Indian J.Sci.Res. 17(2): 503-509, 2018

ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

A NEW NO CERTIFICATE CRYPTO-ENCRYPTION TECHNIQUE IN DYNAMIC ORACLE MODEL

[1]R Swetha, [2]T Malathi

[1]Department of Computer Science & Engineering, St. Martins Engineering College, Dhulapally, Hyderabad

[2] Department of Computer Science & Engineering, Aurora's Scientific, Technological and Research Academy, Chandrayangutta, Hyderabad

Abstract: **-**Network intrusion is a critical challenge in information and communication systems amongst other forms of fraud perpetrated over the Internet. In IDS the dynamic oracle model is a model used in cryptographic security proofs, in which concrete primitives such as hash functions are replaced with a "dynamic oracle": a hypothetical black box that maps its inputs to truly dynamic outputs, but does it in such a way that the same input always yields the same output. No certificate effective key management was used for the purpose of creating secure pairwise node communication and group oriented key communication within clusters. Despite the fact that the traditional public key infrastructure provides Level 3 trusted authority, but its two major problems of scalability and certificate management raised the need to an alternative security infrastructure. That motivated the appearance of new technologies to replace the traditional PKI, such as the Identity based encryption, the no certificate encryption, etc. But all those new technologies are yet immature and could not introduce a trust level more than Level 2, except few trials at the level of the authority. This paper aims at introducing an integrated hierarchal no certificate scheme with a Level 3 trust authority. This is done through merging the traditional PKI hierarchy and the no certificate technology in one scheme. The new scheme employs the X509 certificate format and is free of the scalability and certificate management problems of the PKI model.This key management procedure was used to secure the node when it move across different clusters and key revocation process for compromised node. It also protect against the various attacks when the sensor node gets communicated by maintaining the key efficiently provided.

Keywords: No certificate cryptography, public key infrastructure, dynamic oracle model, security services, trust levels

I. Introduction

Public Key Infrastructure (PKI) is a complete system to manage the public keys in any public key cryptography-based application using the concept of digital certificates. The PKI provides authentication of system users by allowing some trusted third-party to sign the public key of any entity in the system. In the context of PKI, any entity in the system can verify the authentication of any other entity by verifying its signed certificate using the trusted third-party's public key. In this way, any other crypto graphic services (like confidentiality and non-repudiation) can be achieved and implemented.

Furthermore, PKI has some well-established trust models that meet the organization and requirements. Examples of these trust models are hierarchal and bridge models. When the system scale gets large, the number of signed digital certificates also gets large. Therefore the overhead of the management of these certificate increases. Moreover, other issues like public key

II. Literature Survey

Revocation and its related notification methods are raised. However, in spite of the maturity of the PKI and its wide applications and usage, the PKI has main two challenges. These challenges are scalability and certificate management [1, 10].Some other paradigms of public key cryptography are introduced to overcome the PKI challenges and simplifying the key management. Identity-based Public Key Cryptography (ID-PKC) (which was invented by Boneh and Franklin [2]) and Certificate less Public Key Cryptography (CL-PKC) (invented in 2003 by Al-Ryami and Paterson [1]) are such examples to these paradigms. The CL-PKC addressed the key-escrow problem of the ID-PKC [1] and provided a lightweight infrastructure for managing the public keys of the users in the system with-out using the digital certificates. Since the original Al-Ryami and Paterson scheme [1], many certificate less encryption schemes [3, 11]certificate less digital signature schemes and certificate less key agreement protocols [5, 12] were appeared in the literature.

Some other paradigms of public key cryptography are introduced to overcome the PKI challenges and simplifying the key management. Identity-based Public Key Cryptography (ID-PKC) (which was invented by Boneh and Franklin [2]) and Certificate less Public Key Cryptography (CL-PKC) (invented in 2003 by Al-Ryami and Paterson [1]) are such examples to these paradigms. The CL-PKC addressed the key-

Furthermore, PKI has some well-established trust mod-els that meet the organization and requirements. Examples of these trust models are hierarchal and bridge models. When the system scale gets large, the number of signed digital

[1]**Corresponding Author**

certificates also gets large. Therefore the overhead of the management of these certificate increases. Moreover, other issues like public key revocation and its related notification methods are raised. However, in spite of the maturity of the PKI and its wide applications and usage, the PKI has main two challenges. These challenges are scalability and certificate management [1, 10].Some other paradigms of public key cryptography are introduced to overcome the PKI challenges and simplifying the key management. Identity-based Public Key Cryptography (ID-PKC) (which was invented by Boneh and Franklin [2]) and Certificate less Public Key Cryptography.The pairs of public and private keys of the users. If this third party is malicious, then the security of the whole infrastructure could be compromised. For this, Girualt [6] three levels of trust: At Level 1 trust, 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). At Level 2 trust, the authority does not know users' secret keys, but it can still impersonate a user by generating false guarantees (CL-PKC). At 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).

In 2013 Hassouna et al. [7] proposed an integrated Certificate less public key infrastructure model (CL-PKI). In their model, a different method for generating entity key pair has been introduced. Furthermore, Hassouna et al. [7] incorporated a different binding technique to link the entity's identity with its corresponding keys to en-sure the uniqueness of the key pair. The direct security and management advantages of using this method of key generation are two-factor private key authentication, private key portability, private key recovery and private key archiving [7]. Moreover, Hassouna et al. extended their CL-PKI model by proposing a new security model for certificate less digital signature schemes. Then, they pro-posed a strong and efficient provable secure certificate less digital signature scheme [8] in the Random Oracle Model (ROM) without stating its security proof. Recently, Hassouna et al. [9] stated the complete security proof of the digital signature scheme in the random oracle model [8].

In this paper, we propose a Hierarchal Certificate less Public Key Cryptography Scheme (HCL-PKC) and then use it to construct a Hybrid PKI/CL-PKI scheme. These two schemes are introduced in the context of Hassouna et al.'s Cl-PKI model, hence they enjoy the security proper-ties and key management features of Hassouna et al.'s [7] model.

The rest of this paper is organized as follows. We state Hassouna et al.'s [7] CL-PKI model in Section 2. Hassouna et al.'s [8] digital signature scheme is given in Section 3. In Section 4, we introduce the proposed Hierarchal Certificate less Public Key Cryptography Scheme (HCL-PKC). In Section 5, we give the Hybrid PKI/CL-PKI scheme. Finally, Section 6 concludes the paper.Hassouna et al.'s Certificate-less Public Key Infrastructure Model (CL-PKI)

As stated in [7]: to make the CL-PKC schemes suitable for practical applications, there is a need for some sort of infrastructure as the traditional PKI. Therefore, Hassouna et al. [7] proposed a CL-PKI model with three components: Registration Authority (RA), Key Generation Centre (KGC) and Public Directory (PD).

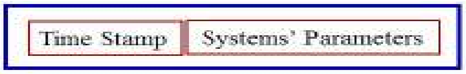
The components of the proposed CL-PKI and their functions are as follows:

He/She then generates two random secret values x_m; x^0_m 2 Z_q. Then, it computes $X_m = x^0_m P$ and sends X_m to the KGC. To provide two factor of authentication and protection for the user's private key against the device theft or compromise, the pro-posed scheme enforces the user to choose a strongPassword pass. The client device uses the hash function H_2 to generate $z_m = H_2(pass)$ and multiplies the base point P by the hashed password to get $z_m P$. The hash function H_2 must be capable to preserve the large size of the hashed value z_m to prevent the brute-force attack on the point $z_m P$. It then uses the hashed value z_m as key along with the M AC function to encrypt the secret value x_m as $M\ AC_{zm}$ (x_m)and sends a copy to the KGC's public directory to be stored together with the point $z_m P$ locally. It is worthy to notice that there is no need to store the password pass or its hash value z_m.Partial-Private-Key-Extract (running by the KGC): When the KGC receives X_m from a user m with an identity ID_m, the KGC first computes $Q_m = H_1(ID_m jj X_m)$, then it generates the partial private key of user m as $D_m = sQ_m$. User m can verify the correctness of his/her partial private key D_m, through testing whether $e(D_m; P\ )$

The Registration Authority (RA): The registration authority plays the same role as the registration authority of the traditional PKI. The user might interact with this authority and provides proofs of his personal information like names, address, national ID number and email address. After the information of the user, it gives the user a unique random generated password for latter authentication purposes, in addition to the system parameters, generated by the KGC server in a token or any electronic media.

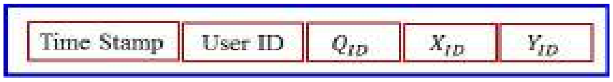
The Key Generation Centre (KGC): The KGC is responsible of generating its master secret and the system parameters. It has to keep it's master secret in a secure storage and publish the system parameters in a public directory. The KGC also has a database that holds the user identities with their password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

The KGC's Public Directory (PD): The public directory is responsible of storing the KGCs' pub-lic parameters, users identities, users partial private keys, users public key and other user parameters. It is controlled and updated by the KGC. The con-tents of the PD are available for only the authenticated users, who do not have the right to write in it. The typical format of the public directory records are given in Figure 1 and Figure 2, respectively.



Time Stamp	Systems' Parameters

Figure 1: Systems' parameters record



Time Stamp	User ID	Q_{ID}	X_{ID}	Y_{ID}

Figure 2: Contents of the public directory of a user

Typically the RA has connection with the KGC. When the KGC generates the user's password at the registration time, the RA passes it to the user without knowing it.

In [7], Hassouna et al. introduced several methods of authentication between the user and the KGC/PD. The complete description of the model is as following:

Set-Secret-Value (running by the user): A user m with an identity ID_mdownloads the system parameters

$e(Q_m; P_0)$.Set-Public-Key (running by the user): The user m whose identity is ID_m computes $Q_m = H_1(ID_mjjX_m)$, $Y_m = x^0{}_mQ_m$ and sets $<X_m; Y_m>$ as his/her long-term public key P_m. Finally, user m sends Y_m to the KGC.

III. Hassouna et al.'s Certificate less Digital Signature Scheme

In this section, we provide details on the certificate less digital signature scheme that was proposed by Hassouna et al. and its functionality [8].

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_1G_1 ! G_2 is a bilinear map. The KGC randomly generates the system's master key s 2 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 : f0; 1g ! G_1 (Map-to-Point hash function), and H_2 : f0; 1g ^{n} ! Z_q (any crypto-graphic hash function like MD5 or SHA family). 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 an identity ID_m downloads the sys-tem parameters, generates two random secret values x_m; $x^0{}_m$ 2 Z_q. Then, user m computes $X_m = x^0{}_mP$ and sends X_m to the KGC.

The proposed scheme en-forces the user to choose a strong password pass, the system at the client side hashes the password to be $z_m = H_2(pass)$, multiplies the base point P by the hashed password to be z_mP , uses the hashed value z_m as key to encrypt the secret value x_m and generates the Password-based Encryption Code (PEC) as P EC_{zm} (x_m), sends a copy of it to the KGC's pub-lic directory and stores it along with the point z_mP locally.

Partial-Private-Key-Extract (running by the KGC): On receiving X_m computed by user m with identity ID_m, the KGC rst computes $Q_m = H_1(ID_m)$, then it generates the partial private key of user m as $D_m = sQ_m$.

Set-Public-Key (running by the user): The user m with identity ID_m computes $Q_m = H_1(ID_m)$, $Y_m = x^0{}_mQ_m$ 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: User m's private key is $S_m = (x_m + z_m)D_m = (x_m + z_m)sQ_m = (x_m + z_m)sH_1(ID_m)$. Also, the user generates the secret term $Z_m = x_mP$.

Sign: The user generates the signature of the mes-sage M using his secret terms fx_m; Z_mg as follows:

The signer generates a big random integer a 2

G_2.

The signer calculates M $P_m = H_1(m)$ 2 G_1.

The signer calculates M $P_{1m} = ax_mM$ P_m 2 G_1.

The signer calculates $s_m = e(M\ P_m; Z_m)^{ax0}m = e(M\ Pm; P)^{ax}m^{x0}m$.

The signer sends = (m; M P_{1m}; s_m) as the sig-nature.

Verify: After receiving the signature = (m; M P_{1m}; s_m), the verier uses the public key $<X_m; Y_m>$ of user m to verify the signature as fol-lows:

The verier checks whether $e(X_m; Q_m) = e(Y_m; P\ )$. If it holds then user m's public key is authenticated, otherwise the signature is re-jected.

The verier calculates M $P_m{}^0 = H_1(m)$ 2 G_1.

If M P_{1m} = M $P_m{}^0$ or $s_m = e(H_1(m); X_m)$ then the verier rejects the signature. Otherwise, the verier calculates $r_m = e(M\ P_{1m}; X_m)$.

The verier accepts the signature i$r_m = s_m$, otherwise he/she rejects the signature.

Hassouna et al.'sKerberos Security Model

The modern Kerberos has undergone several major re-visions. In each review, significant improvements have been made like scalability and security. The version 1 through 3 were used internally and as to version 4 was the rst version distributed to the public was Kerberos V4, which has been limited in some nations due to the limitations of used encryption algorithms. These limitations made norms to evolve a new protocol that contains all the features presented in the Kerberos V4, with the addition of features such as extensible encryption types and more transparent authentication to create the version 5 of Keberos [12].After all these changes and with the development of computer system, Kerberos V5 still vulnerable against attacks such as attacks by brute force and dictionary.

They still represent a real challenge for this protocol. These conclusions made thinking several researchers to propose solutions such as the use of asym-metric cryptographic primitives [10], in order to make the keys generation more reliable, or the introducing of new technologies such as smart card. In this section, we present the communication phase based on two strong points: cryptographic primitives and tickets, and the various requests exchanged between a client and the KDC server to access a service.

Hassouna et al.'s Security Model

In Hassouna et al. [8] two types of adversaries were considered: Type I and Type II adversaries according to the term Z_m as follows:

Type I Adversary A_I : This adversary is allowed to replace the term Z_m by a valid value of his choice, but is not allowed to replace users' public keys and has not access to the master secret key s.

Type II Adversary A_{II} : This adversary has an ac-cess to the master secret key s, and is allowed to replace users public keys with valid values of his choice, but is not allowed to replace the term Z_m.

Type I adversary represents an outsider attacker and typeattacker is a malicious KGC. The first game is performed between a challenger C and a Type I adversary A_I as follows.

Setup. The challenger C runs Setup algorithm and generates a master secret key msk and pub lic system parameters params. C gives params to A_I , while keeping msk secret. Queries. A_I may adaptively issue the following queries to C.{ Partial private key queries: Upon receiving a partial private key query for an identityID, C returns the partial private key with respect to identity ID to A_I .

{Public key queries: Given an identity ID, C returns the corresponding public key terms < X_A; Y_A> to A_I .

{Replace public key: Given an identity ID with a pair of values ($x^{0}_{ID}{}^{1}$; $pk_{ID}{}^{1}$) which are chosen by A_I , C updates the user ID orig-inal secret/public key (x^{0}_{ID}; pk_{ID}) to the new ($x^{0}_{ID}{}^{1}$; $pk_{ID}{}^{1}$).

{Z key Extraction queries: This is a new oracle in this security model, given an identity ID, C returns the corresponding Z key value Z_{ID}.

{Replace Z key: This is a new ora-cle in this security model which on input (ID; x^{1}_{ID}; $Z_{ID}{}^{1}$), C replaces the user ID original term (x_{ID}; Z_{ID}) by (x^{1}_{ID}; $Z_{ID}{}^{1}$).

{Private key queries. Upon receiving a private key query for an identity ID, C returns the corresponding private key sk_{ID} to A_I .

{Sign queries: Proceeding adaptively, A_I can request signatures on any messages m withrespect to an identity ID. C computes sig-nature, and returns to A_I .

Forgery. Eventually, A_I outputs a certificateless signature on message m cor-responding to public key pk_{ID} for anidentityID . A_I wins the game if Verify(params; ID ; pk_{ID} ; m ;) = 1 and the following conditions hold {A_I has never been queried Partial private key oracle on ID .

The success probability of A_I is defined as the probability that it wins in game I.Game II. This game is performed between a challenger C and a Type II adversary A_{II} as follows.

Setup. The challenger C runs A_{II} on k and a special Setup, and returns a master secret keymsk and public system parameters paramstoA$_{II}$.

Queries. In this phase, A_{II} can adaptively ac-cess the Private key oracle, Public key oracle, Replace public key oracle, Z key oracle, Re-place Z key oracle and Sign oracle, which are the same as that in Game

Forgery. A_{II} outputs a certificate less signature on message m corresponding to public key pk_{ID} for an identity ID . A_{II} wins the game if Verify(params; ID ; pk_{ID} ; m ;) = 1 and the following conditions hold:

{ A_{II} has never been queried Private key or-acle on ID .

{ A_{II} has never been queried Replace Z key oracle on ID .

{ A_{II} has never been queried Signature oracle on (ID ; m).

The success probability of A_{II} is defined as the probability that it wins in Game II

Accordingly, the security of any certificate less digital signature scheme in the Random Oracle Model (ROM) can be given as follows.

Definition 1. A certificate less signature scheme is (t; q_H ;q_e; q_z; q_{sk}; q_{pk}; q_s;)-existentially unforgeable against Type I adversary under adaptively chosen mes-sage attacks if no t-time adversary A_I , making at most q_H to the random oracles, q_e partial private key queries, q_z to the Z key queries, q_{sk} private key queries, q_{pk} public key queries and q_s signature queries, have a success probability at least in Game I.

Definition 2. A certificate less signature scheme is (t; q_H ;q_z; q_{sk}; q_{pk}; q_s;)-existentially unforgivable against Type II adversary under adaptively chosen message at-tacks if no t-time adversary A_{II} , making at most q_H to the random oracles, q_z to the Z key queries, q_{sk}private key queries, q_{pk} public key queries and q_s signature queries, have a success probability at least in Game II

Definition 3. A certificate less signature scheme is existentiallyunforgeable under adaptively chosen message attack (EUF-CMA), if the success probability of any polynomially bounded adversary in the above two games is negligible.

Theorem 1.Hassouna et al.'s [8] digital signature scheme is secure against existential forgery under adap-tively chosen message attacks in the random oracle model with the assumptions that CDHP(Computation Diffie-Hellman Problem) and BDHP (Bilinear Diffie-Hellman Problem) in G_1 are intractable.

Setup. The challenger C runs A_{II} on k and a special Setup, and returns a master secret keymsk and public system parameters paramstoA_{II} .

Queries. In this phase, A_{II} can adaptively ac-cess the Private key oracle, Public key oracle,

Replace public key oracle, Z key oracle, Re-place Z key oracle and Sign oracle, which are the same as that in GameForgery. A_{II} outputs a certificate less signature on message m corresponding to public key pk_{ID} for an identity ID . A_{II} wins the game if Verify(params; ID ; pk_{ID} ; m ;) = 1 and the following conditions hold:

{ A_{II} has never been queried Private key or-acle on ID .

{ A_{II} has never been queried Replace Z key oracle on ID .

{ A_{II} has never been queried Signature oracle on (ID ; m).

The success probability of A_{II} is defined as the probability that it wins in Game II.

Accordingly, the security definitions of any certificate less digital signature scheme in the Random Oracle Model (ROM) can be given as follows.

Definition 1. A certificate less signature scheme is (t; q_H ;q_e; q_z; q_{sk}; q_{pk}; q_s;)-existentially unforgeable against Type I adversary under adaptively chosen mes-sage attacks if no t-time adversary A_I , making at most q_H to the random oracles, q_e partial private key queries, q_z to the Z key queries, q_{sk} private key queries, q_{pk} public key queries and q_s signature queries, have a success probability at least in Game I.

Definition 2. A certificate less signature scheme is (t; q_H ;q_z; q_{sk}; q_{pk}; q_s;)-existentially unforgeable against Type II adversary under adaptively chosen message at-tacks if no t-time adversary A_{II} , making at most q_H to the random oracles, q_z to the Z key queries, q_{sk}pri-vate key queries, q_{pk} public key queries and q_s signature queries, have a success probability at least in Game II.

Definition 3. A certificate less signature scheme is existentially unforgeable under adaptively chosen message attack (EUF-CMA), if the success probability of any polynomially bounded adversary in the above two games is negligible.

Theorem 1.Hassouna et al.'s [8] digital signature scheme is secure against existential forgery under adap-tively chosen message attacks in the random oracle model with the assumptions that CDHP(Computation Diffie-Hellman Problem) and BDHP (Bilinear Diffie-Hellman Problem) in G_1 are intractable.

The full proof of Theorem 1 in the random oracle model is stated in [9].The Proposed Hierarchal Certificate less Public Key Cryptography Scheme (HCL-PKC

Al-Ryami and Paterson introduced a Hierarchal Certificate less Encryption scheme (HCL-PKE) in their originalpaper [1]. Their HCL-PKE did not provide a trust Level 3 at the sense of Girualt'sDefinition[6]. Therefore, it was not acceptable as alternative to the traditional hierarchal PKI. In this section, we use Hassouna et al.'s [8] signature scheme as assistant technique to propose a new Hierarchal Certificate less Cryptography scheme (HCL-PKC) which is based on Hassouna et al.'s [7] CL-PKI model. The proposed HCL-PKC (See Figure 3) is straightforward and could provide a trust Level 3.

Root KGC Setup. The KGC chooses a secret parameter k to generate G_1; G_2; P; e, where G_1 (additive group) and G_2 (multiplicative group) are two groups of a large prime order q, P is a generator of G_1 and e : $G_1 G_1$! G_2 is a bilinear map. The KGC randomly generates the system's master keys x_0; x^0_0 2 Z_q and computes the system public key $X_0 = x^0_0 P$ and the private key term $Z_0 = x_0 P$. Then, the KGC chooses cryptographic hash functions H_1 and H_2, where H_1 : f0; 1g G_1 ! G_1 and H_2 : f0; 1g ! Z_q. Finally, the KGC publishes the system parameters params =< G_1; G_2; e; P; X_0; H_1; H_2; n >, while the secret master-keys are saved and secured by the KGC.

Set-Secret-Value. The user at level t with identiityID$_t$, where ID_0 is the identity of the root KGCdownloads the system parameters params, generates two random secret numbers x_t; x^0_t 2 G_2. As in the signature scheme, we enforce the user to choose astrong password pass, the system at the client side hashes the password to be z_m = H_2(pass), multiplies the base point P by the hashed password to be $z_m P$, uses the hashed value z_m as a key to encrypt the secret value x_m and generates the Password-based Encryption Code (PEC) as P EC_{zm} (x_m), sends copyof it to the KGC's public directory and stores copy of it along with the point $z_m P$ locally.

Set-Public-Key. The user at level t calculates its public key (X_t; Y_t) as $X_t = x^0_t P$ and $Y_t = x^0_t Q_t$ where $Q_t = H_1(ID_t; X_t)$. Then, the user sends X_t to the previous user in the hierarchy ID_{t1}.

Every user in the system has a unique record in the Public Directory (PD) which contains the information

fID$_t$; Q$_t$; X$_t$; Y$_t$; P EC$_{zt}$ (x$_t$); M P$_{1t}$; s$_t$g. We can think about the user's record as X.509 certificate. Hence, the

Interoperability between the traditional PKI system and this proposed HCL-PKC scheme will be easy because the two systems will be compatible.

Furthermore, the proposed HCL-PKC scheme provides a new mechanism to authenticate the user's public key and provides a trust Level 3 as same as the hierarchal PKI does. That means if the user's public key has been replaced, then no one accepts the user's intermediate KGC can do that. This because no one can replace the signature term by a valid one except the user's intermediate KGC. Therefore, the user can detect and determine the entity that has replaced his/her public key.

Even if the KGC or the intermediate KGC replaces (temporarily) the public key (as in the traditional PKI system) in order to compromise that user for decryption or signature forgery, this attack will fail because the user's private key is calculated from another different secret value. So, replacing the user's public key is not enough for compromising that user.

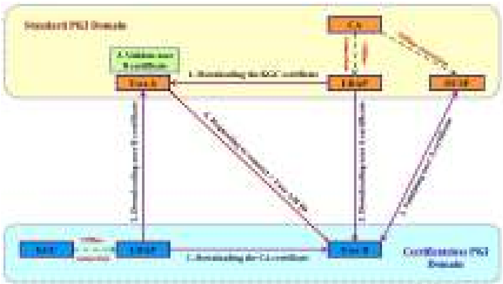


Figure 3: The proposed HCL-PKC model

Therefore, the separation of public/private key generation provides strong security feature.

Hybrid PKI/CL-PKI Scheme Suppose we have organization with two domains, the rst domain utilizes the traditional PKI with one CA and one LDAP server for trust distribution. The other domain has the Hassouna et al.'s [7] CL-PKI which has the same structure as the traditional PKI, i.e it uses X.509 certificate format to load the certificate less user's information with the signature as Hassouna et al.'s [8] one stored encrypted value x_t, and after that uses the extracted value x_t to calculate the full private key by $(x_t+z_t)D_t$ and the term $Z_t = x_tP$. In case of a mismatch, the system aborts the process. match, then the password is correct and the user is authenticated. The user then uses (z_t) as a key to decrypt the

Then, the two domains can operate smoothly as follows:

Bridge Model: Bridge trust model can be used between the CA of the PKI and the KGC of the CL-PKI. Then, the CA generates and signs the X.509 certificate (using a standard PKI and ECC-based signature scheme like ECDSA) to the KGC that includes the KGC's public parameters. Also, the KGC generates and signs the X.509 certificate (using the Hassouna et al.'s signature scheme) to the CA that includes the CA's public key. The CA stores the KGC's certificate into its local LDAP server and also the KGC stores the CA's certificate into its local LDAP server. Since the recent versions of the PKI-enabled protocols like TLS v1.2 protocol [4] have be-come supportive to the Elliptic Curve Cryptosystems like ECDSA signature scheme and ECDH key exchange protocol as Hassouna et al.'s CL-PKI-enabled protocols did, then it is possible to agree on using the ECDH for key exchange protocol to generate the symmetric key. The other parameters can be agreed on at the handshake phase of the transaction. Note that the users at the PKI domain needs to equipped with the pairing algorithm in order to do the signature generation/verification.

Extract-Partial-Private-Key. The user at level t 1 accepts the request of the users at level t (the request contains the terms Q_t and X_t) and calculates their partial private key D_t as $D_t = x_{t1}Q_t$. Furthermore, the user at level t 1 signs the public term X_t of the user at level t using the proposed CL-SS scheme with the terms Z_{t1} and the per-signature random number a_{t1} and creates the signature as (X$_t$; M P$_{1t}$; s$_t$) and puts this signature along with the rest of user's public terms into the public directory fID$_t$; Q$_t$; X$_t$; Y$_t$; M P$_{1t}$; s$_t$g.

Set-Private-Key. Every time the user at level t needs to calculate and use his/her full private key, he/she enters his/her password, the system hashes it as z_m^0, calculates z_t^0P andpointz$_m$P . If the comparison result in a

PKI Domain's User: User A in the PKI domain when encrypting/signing a message to user B in the CL-PKI domain, he/she needs to do as follows:UserArst request B's certificate either directly from user B or from the CL-PKI's LDAP server. After the user A gets user B's certificate, down-loads the KGC's certificate from his/her local LDAP server. Then, he/she uses CA's public key to validate the KGC's certificate. If it is not valid, then user A rejects and aborts the transaction. If the KGC's certificate is valid, then user A extracts KGC's public key and uses it to verify B's certificate by verifying the signature on the user B's certificate using the Hassouna et al. signature scheme. User A also can verify the expiry/revocation of the user B's certificate using either the CRL mechanism or the OCSP protocol. After user A authenticates user B, then users A and B can start the handshake protocol to agree on the key size, generate per-session symmetric encryption key using ECDH protocol, agree on the encryption algorithm, hash function and the signature algorithm (ECDSA for PKI users and the Hassouna et al.'s one for CL-PKI users). CL-PKI Domain's User: User B in the CL-PKI domain when encrypting/signing a message to user A in the PKI domain, he/she does the following: User B requests A's certificate either directly from user A or from the PKI's LDAP server. After user B gets user A's certificate, downloads the CA's certificate from his/her local LDAP server, then he/she uses KGC's certificate to authenticate the CA's certificate (using Hassouna et al.'s signature scheme). If it is not valid, then user B rejects and aborts the transaction. If the CA's certificate is valid, then user B extracts CA's public key and uses it to verify B's certificate (prefer to use ECDSA algorithm).User B also can verify the expiry/revocation of the user A's certificate using either the CRL mechanism or the OCSP protocol rule based as in the traditional PKI system. After user B authenticates user A, then users A and B can start the handshake protocol to agree on the encryption key size, generate per-session symmetric encryption key using ECDH protocol, agree on the encryption algorithm, hash function and the signature scheme (ECDSA for PKI users and Hassouna et al.'s one for CL-PKI users).

IV. Conclusions and Remarks

This paper used the Hassouna et al[8] signature scheme and proposed a trust Level 3 hierarchal certificate less public key cryptography scheme. The proposed hierarchal scheme is based on Hassouna et al.'s [7] CL-PKI model. Therefore, it enjoys the same security features that CL-PKI has, along with the interesting trust Level 3 satisfaction property. The paper also proposed a new Hybrid PKI/CL-PKI scheme that provides interoperability model between traditional PKI and CL-PKI systems in one organization under the X.509 certificate format.

References

[1] Heinzelman W, Chandrakasan A, Balakrishnan H, “Energy-Efficient communication protocol for wireless microsensor networks”, Proc. of the 33rd Annual Hawaii Int l Conf. on System Sciences. Maui: IEEE Computer Society, 2000, pp. 3005-3014.

[2] Heinzelman W, Chandrakasan A, Balakrisan H. “An application specific protocol architecture for wireless microsensor networks”, IEEE Transaction on Wireless Networking, 2002, vol. 1, no. 4, pp. 660-670.

[3] S. Lindsey and C. S. Raghavendra, “PEGASIS: Power-efficient gathering in sensor information systems,” Proc. of the IEEE Aerospace Conf. Montana: IEEE Aerospace and Electronic Systems Society, 2002, pp. 1125-1130.

[4] S. SoroandW. Heinzelman, “Prolonging the lifetime of wireless sensor networks via unequal clustering”, Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium, 2005.

[5] Li C F, Ye M, Chen G H and Wu J, “An energy-efficient unequal clustering mechanism for wireless sensor networks”, Proc. of the 2nd IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), Washington, DC, 2005.

[6] Krishnamachari B, Estrin D, and Wicker S. “Modeling data-centric routing in wireless sensor networks”. IEEE Infocom Proceedings. New York: IEEE Computer Society, 2002, pp. 2-14.

[7] HuseyinOzgur Tan, Ibrahim Korpeoglu. “Power efficient data gathering and aggregation in wireless sensor networks”. SIGMOD Record, 2003, vol. 32, no. 4, pp. 66-71.

[8] M. Ye, C. F. Li, G. H. Chen, and J. Wu, “EECS: an energy efficient clustering scheme in wireless sensor networks”, Proc. of IEEE Int’l Performance Computing and Communications Conference, 2005, pp. 535-540.

[9] Ye M, Li C F, Chen G H and Wu J. “An energy efficient clustering scheme in wireless sensor networks,” International Journal of Ad Hoc & Sensor Wireless Networks, 2007, vol. 3, no. 2, pp. 99-119.

[10] Lindsey S, Raghavendra CS. “PEGASIS: power-efficient gathering in sensor information systems,” Proc. of the IEEE Aerospace Conf. Montana: IEEE Aerospace and Electronic Systems Society, 2002, pp. 1125-1130.

[11] LIU Ming, CAO Jian-Nong, CHEN Gui-Hai, CHEN Li-Jun, WANG Xiao-Min, GONG Hai-Gang. “EADEEG: an energy-aware data gathering protocol for wireless sensor networks,” Journal of Software, 2007, vol. 18, no. 5, pp. 1092ĭ1109.

[12] Tao Yang, ZhengYaling. “The combination of the optimal number of cluster-heads and energy adaptive cluster-head selection algorithm in wireless sensor networks,” Wireless Communications, Networking and MobileComputing,2006.WiCOM 2006.International Conference onSept2006,pp 1-4.