the following steps to construct the partial private key for entity m with identifier ID_m .
   (a) Computes $Q_m = H_1(ID_m \,||X_m)$.
   (b) Outputs the partial private key $D_m = sQ_m \in G_1^*$.
   Entity m when armed with its partial private key D_m , it can verify the correctness of the partial private key D_m by checking $e(D_m, P) = e(Q_m, P_0)$.
4. **Set-Private-Key (running by the user):** The inputs of this algorithm are *params*, D_m (the partial private key of entity m) and $x_m \in Z_q^*$ (the secret value of entity m. It transforms the partial private key D_m to a private key S_m by computing $S_m = x_m D_m = x_m s Q_m \in G_1^*$.
5. **Set-Public-Key (running by the user):** The inputs of this algorithm are *params* and $x_m \in Z_q^*$, which is the secrete value of entity m. It then constructs the public key of identity m as $P_m = < X_m,\ Y_m >$, where $X_m = x_m P$ and $Y_m = x_m P_0 = x_m sP$.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The purpose of the binding technique that was used in Al-Riyami and Paterson [1] scheme, is to enforce the users to have one public/private key pairs in the system, and if there are two working public keys of any user, then it is an indication that the other key is generated by the KGC, which is equivalent to CA certificate forgery in traditional PKI. There are some modified schemes appeared in the literature from the original Al-Riyami and Paterson scheme [1], for example Mokhtarnameh et al. [19] proposed little modification on original scheme by setting the user's public key $P_m = x_m Q_m$, and they used this new public key in their proposed two party key agreement protocol in the same paper, Yang et al. [40] showed that the two party key agreement protocol that proposed by Mokhtarnameh et al. [19] is attackable by the man-in-the-middle attack and also explained that the Mokhtarnameh et al. [19] did not provide one-to-one correspondence between the user's identity and user' public key as they claimed, Mohammed et al [18] explained that Mokhtarnameh [19] and Yang et al. [40] schemes suffer from key escrow problem by showing that the KGC can compute the user's private key $S_m = s\, Y_m$ because the public key components $Y_m = x_m Q_m$.

2.3 Trust levels in public key infrastructures

The traditional CA in the PKI and KGC in the ID-BC and CL-PKC are all assumed to be trusted authorities. This trusted authority is considered the heart of the whole infrastructure, because it controls the system components and parameters, publishes the system parameters and the users' public keys, and in addition to that it might play a partial or a full role in generating the pairs of public and private keys of the users.

However, the assumption of the honesty of this authority can have severe consequences on the security of the whole infrastructure, if the third party is malicious. In such a case, it can carry out key replacement attacks on the users' public/private keys pairs. This motivated Gerault [22] to define three levels of trust between the user and the authority:

Trust level 1: the authority knows (or can easily compute) users' secret keys and therefore, can impersonate any user at any time with-out being detected (the KGC of the ID-PKC).

Trust level 2: the authority does not know users' secret keys, but it can still impersonate a user by generating false guarantees (CL-PKC).

Trust level 3: the authority cannot compute users' secret keys, and if it does so, it can be proven that it generates false guarantees (The CA in the traditional PKI).

3. THE GAP BETWEEN THE PKI AND CERTIFICATELESS PUBLIC KEY INFRASTRUCTURES

In this section, we present some comparative faces between the main three paradigms of public key management schemes: PKI, ID-BC and CL-PKC. The comparison factors are *maturity*, *standards*, *trust level*, *public key distribution mechanism*, *public key authentication mechanism* and *scalability overhead*. These factors are chosen to raise the strengths and weaknesses of each method.

The first factor maturity measures the reliability of the method based on the number and time it uses, and hence indicates how the method is efficient in achieving its stated goals. For example, the traditional PKI is used among the last 20 years in several and diverse applications around the world and has a good security impact on the web.

The second factor is standardization, the number of standard RFCs documents that made by some well-known standardization organizations like the IEEE and PKIX from the RSA Group.

The third factor is the Trust level, which measures the maximum trust level that can be achieved when applying the method. The trust level generally measures the degree of trust on the Key Generation Center (KGC) and as if it is high then the chance of the KGC to forge/escrow the users' public keys is low.

The fourth factor is the mechanism that is used to hold the public key like the digital certificate in the traditional PKI.

The fifth factor is the public key authentication mechanism, which explains how the public key of any given user is authenticated whether by digital signature, as in the traditional PKI or by a binding technique as in the CL-PKC or nothing as in the ID-BC.

Finally, Scalability and Key Management factor, which measures how the large number of users (and hence the number of public keys) can impact on the management overhead (by the KGC) of these keys. Table 1 summarizes the results of comparison

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

between PKI, ID-BC and CL-PKC according to the above mentioned five factors.

Table 1: Comparisons between the PKI, ID-BC and the CL-PKC

Method	PKI	ID-BC	CL-PKC
Maturity	Yes	No	No
Standards	RFC3820 RFC2560 RFC2510	RFC5091 RFC6509 RFC7859	None
Trust level	3	1	2-3
Public key distribution mechanism	X.509 Certificate	No standard format	No standard format
Public-key authentication mechanism	Digital Signature	None	Binding Technique
Scalability overhead	High	Low	Low

From Table 1, we can say that the traditional PKI is the most mature method with several standards of implementation and management but has one limitation, which is the high scalability overhead when the number of users in the system gets large. The processes of creating, distributing, validating, renewal and revoking of the digital certificate becomes inefficient and costly, because more intermediate CAs, large space certificate repository, very fast OCSP protocols and well-skilled IT staff are needed. On the other hand, the public keys in ID-BC are the public identities of the users in the system, then there is no need to store and distribute the public keys and hence the scalability is high and the overhead is low, but ID-BC cannot achieve trust level 3 and hence it suffers from key escrow critical security problem, which means that the KGC can forge any user to other users in the system and perform decryption/signature generation in behind of the user. The CL-PKC is an intermediate solution between the PKI and the ID-BC. It solves the key escrow problem of ID-BC and minimizes the management overhead of the PKI. The limitation of CL-PKC is the lack of RFC's standards.

4. DESIGN OF AN INTEGRATED CERTIFICATE LESS PUBLIC KEY INFRASTRUCTURE

The purpose of this section is to discuss how to fill the gap between the traditional PKI and the CL-PKC to design an integrated certificateless public key infrastructure based on the certificateless cryptography.

4.1 Designing a KGC with trust level 3

The binding technique of Al-Riyami and Paterson [1] qualified the KGC in their CL-PKC to have a trust level 2. This is because the KGC can still impersonate a user identity by replacing its pair of public/private keys without being detected. The security proof of the original scheme has been left as an open problem.

Most of the subsequent works (to Al-Riyami and Paterson) in CL-PKC found in the literature have not tried to introduce a system setup that can promote the KGC trust level to level 3. However, few researches in the literature focused in designing CL-PKC systems with KGCs of trust level 3, through improving the security model for the encryption or signature schemes. For example, in [41] Yang and Tan, the notion of key dependent certificateless encryption/signature scheme was introduced to provide a KGC with trust level 3 in the standard model. Another work was introduced by Li et al. in [23], who revealed that any provably secure conventional certificateless encryption (CLE)/signature (CLS) scheme with KGC trust level 2 can be transformed into the corresponding provably secure key dependent CLE/CLS scheme with KGC trust level 3.

A third approach for designing a KGC with trust level 3 was proposed by Hassouna et al. in [17]. They proposed modified public key cryptographic and binding schemes. The complete description of the model is as following:

Setup (running by the KGC): the KGC chooses a secret parameter k to generate G_1, G_2, P, e where G_1 and G_2 are two groups of a prime order q, P is a generator of G_1 and $e: G_1 \times G_1 \rightarrow G_2$ is a bilinear map. The KGC randomly generates the system's master key $s \in Z_q^*$ and computes the system public key $P_{pub} = sP$. Then the KGC chooses cryptographic hash functions H_1 and H_2 , where $H_1: \{0,1\}^* \times G_1 \rightarrow G_1$ and $H_2: \{0,1\}^* \rightarrow \{0,1\}^n$. Finally, the KGC publishes the system parameters $params = < G_1, G_2, e, P, P_{pub}, H_1, H_2, n >$, while the secret master-key is saved and secured by the KGC.

Set-Secret-Value (running by the user): a user m with the identity ID_m downloads the system

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parameters, picks two random secret values $x_m, x'_m \in Z_q^*$. Then, user m computes $X_m = x'_m P$ and sends X_m to the KGC. To provide two factor authentication and protecting the user's private key in case of device theft or compromise, the proposed scheme then enforce the user to choose a strong password *pass*, the system at client side hashes the password to be $z_m = H_2(pass)$, multiplies the base point P by the hashed password to be $z_m P$ (using special hash function to reserve the large size of the hashed value z_m to prevent brute-force attack on the point $z_m P$ and after that get the user's hashed password), use the hashed value z_m as key along with the *MAC* function to encrypt the secret value z_m as $MAC_{z_m}(x_m)$, sends copy of it to the KGC's public directory and stores a copy along with the point $z_m P$ locally. Note that there is no need to store the password $pass$ or its hash value z_m.

Partial-Private-Key-Extract (running by the KGC): on receiving X_m computed by user m with identity ID_m, the KGC first computes $Q_m = H_1(ID_m || X_m)$, then it generates the partial private key of user m as $D_m = sQ_m$. User m verifies his/her partial private key D_m by checking whether $e(D_m, P) = e(Q_m, P_0)$.

Set-Public-Key (running by the user): the user m with identity ID_m computes $Q_m = H_1(ID_m || X_m)$, $Y_m = x'_m Q_m$ and sets $< X_m, Y_m >$ as his/her long-term public key P_m. Finally, user m sends Y_m to the KGC.

Set-Private-Key (running by the user): every time the user needs to calculate and use his/her full private key, he/she enters his/her password, the system hashes it as z'_m, calculates $z'_m P$ and comparing it with stored $z_m P$, if it is equals then the password is correct and the user is authentic, hence it is used as a key to decrypt the stored $MAC_{z_m}(x_m)$. By extracting x_m the user becomes able to calculate the full private key by$(x_m + z_m)D_m$, otherwise the system aborts the process. We note that the private key must never be stored on the client and it shall be deleted after every usage.

4.2 Public key distribution in the certificateless PKI

To the best of our knowledge, only the work in Hassouna et al. [15] provides a mechanism for distributing the public keys in a certificateless PKI environment. The authors provided a PKI like structure, which contains three authorities for managing and distributing the public keys. The details of these authorities are as follows.

1. The Registration Authority(RA): It's function like in traditional PKI, the user may interact with this authority and fill in registration form, provides with personal information like names, address, national ID number and email address, after the RA authenticate the information of the user, it gives the user a unique random generated password for latter authentication purposes, in some cases the RA may give the user the system parameters which generated by the KGC server in a token or any electronic media.

2. The Key Generation Center (KGC): responsible for generating its master secret and the system parameters, keep its master secret in a secure storage and publish the system parameters in a public directory. KGC also has database that holds the user identities with their password hashed by any strong cryptographic hash function like MD5 or SHA-1.

3. The KGC's Public Directory(PD): it is a public directory system that consists of the KGC's public parameters, users' identities, users' partial private keys, users public key and other user parameters. It should be well controlled, updated only by the KGC, read only and accessible just by authenticated users.

In [15], the authors explained the exact function of each of the above components and gave several methods of authentication between the user and the KGC/PD.

4.3 Trust models of CL-PKI

It is possible to imbed the public keys of the users in X.509 certificates, which contains the signature of the KGC authority. This way, the CL-PKI enjoys all the features of the PKI, without suffering the scalability and key management problems. Therefore, all the trust models of the PKI can be inherited by the CL-PKI.

Hassouna et al. [17], introduced a hierarchal certificateless public cryptography scheme as an alternative to the traditional hierarchal PKI. They also described a hybrid scheme, in which clients in a PKI domain can communicate with clients in a CL-PKI domain through a bridge.

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. SECURITY MODELS

To measure the security claim of any cryptographic scheme/protocol, we need to state clearly two things: security notion and adversary power. The security notion is the security goals that must be satisfied after applying the cryptographic scheme/protocol. The adversary power determines the computational power of the attacker, i.e. the time that has have and oracles that can access (in the Random Oracle Model (ROM)). Since this chapter focuses on the CL-PKC schemes, only the security models that are related to the CL-PKC will be studied. Each type of CL-PKC scheme has its own security model, CL-PKE (Certificateless Encryption) has its own security model different than the security model that is designed especially for CL-PKC digital signature and so forth. Therefore, we will start by CL-PKC encryption's security models.

5.1 Security Models for CL-PKE

The security of a certificateless encryption scheme is expressed by two (very similar) games. In this section, we will describe a basic framework. In both cases, an attacker $A = (A_1, A_2)$ is trying to break the IND − CCA2 (Indistinguishability with Adaptive Chosen Ciphertext Attack) security of the scheme, the formal model describing confidentiality. The game runs as follows:

1. The challenger generates a master key pair $(mpk, msk) = Setup(1^k)$.
2. The attacker executes A_1 on mpk and (possibly) some extra information *aux*. During its execution A_1, it may have access to certain oracles (described subsequently). A_1 terminates by outputting an identity ID^*, two messages of equal length (m_0, m_1), and some state information *state*.
3. The challenger randomly chooses a bit $b \in \{0, 1\}$ and computes the challenge ciphertext $C^* = Encrypt(mpk, ID^*, pk_{ID^*}, m_b)$ using the value of pk_{ID^*} currently associated with the identity ID^*. If the public key pk_{ID^*} does not exist, then the challenger computes a public key pk_{ID^*} for ID^* by running the Set-Secret-Value and Set-Public-Key algorithms.
4. The attacker executes A_2 on the input (C^* , *state*). During its execution A_2 may have access to certain oracles (described subsequently). A_2 terminates by outputting a guess b' for b.

The attacker wins the game if $b' = b$ and its advantage is defined to be:

$$Adv_A = |\Pr(b = b') - 1/2|$$

The list of oracles that the attacker may have access to are:

- **Request Public Key:** The attacker supplies an identity ID and the challenger responds with the public key pk_{ID} for ID. If the identity ID has no associated public key, then the challenger generates a public key for ID by running Set-Public-Key and Set-Secret-Value (as necessary).
- **Replace Public Key:** This oracle models the attacker's ability to convince a legitimate sender to use an invalid public key. This can happen because public keys are no longer verified by a trusted third party, and a user may be given a false public key by an attacker and believe it to be correct. The attacker supplies an identity ID and a valid public key value $pk_{ID} \in PK$, and the challenger replaces the current public key value with the value pk_{ID}.
- **Extract Partial Private Key:** The attacker supplies an identity ID and the challenger responds with the partial private key d_{ID}. If the identity has no partial private key, then the challenger generates a partial private key by running Extract-Partial-Private-Key on ID using msk.
- **Extract Private Key:** The attacker supplies an identity ID and the challenger responds with the private key sk_{ID}. If the identity has no associated private key, then the challenger generates one using Set-Private-Key (after running the Set-Secret-Value algorithm and the Extract-Partial-Private-Key algorithm as necessary). An attacker may also have access to one or more different types of decryption oracle:
- **Strong Decrypt:** The attacker supplies an identity ID and a ciphertext c, and the challenger responds with the decryption of $c \in C$ under the private key s_{ID} . Note that if the attacker has replaced the public key for ID, then this oracle should return the correct decryption of c using the private key that inverts the public key pk_{ID} currently associated with the identity ID (or $\perp$ if no such private key exists).
- **Weak SV Decrypt:** The attacker supplies an identity ID, a secret value $x_{ID} \in S$, and a ciphertext $c \in C$. The challenger computes the full private key sk_{ID} for the identity from the (correct) partial private key d_{ID} and the supplied secret value x_{ID}, then returns the decryption of c under this private key sk_{ID}. If either process fails, then the oracle returns $\perp$. Note that this

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

functionality can be achieved by a strong decryption oracle.

- **Decrypt:** The attacker supplies an identity ID and a ciphertext $c \in C$, and the challenger responds with the decryption of c under the original private key sk_{ID} for ID. Note that this functionality can be achieved by a strong decryption oracle.

The Weak SV Decrypt oracle is so named as the attacker chooses the secret value, which is to be combined with the correct partial private key to give the full private key to be used for decryption.
A certificateless scheme is proven secure by showing that any attacker attempting to break the scheme only has a negligible chance of success.

Definition 4. Negligible Function: A function $f: N \rightarrow R$ is said to be negligible if, for every polynomial p, there exists an integer $N(p)$ such that $|f(x)| \leq 1/|p(x)|$ for all $x \in N(p)$.

5.1.1 Type I Attacker

The Type I security model is designed to protect against a third-party attacker (i.e. anyone except the legitimate receiver or the KGC), who is trying to gain some information about a message from its encryption form. There has been some debate about how to precisely formulate this notion and we survey the main attempts in this section.
Strong Type I Security is the original definition proposed by Al-Riyami and Paterson [3] as follows.

Definition 5. A certificateless encryption scheme is Strong Type I secure if every probabilistic, polynomial-time attacker $A^I = (A_1^I, A_2^I)$ has negligible advantage in winning the IND − CCA2 game subject to the following oracle constraints:

- A^I cannot extract the private key for the challenge identity ID^* at any time,
- A^I cannot extract the private key of any identity for which it has replaced the public key,
- A^I cannot extract the partial private key of ID^* if A I replaced the public key pk_{ID^*} before the challenge was issued,
- A_2^I cannot query the Strong Decrypt oracle on the challenge ciphertext C^* for the identity ID^* unless the public key pk_{ID^*} used to create the challenge ciphertext has been replaced,
- A^I cannot query the weak SV Decrypt or Decrypt oracles (although this functionality can be given by the Strong Decrypt oracle).

In this model, the attacker is given no extra information, i.e. *aux* is the empty bit-string. This model gives as much power as possible to the attacker.

It should be noted that the model expects the challenger to be able to correctly respond to decryption queries made on identities for which the attacker has replaced the public key. This is a very strong notion of security and it is unclear whether it represents a realistic attack scenario. In general, decryption oracles are provided to an attacker to model the fact that the attacker may be able to gain some information from a legitimate receiver about the decryptions of some ciphertexts.

This situation cannot happen if we replace a public key: when we replace a public key, we are duping a sender into encrypting a message using a false public key that the receiver has not published. Under no circumstances will the receiver then attempt to decrypt that ciphertext using the private key corresponding to that replaced public key. Hence, providing a decryption oracle that will accurately decrypt ciphertexts encrypted under the replaced public key gives the attacker more power than it would have in practice.

This represents an interesting philosophical question in the construction of security models: do we give the attacker as much power as is possible (perhaps subject to the restriction that we must still be able to construct secure certificateless encryption schemes)? Or should the model only try to reflect a realistic attacker's abilities? The former approach leads to strong security models, and potentially more complex schemes. The latter approach may lead to more efficient schemes, but a schemes security can only be guaranteed if an attacker's abilities have been correctly modelled.

Weak Type Ia Security Several authors have judged Al-Riyami and Paterson's Type I security model to be too strong and proposed weaker versions. We will consider each of the major alternatives in turn. The strongest of these definitions, which we will term Weak Type Ia security, has been put forward by Bentahar et al [10].

Definition 6. A certificateless encryption scheme is Weak Type Ia secure if every probabilistic, polynomial-time attacker A^I has negligible advantage in winning the IND−CCA2 game subject to the following oracle constraints:

- A^I cannot extract the private key for the challenge identity ID^* at any time,
- A^I cannot extract the private key of any identity for which it has replaced the public key,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- A^I cannot extract the partial private key of ID^* if A^I replaced the public key pk_{ID^*} before the challenge was issued,
- A^I cannot query the Strong Decrypt oracle at any time,
- A_2^I cannot query the Weak SV Decrypt oracle on the challenge ciphertext C^* for the identity ID^* if the attacker replaced the public key pk_{ID^*} before the challenge was issued.
- A_2^I cannot query the Decrypt oracle on the challenge ciphertext C^* for the identity ID^* unless the attacker replaced the public key pk_{ID^*} before the challenge was issued.

In this model, the attacker is given no extra information, i.e. *aux* is the empty bit-string. It should be noted that the original notion of Weak Type Ia security [10] did not give the attacker the ability to request decryptions using the original private key value after the public key had been replaced.

Weak Type Ib Security A weakening of this model gives Weak Type Ib security [31]:

Definition 7. A certificateless encryption scheme is Weak Type Ib secure if every probabilistic polynomial-time attacker A^I has negligible advantage in winning the IND−CCA2 game subject to the following oracle constraints:

- A^I cannot extract the private key for the challenge identity ID^* at any time,
- A^I cannot extract the private key of any identity for which it has replaced the public key,
- A^I cannot extract the partial private key of ID^* if A^I replaced the public key pk_{ID^*} before the challenge was issued,
- A^I cannot query the Strong Decrypt or Weak SV Decrypt oracles,
- A_2^I cannot query the Decrypt oracle on the challenge ciphertext C^* and the identity ID^* unless the attacker replaced the public key pk_{ID^*} before the challenge was issued.

In this model, the attacker is given no extra information, i.e. *aux* is the empty bit-string. In this model, the attacker can replace public keys (i.e. dupe senders) and can ask for decryptions of ciphertexts using the original private key values but cannot dupe a recipient into decrypting messages using a secret value chosen by the attacker. This reflects security in a situation where users generate their public key values correctly (i.e. by using the Set-Secret-Value and Set-Public-Key algorithms) and never change their public key values once they are set.

Weak Type Ic Security Lastly, mostly for comparison with Type II attackers, we present a final weak notion of security. This model of security was briefly considered in an early version of a paper by Baek and Wang [8]. This notion of security can be achieved by a public-key encryption scheme alone.

Definition 8. A certificateless encryption scheme is Weak Type Ic secure if every probabilistic, polynomial-time attacker A^I has negligible advantage in winning the IND − CCA2 game subject to the following oracle constraints:

- A^I cannot replace any public keys at any time,
- A^I cannot extract the private key for the challenge identity ID^* at any time,
- A^I cannot query the Strong Decrypt or Weak SV Decrypt oracles,
- A^I 2 cannot decrypt the challenge ciphertext C^* for the identity ID^* .

In this model, the attacker is given no extra information, i.e. *aux* is the empty bit-string.

5.1.2 Type II Attacker

The second security definition is designed to capture the notion that an honest-but-curious key generation center should not be able to break the confidentiality of the scheme. Here we allow the attacker to have access to master private key by setting *aux* = *msk*. This means that we do not have to give the attacker explicit access to an Extract Partial Private Key oracle, as they are able to compute these value for themselves. The most important point about Type II security is that the KGC can trivially break the scheme if it can replace the public key for the challenge identity before the challenge is issued.

Weak Type II Security Al-Riyami and Paterson [3] choose to prevent the trivial key replacement attack from occurring by forbidding the KGC from replacing any public keys at all, proposing the following model:

Definition 9. A certificateless encryption scheme is Weak Type II secure if every probabilistic, polynomial-time attacker $A^{II} = (A_1^{II}, A_2^{II})$, which is given the auxiliary information *aux* = *msk*, has negligible advantage in winning the IND−CCA2 game subject to the following oracle constraints:

- A^{II} cannot extract the private key for the challenge identity ID^* at any time,
- A^{II} cannot query the Extract Partial Private Key oracle at any time,
- A^{II} cannot replace public keys at any time,
- A^{II} cannot query the Strong Decrypt or Weak SV Decrypt oracles at any time,

- A_2^{II} cannot query the Decrypt oracle on the challenge ciphertext c and the identity ID^*.

This roughly corresponds to the weakest notion of security proposed for Type I attackers, and it is easy to see that any scheme that is Weak Type II secure is necessarily Weak Type Ic secure. Furthermore, this notion of security can be achieved by a public key encryption scheme alone, i.e. a scheme which contains no identity-based component and in which the user simply publishes a public key. In such a situation, it is easy to see that the above definition of Weak Type II security corresponds exactly to the "multi-user" definition of IND − CCA2 security.

Strong Type II Security The Weak Type II model prevents the attacker from replacing public keys. However, by denying the KGC the ability to replace public keys or query more powerful decryption oracles, we might be denying it the ability to perform certain attacks that might occur in practice, and we are certainly not providing it with the huge level of power provided to a Strong Type I attacker.

Hence, we should consider whether the KGC gains any advantages if we allow it to replace public keys (subject to the restriction that it cannot replace the public key of the challenge identity until after the challenge has been issued) or allow it access to more powerful decryption oracles.

Clearly, unless we permit the attacker to access a specialized decryption oracle, the ability to replace is public keys is useless to an attacker. This is because the attacker cannot replace the challenge public key before the challenge ciphertext is issued, hence, the challenger never gives the attacker any information based on a replaced public key value. The weak decryption oracle is of no use to an attacker because the attacker can always compute the full private key of a user given their identity ID and their secret value x_{ID} themselves.

Definition 10. A certificateless encryption scheme is Strong Type II secure if every II probabilistic, polynomial-time attacker $A^{II} = (A_1^{II}, A_2^{II})$, which is given the auxiliary information $aux = msk$, has negligible advantage in winning the IND−CCA2 game subject to the following oracle constraints:

- A^{II} cannot extract the private key for the challenge identity ID^* at any time,
- A^{II} cannot extract the private key of any identity for which it has replaced the public key,
- A^{II} cannot query the Extract Partial Private Key oracle at any time,
- A_1^{II} cannot output a challenge identity ID^* for which it has replaced the public key,
- A_2^{II} cannot query the Strong Decrypt oracle on the challenge ciphertext c for the identity ID^* unless the public key pk_{ID^*} used to create the challenge ciphertext has been replaced,
- A^{II} cannot query the Weak SV Decrypt or Decrypt oracles (although this functionality can be given by the Strong Decrypt oracle).

Clearly, unless we permit the attacker to access a specialized decryption oracle, the ability to replace is public keys is useless to an attacker. This is because the attacker cannot replace the challenge public key before the challenge cipher-text is issued; hence, the challenger never gives the attacker any information based on a replaced public key value. The weak decryption oracle is of no use to an attacker because the attacker can always compute the full private key of a user given their identity ID and their secret value x_{ID} themselves. Hence, all the Weak Type II security models that we might propose (based on the Weak Type I security models) are equivalent.

Definition 11. A certificateless encryption scheme is Strong Type II secure if every II probabilistic, polynomial-time attacker $A^{II} = (A_1^{II}, A_2^{II})$, which is given the auxiliary information $aux = msk$, has negligible advantage in winning the IND − CCA2 game subject to the following oracle constraints:

- A^{II} cannot extract the private key for the challenge identity ID^* at any time,
- A^{II} cannot extract the private key of any identity for which it has replaced the public key,
- A^{II} cannot query the Extract Partial Private Key oracle at any time,
- A_1^{II} cannot output a challenge identity ID^*, for which it has replaced the public key,
- A_2^{II} cannot query the Strong Decrypt oracle on the challenge ciphertext C for the identity ID^* unless the public key pk_{ID^*} used to create the challenge ciphertext has been replaced,
- A^{II} cannot query the Weak SV Decrypt or Decrypt oracles (although this functionality can be given by the Strong Decrypt oracle).

5.2 Security Models for Certificateless Digital Signature (CL-DS)

As defined in Al-Riyami and Paterson [1], there are two types of adversary/attacker against the CL-DS with different capabilities:

• **Type I Attacker:** This type of adversary A^I does not have access to the master-key, but it has the ability to replace the public key of any entity with a

value of his choice, because there is no certificate involved in certificateless signature schemes.

• **Type II Attacker:** This type of adversary A^{II} has access to the master-key but cannot perform public keys replacement.

Nevertheless, no formal security model was presented in neither [1] nor [2]. In this section, firstly we provide a formal definition of existential unforgeability of a certificateless signature (CLS) scheme under both two types of chosen message attack. They are defined using the following game between an adversary $A \in \{A^I, A^{II}\}$ and a challenger C.

5.2.1 Type I Attacker

- **Setup:** C runs the algorithm to obtain the system parameter lists *params*, C then sends *params* to the adversary A^I.
- **Partial-Private-Key Queries:** A^I can request the Partial-Private-Key of the user, whose identity is ID. In respond, C outputs the Partial-Private-Key d_{ID}.
- **Public-Key-Replacement:** For any user whose identity is ID, A^I can choose a new Secret-Value x and compute the new public key (X, Y). A^I then sets (X, Y) the new public key of this user and submits (x, X, Y, ID) to C. Then, C will record these replacements which will be used later.
- **Sign Queries:** A^I can request user's (whose identity is ID) signature on a message M. In respond, C outputs a signature for the message M, which is a valid signature under the public key A^I has replaced earlier.
- **Output:** Finally, A^I outputs a target message/signature pair (M^*, σ^*) of the user whose identity is ID^*. This message/signature pair must satisfy the following requirements:
  1. This signature is valid under the public key (X^*, Y^*) chosen by A^I .
  2. A^I does not request the Partial-Private-Key of this user whose identity is ID^*.
  3. M^* has never been queried during the Sign Queries.

The success probability of an Type I adversary to win the game is defined by:

$$Succ_{A^I}^{EF-CMA}$$

Where *EF-CMS* means Existential Forgeable with Chosen Message Attacker.

Definition 12. certificateless signature scheme is existential unforgeable against Type I chosen-message attacks if and only if the probability of success of any polynomial bounded Type I adversary in the above game is negligible. In other words, $Succ_{A^I}^{EF-CMA}(k) \leq \varepsilon$, where k is the system's security parameter.

5.2.2 Type II Attacker

- **Setup:** C runs the algorithm to obtain the system parameter lists params and also the system's master-key: s, then C sends *params* and s to the adversary A^{II}.
- **Sign Queries:** A^{II} can request user's (whose identity is ID) signature on a message M . In respond, C outputs a signature σ for a message M.
- **Output:** Finally, A^{II} outputs a target message/signature pair (M^*, σ^*) of the user, whose identity is ID. This message/signature pair must satisfy the following requirements:
  1. This signature is a valid one, i.e. it passes the verification algorithm.
  2. M^* has never been queried during the Sign Queries.

The success probability of an Type II adversary to win the game is defined by:

$$Succ_{A^{II}}^{EF-CMA}$$

Definition 13. A certificateless signature scheme is existential unforgeable against Type II chosen-message attacks iff the probability of success of any polynomial bounded Type II adversary in the above game is negligible. In other words $Succ_{A^{II}}^{EF-CMA}(k) \leq \varepsilon$, where k is the systems security parameter.

Definition 14. A certificateless signature scheme is existential unforgeable against chosen-message attacks iff it is secure against both types of adversaries.

This way, almost all the digital signature schemes in the literature are built and proved to be secure against these two types of adversary with the help of hardness of known cryptographic primitives such as Computation Diffie-Hellman Problem (CDHP), Decisional Diffie-Hellman Problem (DDHP) and Bilinear Diffie-Hellman Problem (BDHP).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. CONCLUSIONS AND REMARKS

This paper addressed the weaknesses of the traditional public key infrastructure (scalability and certificate management) and the identity-based encryption (key escrow problem) and pointed to the certificateless cryptography as a solution to these problems. But since the certificateless cryptography is yet immature to become an infrastructure, it was necessary for us to look at the gaps between the PKI and CL-PKC. The PKI is a complete infrastructure to provide all the security services, whereas the CL-PKC is not built upon an infrastructure, which provides the authenticity and non-repudiation due to the problem of carrying out key replacement by a malicious KGC in case of level 2 trust level.

To fill this gap, we pointed to two important tasks that shall be established first. The first task is to design a KGC with trust level 3 as found in the PKI. The second task is to design an infrastructure for the certificateless cryptography, similar to the PKI, where the KGC with trust level 3 replaces the CA, and a public directory is used for the public key distribution.

It is possible to imbed the public keys of the users in X.509 certificates, which contains the signature of the KGC authority. This way, the CL-PKI enjoys all the features of the PKI, without suffering the scalability and key management problems. Therefore, all the trust models of the PKI can be inherited by the CL-PKI.

REFERENCES:

[1] S. Al-Riyami and K. Paterson, Certificateless public key cryptography, *in Asiacrypt 2003, ser. Lecture Notes in Computer Science, C. Laih, Ed.*, 2003, pp. 452–473, full version available at Cryptology ePrint Archive.

[2] M. O. Albasheer and E. Bashier, Enhanced Model for PKI Certificate Validation in The Mobile Banking, *in 2013 International conference on computing, electrical and electronic engineering (ICCEEE), Khartoum, Sudan*, 470-476, 2013.

[3] P. Barreto, H. Kim, B. Lynn, , and M. Scott, Efficient algorithms for pairing-based cryptosystems, *In Advances in Cryptology(CRYPTO2002 of LNCS, Springer-Verlag*), 2442 , 2002, pp. 354–368.

[4] P. Barreto, B. Lynn, and M. Scott, Constructing elliptic curves with prescribed embedding degrees, *In Security in communication networks(SCN'2002 of LNCS, Springer-Verlag)*, 2576, 2002, pp. 263–273.

[5] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, pp. 586–615, 2003.

[6] D. Boneh, H. Shacham, and B. Lynn, Short signatures from the Weil pairing, *In C. Boyd, editor, Advances in Cryptology(ASIACRYPT 2001, volume 2248 of LNCS, Springer-Verlag)*, 2001, pp. 514–532.

[7] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *in Advances in Cryptology(CRYPTO 2001, LNCS, Springer-Verlag)*, 2139, 2001, pp. 213–229.

[8] A. W. Dent, B. Libert, and K. G. Paterson, Certificateless encryption schemes strongly secure in the standard model, *in Public Key Cryptography*, 2008, pp. 344–359.

[9] R. Dupont, A. Enge, and F. Morain, Building curves with arbitrary small mov degree over finite prime fields, 2002.

[10] S. Galbraith, K. Harrison, and D. Soldera, Implementing the tate pairing, *In Algorithmic Number Theory 5th International Symposium (ANTS-V, volume 2369 of LNCS, Springer-Verlag)*, 2002, pp. 324–337.

[11] S. Galbraith, Supersingular curves in cryptography, *In C. Boyd, editor, Advances in Cryptology(ASIACRYPT 2001, volume 2248 of LNCS, Springer-Verlag)*, 2001, pp. 495–513.

[12] J.C. Cha and J.H. Cheon, An identity-based signature from gap diehellman groups, *in Public Key Cryptography - PKC 2003, Y. Desmedt, Ed.*, 2567, 2002, pp. 18–30.

[13] F. Hess, Efficient identity-based signature schemes based on pairings, *in Selected Areas in Cryptography 9th Annual International Workshop*, K. Nyberg and H. Heys, Eds., 2595, 2003, pp. 310–324.

[14] S. MICALI, Novomodo : Scalable certificate validation and simplified pki management, *in Proc. 1st Annual PKI Research Workshop (2002)*, 2002, pp. 15–25.

[15] M. Hassouna, B. Barri, N. Mohamed, and E. Bashier, An integrated public key infrastructure model based on certificateless cryptography, *International Journal of Computer Science and Information Security*, 11(11), pp. 1–10, 2013.

[16] M. Hassouna, E. Bashier, and B. Barry, A short certificateless digital signature scheme, *In International Conference of Digital Information Processing, Data Mining and Wireless Communications*, pp. 120–127, 2015.

[17] M. Hassouna, E. Bashier, and B. Barry, A new level 3 trust hierarchal certificateless public key

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

cryptography scheme in the random oracle model, *International Journal of Information Security*, 19(4), 551—558, 2017.

[18] N. Mohamed, M. Hassouna, and E. Bashier, A secure and efficient key agreement protocol based on certificateless cryptography, *International Journal of Intelligent Computing Research (IJICR)*, 3, 2012.

[19] R. Mokhtarnameh, S. Ho, and N. Muthuvelu, An enhanced certificateless authenticated key agreement protocol, *in Proc. of the 13th International Conference on Advanced Communication Technology(ICACT)*, 2011, pp. 802–806.

[20] C. Gentry and A. Silverberg, Heirarchical ID-based cryptography, *In ASIACRYPT 2002*, 2501, 2002, pp. 548–566.

[21] L. Chen, K. Harrison, A. Moss, D. Soldera, and N. Smart, Certification of public keys within an identity based system, *in Information Security*, 5th International Conference, 2433, 2002, pp. 322–333.

[22] M. Girault, Self-certified public keys, *In Advances in Cryptology (EUROCRYPT'91), LNCS* 547, pp. 490–497, Springer, 1992.

[23] V. N. Krishna, A. H. Narayana, K. M. Vani, Window method based cubic spline curve public key cryptography, *International Journal of Electronics and Information Engineering*, 4(2), pp. 94–102, 2016.

[24] Li F., Gao W., Xie D., Tang C. Certificateless Cryptography with KGC Trust Level 3 Revisited. *In: Sun X., Chao HC., You X., Bertino E. (eds) Cloud Computing and Security. ICCCS 2017. Lecture Notes in Computer Science*, 10603, pp 292-304, 2017.

[25] J. K. Liu, M. H. Au, and W. Susilo, Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model, *in Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pp. 273–283, 2007.

[26] R. Mokhtarnameh, S. Ho, and N. Muthuvelu, An enhanced certificateless authenticated key agreement protocol, *in Proceedings of the 13th International Conference on Advanced Communication Technology*, pp. 802–806, 2011.

[27] K. Paterson, Cryptography from pairings: a snapshot of current research, *Information Security Technical Report*, 3, pp. 41–54, 2002.

[28] K.G. Paterson, Id-based signatures from pairings on elliptic curves, *Electronics Letters*, 18, pp. 1025–1026, 2002.

[29] A. Shamir, Identity-based cryptosystems and signature schemes, *In Advances in Cryptology-CRYPTO'84*, 196, 1984, pp. 47–53.

[30] R. Sakai, K. Ohgishi, and M. Kasahara, Cryptosystems based on pairing, In The 2000 Symposium on Cryptography and Information Security, 2000.

[31] N. P. Smart, An identity based authenticated key agreement protocol based on the Weil pairing, *Electronics Letters*, 13, 2002.

[32] N.P. Smart, Access control using pairing based cryptography, *in Proceedings CT-RSA 2003*, 2612, 2003, pp. 111–121.

[33] S. S. D. Selvi, S. S. Vivek, and C. P. Rangan, CCA2 secure certificateless encryption schemes based on rsa, *IACR Cryptology ePrint Archive*, 2010, p. 459, 2010.

[34] H. Xiong, Z. Qin, and F. Li, An improved certificateless signature scheme secure in the standard model, *Fundamenta Informaticae*, 88, 2008.

[35] L. Zhang and F. Zhang, A new provably secure certificateless signature scheme, *in 08 IEEE International Conference on Communications*, 2008, pp. 1685–1689.

[36] S. S. D. Selvi, S. S. Vivek, and C. P. Rangan, Certificateless kem and hybrid signcryption schemes revisited, *in ISPEC*, 2010, pp. 294–307.

[37] S. S. Vivek, S. S. D. Selvi, C. P. Rangan, CCA2 secure certificateless encryption schemes based on RSA, *In Proceedings of the International Conference on Security and Cryptography (SECRYPT'10)*, pp. 208–217, 2011.

[38] C. Wang, D. Long, and Y. Tang, An efficient certificateless signature from pairing, *International Journal of Network Security*, 8(1), pp. 96–100, 2009.

[39] W. Xie and Z. Zhang, Certificateless signcryption without pairing, *IACR Cryptology ePrint Archive*, 2010, p. 187, 2010.

[40] W. Xie and Z. Zhang, Efficient and provably secure certificateless signcryption from bilinear maps, *in WCNIS*, 2010, pp. 558–562.

[41] H. Yang, Y. Zhang, and Y. Zhou, An improved certificateless authenticated key agreement protocol, *Cryptology* ePrint Archive, Report 2011/653, 2011, http://eprint.iacr.org/.

[42] G. Yang and C. H. Tan, Certificateless cryptography with KGC trust level 3, *Theoretical Computer Science*, 412(39), 2011, pp. 5446-5457

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[43] Y. Yuan and C. Wang, A secure certificateless signature scheme in the standard model, *Journal of Computational Information Systems*, 9(11), pp. 4353–4362, 2013.

[44] L. Zhang and F Zhang, A new provably secure certificateless signature scheme, *in IEEE International Conference on Communications*, pp. 1685–1689, 2008.