Secure and Efficient Threshold Key Issuing Protocol for ID-based Cryptosystems

K. Phani Kumar, G. Shailaja, Ashutosh Saxena

Secure Technology Lab.,
Institute for Development and Research in Banking Technology
Castle Hills, Masab Tank, Hyderabad 500057, INDIA.
kpkumar@mtech.idrbt.ac.in, gshailaja@mtech.idrbt.ac.in, asaxena@idrbt.ac.in

Abstract

Key issuing protocols deal with overcoming the two inherent problems: key escrow and secure channel requirement of the identity based cryptosystems. An efficient key issuing protocol enables the identity based cryptosystems to be more acceptable and applicable in the real world. We present a secure and efficient threshold key issuing protocol. In our protocol, neither KGC nor KPA can impersonate the users to obtain the private keys and thus it achieves the trust level III [16]. The protocol is secure against replay, man-in-the-middle and insider attacks.

1 Introduction

Traditional public key cryptography (PKC) uses certificates, issued by a certification authority (CA), to bind the users with their public keys. Although certificates are the best alternative for key distribution, it has the following drawbacks: (i) users should send certificates along with the message and signature (ii) receiver should verify the certificate validity before verifying the actual signature (iii) managing the certificates and maintaining the certificate revocation lists (CRL) is cumbersome. Moreover if there is a chain of CAs involved in the process of issuing a certificate, verifier should verify all the certificates in the chain.

Identity based (ID-based) cryptography introduced by Shamir [12] in 1984, overcomes these problems by avoiding the use of certificates. Identity based cryptography uses the user's identity such as social security number (SSN), passport number as his public key. The private keys of the users are issued by a private key generator (PKG) through a secure channel, after verifying the user's credentials. Thus, the trust over PKG removes the need of certificates in ID-based cryptography. Any identity based cryptosystem includes two phases namely `Setup` and `Key extraction/generation and issuing` that are carried out by PKG.

Even though ID-based cryptography overcomes the problems in the traditional PKC, it suffers from two inherent problems: key escrow and secure channel requirement. The PKG has the

knowledge of the users' private keys and therefore can decrypt any cipher text or forge signature on any message which is known as key escrow problem. Moreover key issuing requires secure channel to avoid eavesdropping. Due to these reasons identity based cryptosystems are confined to closed groups [12].
Trust between the users and the trusted third party (TTP) plays an important role in public key cryptosystems. In 1998, Girault introduced three trust levels [16], based on the following trust assumptions

- level I: The TTP knows (or can easily compute) the users' private keys and therefore can impersonate any user at any time without being detected, also known as **key escrow** problem.
- level II: The TTP does not know (or cannot easily compute) the users' private keys. But, the TTP can still impersonate a user by generating a false public key without being detected.
- level III: The TTP does not know (or cannot easily compute) the users' private keys. Moreover, it can be proved that the TTP generated false public keys of users if it does so.

Key issuing protocol for identity based cryptosystems deals with the `Key generation and issuing` phase to avoid the key escrow problem and need for secure channel. An efficient key issuing protocol enables the identity based cryptosystems to be more applicable in the real world. The first key issuing protocol was presented by Boneh and Franklin [4] in 2001. Later on, Lee et al. [9] and Gangishetti et al. [5] have proposed key issuing protocols which use one key generation center (KGC which is nothing but PKG) and multiple key privacy authorities (KPAs) for issuing the private keys to the users. In their approach the key escrow problem can be avoided if atleast one of the KPAs is honest. However, private keys of all the users have to be reconstructed if the private key of even one of the KPAs is compromised. A Threshold key issuing protocol allows one to distribute the power of computing the private key of the users among several authorities in such a way that (1) no group of corrupt authorities (equal to or smaller than a given threshold) can figure out what the user secret is, even if they cooperate; known as *unforgeability* (2) a large enough number of authorities (a number larger than the threshold) are required for generating the user secret.

In this paper, we propose a secure and efficient (t, n) threshold key issuing protocol which involves one KGC and n KPAs. Our protocol does not require secure channel for key issuing and eliminates the key escrow problem completely. We show that the protocol achieves the trust level III, thus overcoming the problem of KGC impersonation existing in several schemes [13] [9] [5]. We also show that replay, man-in-the-middle and insider attacks are not possible on the proposed protocol.

The rest of the paper is organized as follows: In section 2, we review the various existing key issuing protocols. We give the mathematical concepts and hard problems in section 3, followed by the model for the proposed protocol in section 4. In section 5, we explain the threshold key issuing protocol. Section 6 gives the salient features and security analysis of the proposed protocol followed by the conclusions.

2 Related Work

The problem of secure key issuing can be addressed in two different ways.

1. Multiple key issuing authorities [4] [1] [10] [8] [9] [5] [13].
2. Embedding user chosen secret information in the private key issued by the key issuing authorities [7] [3].

We give a brief review of these schemes in this section.

In 2001, Boneh & Franklin [4] addressed the problem of key escrow in identity based cryptosystems using distributed PKGs i.e. instead of one PKG issuing the user secret they used n PKGs. User obtains partial private keys from each PKG and combines them to get the private key. Thus, the key escrow problem can be avoided if atleast one-out-of-n PKGs is honest. They also suggested that their approach can be extended to threshold key issuing using Shamir secret sharing [11]. In their approach, all the PKGs are at the same level. Therefore, a user has to be registered at each PKG, which is practically difficult to perform. Moreover, the protocol requires secure channel to issue partial private keys.

In 2002, Chen et al. [1] and Paterson [10] have given solutions which are similar to that of Boneh et al. In these schemes, each trusted party has to check and authenticate user identity independently which is not practically feasible.

In 2003, Hess [8] proposed a protocol using the concept of multiple trust authorities to avoid the key escrow problem. Gentry [7] proposed a certificate based encryption scheme that provides secure key issuing by embedding user chosen secret information in the private key. Later, Al-Riyami and Paterson [3] proposed certificateless public key cryptography. They also used the user chosen information for eliminating the key escrow problem. Though the schemes [7] [3] are successful in removing the key escrow problem, they loose the advantages of ID based crypto systems.

In 2004, Sui et al. [13] proposed a separable and anonymous key issuing protocol without secure channel. However, Kim et al. [14] have shown that their protocol suffers from impersonation attack by KGC. Thus the scheme obtains only trust level I and the problem of key escrow still remains. In the same year, Lee et al. [9] proposed a key issuing protocol, addressing the key escrow problem and secure channel requirement. In this protocol, a users private key is issued by a key generation center, and its privacy is protected by multiple key privacy authorities (KPAs). These authorities work in a sequential mode. Only one authority (the KGC) has to authenticate the user and thus it greatly reduces the cost of user authentication. The scheme also makes use of user-chosen secret information for constructing a secure channel for a user to retrieve his partial private key securely. However, the scheme suffers from the following attacks as pointed out by Gangishetti et al. [5]: (i) impersonation attack (can be done by any user) (ii) insider attack (can be done by any of the KPAs) (iii) Incompetency of KPAs. Moreover, Chunxiang et al. [2] have shown that a malicious KGC can successfully attack the Lee et al.'s protocol to obtain users private keys. Thus, this scheme attains trust level I.

In 2005, Gangishetti et al. [5] proposed a new key issuing protocol, which involves one KGC and n KPAs. According to the protocol, KGC gives a registration identity, r_{ID} to the user during the registration. User uses this r_{ID} as blinding factor while collecting the partial private keys

from KPAs. We observe that a malicious KGC can also get the private key of any user, since it knows the blinding factor r_{ID}. Thus, the problem of key escrow still remains in the protocol and it reaches trust level I.

3 Mathematical Background

In this section, we briefly review the mathematical concepts that we require throughout this paper. We use elliptical curve groups and bilinear maps to describe the protocol, as these are the most commonly used primitives in ID-based cryptography [6]. Nonetheless, the protocol can also be used by the systems built on ordinary groups like ${Z_q}^*$.

3.1 Bilinear pairings

Let G_1 be an additive cyclic group of large prime order q, G_2 be a multiplicative cyclic group of the same order and P be a generator of G_1. A cryptographic bilinear map e is defined as $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

Bilinear: $e(aR, bS) = e(R, S)^{ab}$ $\forall R, S \in G_1$ and $a, b \in Z_q^*$.
Non-degeneracy: For each $R \neq O \in G_1$, there exists $S \in G_1$ such that $e(R, S) \neq 1$, where O is the identity element in G_1 and 1 is the identity element in G_2.
Computable: There exists an efficient algorithm to compute $e(R, S)$ $\forall R, S \in G_1$.

In general implementation, G_1 is the group of points on an elliptic curve and G_2 denotes a multiplicative subgroup of a finite field. Typically, the mapping e is derived from either the Weil or the Tate pairing on an elliptic curve over a finite field. We refer to [4] for more comprehensive description on how these groups, pairing and other parameters are defined.

3.2 Computational Problems

Here, we present some computational hard problems, which will form the basis of security of our threshold key issuing protocol.

Discrete Logarithm Problem (DLP): Given two elements $R, S \in G_1$, find an integer $a \in Z_q^*$, such that $S = aR$ whenever such an integer exists.

Computational Diffie-Hellman Problem (CDHP): For any a, $b \in Z_q^*$, given $< P, aP, bP >$, compute abP.

Decisional Diffie-Hellman Problem(DDHP): For any $a, b, c \in Z_q^*$, given $< P, aP, bP, cP >$, decide whether $c \equiv ab \bmod q$.

Bilinear Diffie-Hellman Problem (BDHP): For any $a, b, c \in Z_q^*$, given $< P, aP, bP, cP >$, compute $e(P, P)^{abc}$.

Gap Diffie-Hellman Problem(GDHP): A class of problems, where DDHP can be solved in polynomial time but no probabilistic polynomial time algorithm exists which can solve CDHP.

4 Model for the proposed protocol

We present the entities involved and the model for the proposed protocol in this section.

4.1 Entities involved

There are three entities involved namely KGC, KPA and user in our protocol.
KGC: Key Generation Center is the central entity meant for user registration. It checks the user identity and issues a proof that the user has been registered. The registration process is done off-line. It also maintains a database of the registered users. This database is publicly available, but can be modified by the KGC only. It also issues partial private keys to the registered users.
KPAs: n Key Privacy Authorities are used to provide the key privacy service. On receiving a blinded request from a user, these KPAs verify whether the user has been registered or not, using the database. If the user is registered, KPAs compute the partial private key and send it to the user. Each KPA maintains a database of received requests, which need not be kept secret to avoid KGC impersonation attack.
User: User is initially registered at KGC. It selects any $t+1$ out of n KPAs and gets the partial private keys from the selected KPAs and the KGC. User combines these partial private keys to get it's private key. A user with identity ID is denoted by U_{ID} and Q_{ID}, D_{ID} are it's public and private keys respectively.

4.2 Phases of the proposed protocol

The proposed threshold key issuing protocol broadly consists of *two* phases (i) `Setup` (ii) `Key Generation and Issuing`. The first phase is carried out only once, securely by the KGC and KPAs. The second phase is carried out jointly by all the entities whenever a new user joins the system. These phases can be described as follows:

- `Setup:` This consists of two sub-phases namely (i) System Setup and (ii) System Key Generation and Distribution
  - `System Setup:` The KGC selects its private key and specifies the system parameters `params`.
  - `System Key Generation and Distribution:` In this phase KPAs interactively compute the system key and distribute it using the distributed key generation protocol of Gennaro et al. [15]. Each KPA computes and sends its public parameters to the KGC.
- `Key Generation and Issuing:` This phase describes how a new user joins the system and constructs the private key securely from the KGC and KPAs.
  - `Registration:` User supplies his credentials and some parameters to KGC for registration. KGC maintains a database of the registered users that is publicly available and issues a proof of registration to the registered users.

- `KGC Request:` User sends request to the KGC to obtain the partial private key.
- `KGC response:` On receiving the user request, KGC checks whether user has been registered or not and issues the blinded partial private key.
- `Blind KPA Request:` User selects any $t+1$ out of n KPAs and requests them in parallel to provide key privacy service by sending a blind request.
- `KPA Response:` Each KPA authenticates the user and issues a blinded partial private key to the authenticated user.
- `Key Retrieval:` On receiving all the blinded partial private keys, user combines them and then unblinds the resulting value to produce its private key.

5 Proposed Threshold Key Issuing Protocol

This section presents the proposed key issuing protocol for the identity based crptosystems. In our protocol, the user has to registered at the KGC to get the partial private keys from KPAs. The user is given the flexibility to choose any $t+1$ out of n KPAs, where $t \leq n \leq (2t+1)$. The following two phases describe the protocol in detail:

- **Setup:** This phase can be further divided into two subphases namely (i) *System Setup* and (ii) *System Key Generation and Distribution.*

  1. `System Setup:` KGC chooses two groups G_1, G_2 of same prime order q and a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$, where G_1 is an elliptical curve group and G_2 is subgroup of a finite field. Let P be the generator of the group G_1. $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \rightarrow {Z_q}^*$ are the two cryptographic hash functions chosen by the KGC. KGC chooses a random number $s_0 \in {Z_q}^*$, which is it's secret and computes its public key as $Pub_{KGC} = s_0 P$.
  2. `System Key Generation and Distribution:` In this phase, n KPAs collaboratively run the Gennaro's key generation and distribution [15] protocol and share the secret s such that any of the $t+1$ KPAs can construct it with their own secret s_i $(i = 1, 2..... n)$. Finally, each KPA_i $(i = 1, 2..... n)$ computes and sends its public key $Pub_{KPA_i} = s_i P$ to the KGC. Now any $t+1$ out of the n KPAs compute the secret key s as follows:

     $$s = \sum_{i \in \Delta} \lambda_i(0) s_i$$

     where λ_i is the lagrange coefficient and Δ consists of the indexes of the participating KPAs.

  KGC verifies the public keys of all KPAs, computes the system public key

  $$Pub_{sys} = s_0 \sum_{i \in \Delta} \lambda_i(0) Pub_{KPA_i}$$

  and publishes the public parameters $\langle G_1, G_2, P, e, Pub_{sys}, Pub_{KGC}, Pub_{KPA_1}, ..., Pub_{KPA_n} \rangle$.

- **Key Issuing:** This phase consists of six sub phases and is meant for user registration and issuing the private key securely to the user. For a user with identity ID, denoted by U_{ID} the public key is $Q_{ID} = H_1(ID)$ and the corresponding private key D_{ID} is obtained as follows:

1. `Registration:` A user U_{ID} chooses $r_{ID} \in {Z_q}^*$ at random and submits $\langle ID,\ {r_{ID}}^{-1}P \rangle$ to KGC to get registered. KGC verifies the user credentials, registers the user and issues a proof of registration $prf_{ID} = s_0 H_1(ID||{r_{ID}}^{-1}P)$ to the user. The tuple $\langle ID,\ {r_{ID}}^{-1}P \rangle$ is stored in the publicly available database maintained by the KGC.
2. `KGC Request:` User U_{ID} computes $r_{ID}P$ and sends the tuple $\langle ID,\ r_{ID}P \rangle$ to the KGC.
3. `KGC Response:` On receiving the tuple, KGC fetches ${r_{ID}}^{-1}P$ corresponding to ID from the database and checks the validity of the received tuple using

   $$e(r_{ID}P,\ {r_{ID}}^{-1}P) \stackrel{?}{=} e(P,\ P).$$

   If the tuple is valid, KGC chooses $x \in {Z_q}^*$, computes the blinded partial private key $W = s_0 Q_{ID} + xP$, $V = x{r_{ID}}^{-1}P$ and sends them to the user.
4. `Blind KPA Request:` On receiving W,V the user U_{ID} un-blinds and obtain the partial private key ${D^0}_{ID} = s_0 Q_{ID} = W - r_{ID}V$. The user can check the correctness of the partial private key as follows:

   $$e({D^0}_{ID}, P) \stackrel{?}{=} e(Q_{ID}, Pub_{KGC})$$

   User selects any $t+1$ out of n KPAs and sends $\langle ID,\ r_{ID}s_0Q_{ID} \rangle$ to the selected KPAs.
5. `KPA Response:` On receiving the tuple $\langle ID,\ r_{ID}s_0Q_{ID} \rangle$, each KPA_i verifies whether partial private key has already been issued for this ID using it's database. If so, KPA_i informs the same to the sender and terminates the operation. If not, KPA_i fetches ${r_{ID}}^{-1}P$ corresponding to ID and checks the validity of the received tuple using

   $$e(r_{ID}s_0Q_{ID},\ r_{ID}^{-1}P) \stackrel{?}{=} e(Q_{ID},\ Pub_{KGC})$$

   and authenticates the user. If so, KPA_i computes blinded partial private key $W_i = s_i r_{ID} s_0 Q_{ID}$ and sends it to the corresponding user. KPA adds the tuple $\langle ID,\ r_{ID}s_0Q_{ID} \rangle$ to it's database.
6. `Key Retrieval:` On receiving all the blinded partial private keys W_i from KPAs, user does the following to obtain its private key:
   (a) Combines all the blinded partial private keys to obtain blinded private key i.e

   $$D'_{ID} = \sum_{i \in \triangle} \lambda_i(0) W_i = r_{ID} s_0 s Q_{ID}$$

   where $\triangle$ contains the indices of the KPAs selected by the user and $\lambda_i(0)$ is the Lagrange coefficient.
   (b) Unblinds private key using r_{ID} i.e.

   $$D_{ID} = {r_{ID}}^{-1} D'_{ID} = s_0 s Q_{ID}$$

   (c) Checks the correctness of the private key by the equality:

   $$e(D_{ID}, P) \stackrel{?}{=} e(Q_{ID}, Pub_{sys})$$

This completes the description of the proposed protocol.

6 Analysis

We analyze the salient features and security of the proposed protocol in this section.

6.1 Salient features

Here we give the features that our protocol enjoys:

1. *Achieves trust level III*: It is clear that KGC issues a part of the private key and does not know the complete private key of the user. However, KGC may try to impersonate a user and obtain partial private keys to construct the private key. We have designed our protocol such that a malicious KGC can be identified if it tries to impersonate a user as explained further. Let us suppose that KGC tries to impersonate a user U_{ID}. For this, KGC modifies the corresponding entry $r_{ID}^{-1}P$ in the database to $r'^{-1}P$ for some random $r' \in {Z_q}^*$, selects any $t+1$ KPAs and approaches them with the tuple $ID, r's_0Q_{ID}$ for the partial private keys. Since, $t \leq n \leq (2t+1)$, atleast one KPA will receive partial private key request from both the user and malicious KGC and it identifies that there is some discrepancy. Using the proof of registration $prf_{ID} = s_0H(ID||r_{ID}^{-1}P)$, the user can now prove that KGC has modified the database entry and impersonated it. Thus, our scheme achieves trust level III.

2. *Fault tolerance*: t or less than t KPAs will not be able to generate the user private key in the protocol. Moreover, the protocol is fault tolerant if $n = 2t+1$ i.e. key issuing is possible even in the presence of t malicious KPAs.

3. *Avoids secure channels*: In general, a secure channel is required to transmit the partial private keys to avoid eavesdropping. We overcome the need for secure channel using the blinding factor r_{ID}.

4. *Robust authentication*: The private key of each user is issued after the following two authentications. (i) User first authenticates with the KGC in off-line mode (ii) User uses the r_{ID} in `KGC Request` and `blind KPA request` to authenticate itself which is online.

5. *Open database*: The databases maintained by the KGC and KPAs need not be kept secret, but their integrity must be guaranteed.

6. *Key revocation*: Key revocation is possible in our protocol if we include the private key expiry time in public key.

6.2 Security Analysis

The security of the proposed threshold key issuing protocol relies on the hardness of solving DLP in elliptic curve groups and is secure against the following attacks.

1. **Unforgeability:** It is not possible to forge the `KGC request` and `Blind KPA request` tuples, since r_{ID} is required to compute these tuples which is known only to the user U_{ID}. The security of proposed protocol relies on the hardness of solving DLP. For instance, finding r_{ID} from the `KGC request` is equivalent to solving DLP. Finding the blinding factor x in the `KGC response`, given $xr_{ID}^{-1}P$ and $r_{ID}^{-1}P$ is equivalent to solving DLP.

2. **Replay attacks:** Since r_{ID} is required to unblind the partial private keys, an adversary cannot obtain private key of the user even if he replays the request tuples.

3. **Man-in-the-middle attacks:** The distributed key generation protocol [15] used in the `Setup` phase is secure against man-in-the-middle attacks. Further in `Key Issuing` phase, if an adversary alters the KGC or KPA response tuples i.e. the partial private keys, then it can be detected in the subsequent phases as the user checks the correctness of the received terms.

4. **Insider attacks:** Dissimilar to [9], in the proposed protocol a KPA cannot cheat the other KPAs since they work in parallel and it does not know the other t KPAs that the user has selected. Moreover, a malicious KGC will be detected if it tries to impersonate a user to obtain partial private keys from the KPAs.

7 Conclusion

We have proposed a secure and efficient threshold key issuing protocol. The protocol does not require any secure channel to issue the private key and is secure until the threshold number of KPAs are compromised. A malicious KGC will be identified if it tries to impersonate the users. We have shown that the scheme achieves the trust level III and is resilient to the replay, man-in-the-middle and insider attacks.

References

[1] L. Chen, K. Harrison, N. P. Smart and D. Soldera, Applications of Multiple Trust Authorities in pairing based Cryptosystems,InfraSec 2002, LNCS 2437, Springer-Verlag, pp. 260-275, 2002.

[2] X. Chunxiang, Z. Junhui and Q. Zhiguang, A Note on Secure Key Issuing in ID-based Cryptography, Cryptology ePrint Archive, Report 2005/180, 2005.

[3] S. Al-Riyami, K. Paterson, Certificateless Public Key Cryptography, Advances in Cryptology - Asiacrypt'03, Springer-Verlag, LNCS 2894, pp. 452-473, 2003.

[4] D. Boneh, M. Franklin, Identity-based Encryption from the Weil pairing, In J. Kilian, editor, Advances in Cryptology-Crypto'01, Springer-Verlag, LNCS 2139, pp. 213-229, 2001.

[5] R. Gangishetti, M. C. Gorantla, M. L. Das, A. Saxena and V. P. Gulati, An Efficient Secure Key Issuing Protocol in ID-Based Cryptosystems, In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), Volume-1, IEEE Computer Society, pp. 674-678, 2005.

[6] M. Choudary Gorantla, G. Raju and Ashutosh Saxena. A Survey on ID-based Cryptographic Primitives, Cryptology ePrint Archive: Report 2005/094. http://eprint.iacr.org/2005/094.

[7] C. Gentry, Certificate-Based Encryption and the Certificate Revocation Problem, Advances in Cryptology- Eurocrypt'03, Springer- Verlag, LNCS 547, pp.272-293, 2003.

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

[9] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, S. Yoo, Secure Key Issuing in ID-based Cryptography", In proceedings of the Second Australian Information Security Workshop-AISW 2004, ACS Conferences in Research and Practice in Information Technology vol.32, pp.69-74, 2004.

[10] K. Paterson, Cryptography from Pairings: a snap shot of current research, Information Security Technical Report, Vol.7(3), pp.41-54, 2002.

[11] A. Shamir, How to Share a Secret, Comm. ACM, vol.22(11), pp.612-613, 1979.

[12] A. Shamir, Identity-based Cryptosystems and Signature Schemes, In Advances in Cryptology - Crypto'84, Springer-Verlag LNCS 196, pp.47-53, 1985.

[13] A. Sui, S. S. M. Chow, L. C. K. Hui, S. M. Yiu, K. P. Chow, W. W. Tsang, C. F. Chong, K. H. Pun and H. W. Chan, "Seperable and Anonymous Identity-Based Key Issuing without Secure Channel" in Proc. of the 11th International Comference on Parallel and Distributed Systems (ICPADS 2005), Vol. 2, pp. 275-279, 2005.

[14] H. Kim, S. Kim, D. Won, Cryptanalysis of Sui et al.'s Second ID-based Key Issuing Protocol without Key Escrow, IJCSNS International Journal of Computer Science and Network Security, Vol.6 No.1B, pp. 247-250, 2006.

[15] R. Gennaro, A. Jareki, H. Krawczyk, T. Rabin, Secure Distributed Key Generation for Discrete-Log Based Cryptosystems, Advances in Cryptology - Eurocrypt'99, Springer-Verlag, LNCS 1592, pp.295-310, 1999.

[16] M. Girault, Self-certified public keys, In EUROCRYPT 1991, , LNCS 547, pp. 490-497, Springer-Verlag, 1991.