Distributed Private Key Generation for Identity Based Cryptosystems in Ad Hoc Networks

Aldar C-F. Chan

Abstract—Identity Based Cryptography (IBC) has the advantage that no public key certification is needed when used in a mobile ad hoc network (MANET). This is especially useful when bi-directional channels do not exist in a MANET. However, IBC normally needs a centralized server for issuing private keys for different identities. We give a protocol distributing this task among all users, thus eliminating the need of a centralized server in IBC for use in MANETs.

I. INTRODUCTION

In Identity-Based Cryptography (IBC), the public key of each user is derived from his identity which could be an arbitrary string. Each user needs to obtain from a server called private key generator (PKG) his private key for his identity. To encrypt a message for a user, only his identity and the PKG’s public key are needed and no public key certification is needed. The PKG can be seen as an admission agent for an ad hoc group.

When used in mobile ad hoc networks (MANETs), IBC has clear advantage over standard public key schemes in that others can send a message to an authenticated user without interaction or pre-arrangement (assuming the PKG’s public key is universally known) — that is, non-interactive session key setup — which is a desired property when only a unidirectional channel exists between two nodes or accessing the certification authority (CA) is impossible; in standard public key schemes, users need to obtain the public key certificate from the recipient or the server.

However, the main problem of adopting IBC in MANETs is that a centralized server is needed as the PKG, which violates the self-organization nature of a MANET. Nodes in a MANET usually belong to different users, implying difficulty in finding a trusted server to issue user private keys. The PKG task must be distributed among all users. We give a protocol for this, thus increasing the usability of IBC for MANETs.

When users have no prior trust established, it would be tempting to use group key agreement to obtain a group key as the initial trust. However, the key agreement protocol needs to be conducted again whenever members join or leave the group and this rekeying process is not efficient in MANET and in most cases cannot survive or tolerate its changing topology. On the other hand, pre-computing all the group keys using key agreement is not feasible due to the huge key storage requirement. Distributing the public key certification task among users has been considered in [4]. Through the application of Feldman’s verifiable secret sharing scheme [7], a construction for sharing the task of the IBC-PKG among all users is given. More specifically, the main contribution of this article is that a distributed PKG implementation for Boneh-Franklin’s IBE [3] is presented, which allows the function of a trusted private key generator (needed for IBC) to be securely distributed among all the participating nodes in a MANET.

II. RELATED WORK

To avoid a single point of vulnerability and increase service availability, decentralized trust management is a widely recognized goal for entity authentication in various networks [1], [10], [5] including MANETs [14], [13]. However, as mentioned in [11], [4], the reliance on a trusted authority in the initialization phase in these schemes limits the flexibility of applying them to some MANET cases which do not require very strong entity authentication. The notions of distributed trust and self-organized entity authentication are thus introduced in [11], [4]. This paper adopts the same notion of distributed trust but uses an operation model more similar to that in [14], [13] which distributes the certificate issuing task among a group of users. But unlike [14] and [13] which requires a trusted authority to bootstrap the initial trust, no trusted authority in needed in this paper. Boneh and Franklin [2] gives a distributed RSA key generation protocol so that at the end a public key which is the product of two large primes is output and each user has a share of the RSA private key.

Gangishetti et. al [8] proposes a threshold key issuing scheme for Boneh-Franklin type of identity based cryptosystems to eliminate the key escrow property (inherent in identity based cryptosystems) through the involvement of users, along with the PKG, in issuing new user private keys. This scheme still needs a trusted third party to act as the PKG, and is thus not suitable for MANETs adopting the distributed trust notion. There are various proposals, such as [8], [9], to eliminate the undesirable property of ”key escrow”. But a trusted offline server is still needed.

Kate et. al [12] also proposed distributed private key generators for identity based cryptography, but this paper gives a more generalized construction, which actually predates [12] and first appeared as part of a doctoral thesis (pages 85-88 of [6]).

III. PRELIMINARIES

A. Computationally Hard Problems

Let $G_1$ be a cyclic additive group generated by a generator $G$, whose order is a prime $q$, and $G_2$ be a cyclic multiplicative group with the same order $q$. A bilinear pairing is a map $e: G_1 \times G_1 \rightarrow G_2$ with the following properties:
1) **Bilinearity:** \( \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} \) where \( P, Q \in G_1 \), \( a, b \in \mathbb{Z}_q^* \).

2) **Non-degeneracy:** \( \hat{e}(G, G) \neq 1 \). Therefore, it is a generator of \( G_2 \).

3) **Computability:** There is an efficient algorithm to compute \( \hat{e}(P, Q) \) for all \( P, Q \in G_1 \).

The security discussion in this paper is based on groups in which the following computational problem is assumed to be hard or any probabilistic poly-time (PPT) solution to the decisional problem is negligibly better than a wild guess.

**Definition 1:** **Computational Bilinear Diffie-Hellman (CBDH) Problem**

Given \( P \in G_1 \), \( aP, bP \) and \( cP \) for some unknowns \( a, b, c \in \mathbb{Z}_q^* \), find \( \hat{e}(P, P)^{abc} \).

**Definition 2:** **Decisional Bilinear Diffie-Hellman (DBDH) Problem**

Given \( P \in G_1 \), \( aP, bP \) and \( cP \) for some unknowns \( a, b, c \in \mathbb{Z}_q^* \), decide whether a given \( y \in G_2 \) satisfies that \( y \equiv \hat{e}(P, P)^{abc} \).

### B. Boneh-Franklin’s PKG

Let \( G_1 \) and \( G_2 \) denote multiplicative groups of order \( q \), and the pairing \( \hat{e} : G_1 \times G_1 \to G_2 \) exists. Assume the CBDH and DBDH assumptions hold for \( G_1 \) and \( G_2 \). A hash function \( H \) is used to map an identity \( ID \in \{0, 1\}^* \) to \( G_1 \).

In the Boneh-Franklin’s scheme, the PKG randomly picks a generator \( G \in G_1 \) and a private key \( x \in \mathbb{Z}_q^* \). The PKG’s public key is then \( (G, xG) \). This is called the setup phase. In the private key extraction phase, when a user with identity \( ID \) requests his private key, the PKG, using its private key \( x \), generates the user’s private key \( xH(ID) \). To encrypt for a particular user with identity \( ID \), we only need to know \( (G, xG) \) and \( ID \). All the procedures in the setup and private key extraction phases are centralized and performed by the PKG. We show in the following a distributed version of the PKG whose tasks are fully distributed among all users in an ad hoc group. The corresponding protocol is called Distributed Private Key Generation (DPKG).

### IV. DISTRIBUTED PRIVATE KEY GENERATION (DPKG) FOR IBC

The DPKG consists of two phases — distributed key generation (DKG) and threshold private key extraction (TPKE). In the first phase, the founding members of an ad hoc group jointly compute the private key \( x \) of the PKG and publish the corresponding public key. In the second phase, each new-coming member with an identity \( ID \) needs to obtain shares from a sufficient number of founding members of the group in order to construct his private key for \( ID \). Note that this share is only for constructing the private key of \( ID \) and has nothing to do with the shares of PKG’s private key held by the founding members. Besides, only founding members could generate a share and other members do not have the same privilege. In other words, it is possible to distinguish between a founding member and a member who has joined afterwards. The founding members, when running the DPKG phase, do not need authentication. They actually share the task of the counterpart of the root certificate authority of PKI in IBC.

We assume that a private channel exists between each pair of users, which may be established through key agreement between the peers (possibly with side-channel authentication such as physical presence); in some scenarios such as peer-to-peer or anonymous networks, an authenticated channel is not necessary, and simple Diffie-Hellman key exchange could be used. This secret could be discarded after the communication, and the non-interactive advantage of IBC sets in afterwards and the user is not restricted to communicating with others met previously. A user only needs to store its own private key and the public key of PKG. In contrast, if a shared secret is used for communication or further entity authentication, the user needs additional storage of one secret per other user, which could lead to considerable overhead; if key exchange is used to establish the session key later on, interaction between the communicating parties is still needed. In addition, for a member not met before, interaction is necessary for both cases. While the same level of non-interaction could be achieved through stored public keys and certificates of other nodes, the storage overhead is again an issue.

### A. Distributed Key Generation

Assume there are \( n \) users: \( P_1, P_2, \ldots, P_n \). Assume that the associated groups of a bilinear pairing are known. Let them be \( G_1 \) and \( G_2 \) of order \( q \), that is, the pairing is \( \hat{e} : G_1 \times G_1 \to G_2 \). A function \( H \) is used to map an identity \( ID \in \{0, 1\}^* \) to \( G_1 \).

The DKG construction is an \( n \) parallel run of Feldman’s verifiable secret sharing [7]. Each user \( P_i \) picks a random secret \( x_i \) and the resulting private key for the group is \( x = \sum_{i=1}^{n} x_i \) and the corresponding public key is \( Y = xG \) for some randomly picked generator \( G \in G_1 \). The DKG runs as follows:

1. A generator \( G \in G_1 \) is randomly chosen by either one player or all players. Joint computation could simply be achieved by summing the generators picked by all players.

2. Each player \( P_i \) randomly picks a secret \( x_i \in \mathbb{Z}_q^* \) and computes \( Y_i = x_iG \). \( P_i \) sets \( a_{i0} = x_i \) and chooses a random polynomial \( f_i(z) \) over \( \mathbb{Z}_q \) of degree \( t - 1 \) as follows:

   \[
   f_i(z) = a_{i0} + a_{i1}z + \ldots + a_{i(t-1)}z^{t-1}
   \]

   \( P_i \) broadcasts \( A_{ik} = a_{ik}G \) for \( k \in [0, t-1] \). Note that \( a_{i0} = Y_i \), \( P_i \) computes the share \( s_{ij} = f_i(j) \mod q \) for \( j \in [1, n] \) and sends \( s_{ij} \) secretly to player \( P_j \).

3. Each \( P_j \) verifies the shares he received from other players by checking for \( i = 1, \ldots, n \):

   \[
   s_{ij}G = \sum_{k=0}^{t-1} j^k A_{ik}.
   \]

   If the check fails for an index \( i \), \( P_j \) broadcasts a complaint against \( P_i \).

4. If \( t \) or more players players complain against a player \( P_i \), then \( P_i \) is considered as faulty and disqualified. Otherwise, \( P_i \) reveals the share \( s_{ij} \) for each complaining player \( P_j \). If any of the revealed shares fails the check again, \( P_i \) is disqualified. The secret stored
by a disqualified player $P_i$ is set to $x_i = 0$ and $Y_j$ equal to the identity element in $G_1$. The set of non-disqualified players is denoted by $QS$.

Step 5. The public key $Y = \sum_{i \in QS} Y_i$ and the share of secret $x$ for $P_j$ is $w_{ij} = \sum_{i \in QS} s_{ij}$. The public verification values are: $A_k = \sum_{i \in QS} A_{ik}$ for $k = 1, \ldots, t - 1$. Note that $A_0 = Y$.

Given any $t$ users $P_1, P_2, \ldots, P_t$, the secret $x$ can be reconstructed as $x = \sum_{i=1}^t L_i(0)w_{i}$ where $L_i(j) = \prod_{1 \leq r \leq t, r \neq i} \frac{w_r}{w_i}$ is the Lagrange coefficient for $P_i$ and $w_{i}$ is the secret share of $P_i$. Note that $w_i G = \sum_{k=0}^t i^k A_k$ for $i \in [1, n]$.

B. Threshold Private Key Extraction

In Boneh-Franklin’s IBE [3], the private key for an identity $ID$ is $xH(ID)$ where $x$ is the server private key. When a new member with identity $ID$ wishes to obtain his private key, he needs to obtain shares from $t$ founding members of the group. The private key extraction is as follows:

Step 1. The new member $P_{\text{new}}$ obtains a share from $P_{i}$ for $i \in [1, t]$:

$$\sigma_{ii} = w_{i} H(ID),$$

Step 2. $P_{\text{new}}$ can check the validity of $\sigma_{ii}$ as follows:

1) Compute $w_{i} G = \sum_{k=0}^{t-1} i^k A_k$.
2) Check $\hat{e}(w_{i} G, H(ID)) = \hat{e}(G, \sigma_{ii})$. $\sigma_{ii}$ is valid only if equality holds.

Step 3. To reconstruct its private key $xH(ID)$, $P_{\text{new}}$ computes $xH(ID) = \sum_{i=1}^t L_i(0)\sigma_{ii}$.

C. Security

The security of the share $w_{i}$ of the PKG’s private key $x$ held by a founding member $P_{i}$ is assured in the process of TPKE, since each $\sigma_{ii}$ would not leak out $w_{i}$ due to the difficulty to compute discrete logarithm in the group $G_1$ as implied by the hardness assumption of the CBDH problem. The hardness of the discrete logarithm problem also ensure that a non-founding member cannot use the shares obtained from the founding members for his private key $xH(ID)$ to obtain any share of $x$. The secret sharing scheme assures that at least $t$ founding members are needed to construct the PKG secret key $x$ or issue a valid secret key for a given identity $ID$. The hardness of the discrete logarithm problem ensure that no founding member could cheat or obtain more share or information of $x$ than other founding members from the exchanges messages including $A_{ik}$.

D. Discussion

When a user has obtained its private key, its identity, say $ID$, becomes authenticated; anyone could run an identification protocol with him to verify whether he really possesses the necessary private key for the identity in question. The advantage of using IBE in MANET over other public key encryption schemes is that no explicit public key certification and hence no interaction is needed when a user wants to send to the other a message. That is, users could communicate even with a unidirectional channel. The guarantee in DPKG is that the legitimacy of a user is certified by at least $t$ founding members which grant him admission to the group. It should be noted that established private channels, say through pairwise key agreement, are needed in the initial key generation phase in DPKG, whereas, the scheme presented in the next section does not rely on established infrastructure.

V. Conclusions

Self-organization and distributed trust is the main objective of any key/trust management scheme for MANETs. Even though identity based cryptography has the advantage that is particularly suitable for unidirectional channels in MANETs, the necessity of a trusted server for private key generation is a great limitation. In this article, a construction for establishing initial trust in a distributed manner is given — a distributed private key generator for IBC based on Feldman’s verifiable secret sharing scheme. This would considerably increase the usability of IBC for MANETs.

REFERENCES