Fifteenth International Workshop on Algebraic and Combinatorial Coding Theory
June 18-24, 2016, Albena, Bulgaria *pp. 111–116*

An efficient certificateless key management architecture to solve IBE and PKI issues

Abderrahman Daif adaif@assystem.com
Assystem AE&OS, 23 place de Wicklow 78180 Montigny le Bretonneux, France.
Paris8 university, 2 Rue de la Libert, 93526 Saint-Denis, France.
Cédric Tavernier[1] ctavernier@assystem.com
Assystem AE&OS, 23 place de Wicklow 78180 Montigny le Bretonneux, France.

Abstract. Among the key management infrastructures, we distinguish certificate based infrastructure (public key infrastructure (PKI), SPKI, PGP...) and IBE (Identity Based Encryption). Each one has its own advantages and disadvantages. For PKI, huge drawbacks come from the management of certificates, revocation, confidence...[8, 7]. Regarding the Boneh and Franklin's IBE [6] two main drawbacks were: the key escrow problem and the fact that all the authority is attributed to a single entity which is the *PKG* (Private Key Generator) that could usurp the identity of each user.
In this article we present a new key management architecture which combines the advantages of a PKI and Boneh and Franklin IBE scheme. This architecture is based on known mathematical operations on elliptic curves pairing (Optimal-Ate pairing [9] on the Barreto Naehrig curve [3]), and basic operations in finite fields.

1 Introduction

A key management infrastructure is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, revoke users identity and manage public-key encryption. The purpose of such system may be to facilitate the secure electronic transfer of information. It is required for activities that require a better protection than a simple password.

These infrastructures are constructed with these main functionalities:
Enrolling: corresponds to the registration of the users;
Rekeying: corresponds to the re-initialization of all private keys;
Revocation: aims to cancel the registration of an user.
The architecture that we present in this paper is based on IBE. It was introduced as an open problem by Adi Shamir in 1984 [6], its purpose was to facilitate the certificates management of e-mail. The most commonly used PKI requires a lot of memory space, and a lot of computation: regarding signature, verification for every communication.

In order to mitigate these constraints in 2001 Boneh and Franklin [2] presented an effective IBE based on the bilinear Diffie-Hellman problem. However

[1] This work was partially supported by SCISSOR ICT project no. 644425 H2020 Framework Program.

this system has several weaknesses compared to a PKI: it requires having an absolute trust in a master entity called PKG (Private Key Generator). Unfortunately this PKG can potentially usurp the identity of users (inherent key escrow problem).

Sattam and Paterson[1] made a certificateless public key scheme that allows any valid entity to construct its own private and public keys. This scheme solves the certificate issue, but in the same time doesn't solve completely the key escrow problem, as explained in the article [4]. Also, this scheme requires that each key update must be done over a private channel.

Lee et al. [5] gave another IBE diagram to limit the PKG's authority by introducing several key protection authorities, but that doesn't address the key escrow issue [4]. In this article we present an improvement of this scheme, which can withstand various attack scenarios that we shall explain later.

2 The key management issues

2.1 Public key infrastructure

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption.

It is also an arrangement that binds public keys with respective identities of entities (persons, organizations). The binding is established through a process of registration and issuance of certificates at and by a certificate authority (CA). The RA (registration authority) validates the correctness of the registration. It is responsible for accepting requests for digital certificates and authenticating the entity making the request. An entity must be uniquely identifiable within each CA domain on the basis of information about that entity. A third-party validation authority (VA) can provide this entity information on behalf of the CA (see figure 1). Unfortunately PKI owns well known identified weaknesses:

- Revoked certificates must be stored in a certificate revocation list (CRL), wich means that the CRL becomes a huge increasing list that contains old certificates stored for years;
- Secret key is not generated by the user, which means that if RA has been corrupted, it can decrypt user's messages;
- The CA can sign a fake certificates;
- Enrolment and rekeying needs to be done via a private channel.

2.2 IBE an alternative solution

Identity-based systems allow each party to generate a public key from a known identity value such as an ASCII string. A trusted third party (PKG) generates the corresponding private key. To operate, the PKG first publishes a master public key, and retains the corresponding master private key (referred as a

master key). Given the master public key, any party can compute a public key corresponding to an identity id by combining the master public key with the identity value. To obtain a corresponding private key, the party authorized to use the identity id contacts the PKG, that uses the master private key to generate the id's private key. As a result, parties may encrypt messages (or verify signatures) with no prior distribution of keys between individual participants. This is extremely useful in cases where pre-distribution of authenticated keys is inconvenient or infeasible due to technical constraints. However, to decrypt or sign messages, the authorized user must obtain the appropriate private key from the PKG.

This architecture solves the revocation problem, it doesn't require a CRL to revoke entities. It also doesn't require a validation authority (VA) to check the authenticity of a public key. All the steps, enrolment, revocation and rekeying are executed by the PKG, which represents a vulnerability point. It may be sensitive to malicious attacks(key escrow). Finally we note that quantity of network traffic between users and PKG may be a problem (see figure 1).

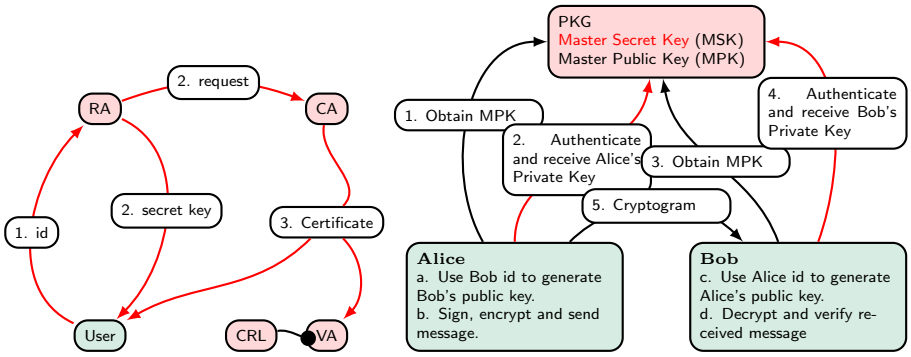

Figure 1: Comparison between the PKI scheme (left) and Boneh and Franklin IBE (right)

3 IBE2, the proposed solution

In our proposed scheme we use the trick considered in [5] that consists in involving a new authority called PKG_{st} (static PKG) which is the key protection authority, and we call PKG_{dy} (dynamic PKG), the usual PKG. Among the advantages of this new scheme, we note that now, the users contribute in the generation of the secret key in a sense that only him owns it.

In [4] Chunxiang et al. proved that when we have collusion between the key protection authority and an user the key escrow problem was not really solved. Indeed, they explain that if there is several authorities, the user must authenticates by each one of them. To solve this issue we set up tow authorities:

- Static PKG: is an off-line server which contributes only on the enrolment phase. Its role is: authenticating the user, providing its partial secret key, and signing this key to prove that it was forged by himself.
- Dynamic PKG: is an on-line server which also contributes in the creation of user's secret key. Its role is: participation in the construction of the user's secret key and rekeying.

3.1 The system parameters

This IBE system works also with an asymmetric pairing as the optimal Ate pairing with the Barreto-Naehrig curves [3]. We remind that computing the discrete logarithm is a difficult problem for such curves. Here is the setup parameters of the encryption system (red variables are secret):

- The identity of the authorities id_{dy} and id_{st};
- $\mathbb{G}_0, \mathbb{G}_1$ and $\mathbb{G}_T$ cyclic groups with a prime order q;
- $\mathcal{M}$ the set of messages;
- An asymmetric bilinear function $e : \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_T$;
- P and P' generators of $\mathbb{G}_0$ and $\mathbb{G}_1$ respectively;
- T keys lifetime;
- Two hash functions: $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1^*$ and $\mathcal{H}_2 : \mathbb{G}_T \to \mathcal{M}$;

The initialization parameters of the system requires the flowing operations:

- PKG_{st} picks randomly a secret key $s_0 \in \mathbb{Z}_q$;
- PKG_{st} computes its public key: $P_0 = s_0 P \in \mathbb{G}_0$;
- PKG_{dy} picks randomly a secret key $s_1 \in \mathbb{Z}_q$;
- PKG_{dy} computes its public keys $P_1 = s_1 P \in \mathbb{G}_0$ and $P_1' = s_1 P' \in \mathbb{G}_1$;
- The system public key is given by $Y = s_1 P_0 = s_0 s_1 P \in \mathbb{G}_0$;

An arbiter (a judge for example) can control the correctness of the public key by checking if $e(Y, P') = e(P_0, P_1')$ is satisfied.

3.2 Enrollment step

We assume here that a user id wants to be registered in the organization. In this case, the key generation takes place in the following steps:

1. Off line authentication of the user by PKG_{st}:
   - Identification of the user by PKG_{st};
   - PKG_{st} computes the public key of id: $Q_{id} = \mathcal{H}_1(id|id_{PKGs}) \in \mathbb{G}_1$;
   - PKG_{st} computes the partial secret key $S_p = s_0 Q_{id} \in \mathbb{G}_1$;
   - PKG_{st} signs S_p: $\text{Sign}(S_p) = s_0 S_p \in \mathbb{G}_1$;
   - PKG_{st} sends $(S_p, \text{Sign}(S_p))$ to the user;
   - The user checks if $e(P, Sign(S_p)) \stackrel{?}{=} e(P_0, S_p)$ is satisfied. (1)
2. Computation of a partial secret key by the user:
   - The user picks randomly the tempory secret mask $t \in \mathbb{Z}_q$;
   - The user computes the partial secret key $S_{p,t} = tS_p = ts_0 Q_{id} \in \mathbb{G}_1$;
   - The user signs the partial secret key $Sign(S_{p,t}) = tSign(S_p) \in \mathbb{G}_1$;

- The user sends $(id, S_{p,t}, Sign(S_{p,t}), P_{0,t} = tP_0)$ to PKG_{dy};

3. On line authentication of the user by PKG_{dy}:
   - PKG_{dy} controls if :
     - $e(P, Sign(S_{p,t})) = e(P_0, S_{p,t})$; (2)
     - $e(P_{0,t}, Q_{id}) = e(P, S_{p,t})$; (3)
   - PKG_{dy} computes $S_t = s_1 S_{p,t} = t s_0 s_1 Q_{id}$;
   - PKG_{dy} signs S_t: $Sign(S_t) = s_1 S_t$;
   - PKG_{dy} sends $(S_t, Sign(S_t))$ to the user;
4. The user's secret key extraction:
   - The user checks if $e(P, Sign(S_t)) = e(P_1, S_t)$; (4)
   - The user extracts his own secret key $S_{id} = \frac{1}{t} S_t = s_0 s_1 Q_{id}$.
   - The user tests: $e(Y, Q) \stackrel{?}{=} e(P, S_{id})$; (5)

The exchanges between the user and $PKGs$ are signed. It allows a secure key exchange between the user and the authorities, and it can be checked through the following tests:

- The test (1) gives the proof to the user that his partial secret key S_p has been constructed by the PKG_{st} this prevents against identity usurpation of PKG_{st}.
- The test (2) gives the proof to the PKG_{dy} that the partial masked secret key $S_{p,t}$ that he received form the user id has been constructed by PKG_{st}. It also proves that the user id has been authenticated by PKG_{st}. Thus it prevents against identity usurpation.
- The test (3) gives the proof to the PKG_{dy} that the request is made by the user id.
- In the same way, the test (4) gives the proof to the user that the masked secret key S_t has been constructed by PKG_{dy}.
- Finally, the test (5) proves that the user's secret key S_{id} has been constructed by PKG_{st} and PKG_{dy}.

The figure 2 summarizes the key distribution between PKG_{dy}, PKG_{st} and the user.

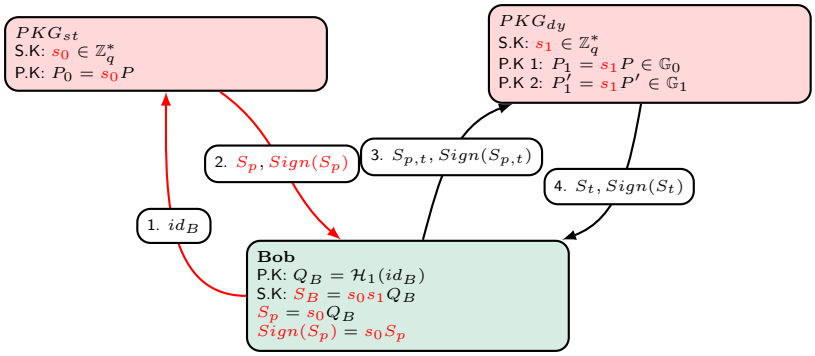

Figure 2: Key distribution in IBE-2

3.3 The rekeying and revocation step

To avoid that revoked entities decipher messages, we have to update keys in a regular way every T days. For that purpose, the PKG_{dy} which is responsible of this task, updates keys (s_1, P_1, P'_1, Y). Then each entity restart stages 2, 3 and 4 of the enrolment steps.

PKG_{dy} is a server that changes periodically its secret key s_1 while PKG_{st} secret key s_0 does not change. This secret key s_0 has to be protected in a such way that a lawyer authority (according rules of certain countries) could access in case of corruption (for example).

For our IBE scheme, we add a temporary identity revocation list (IRL) that lists the revoked entities identity. This list must respect those proprieties:

- Must be cleared (erased) at each rekeying step. Whence, we can revoke the entities in real time without increasing constantly the IRL;
- The IRL is a reading only file for which only the security officer can add the revoked identities;
- The IRL does not require to store keys, but only the users *id*.
- The period T must be chosen in order to make the IRL nearly empty a most of the time, so it will not affect the network traffic fluidity.

References

[1] Sattam S. Al-Riyami and Kenneth G. Paterson, Certificateless Public Key Cryptography, *asiacrypt*, 2003.

[2] Dan Boneh and Matthew K. Franklin, Identity-Based Encryption from the Weil Pairing Advances in Cryptology, *Proceedings of CRYPTO*, 2001.

[3] Paulo S. L. M. Barreto and Michael Naehrig, Pairing-friendly elliptic curves of prime order, *Springer*, 2006.

[4] XU Chunxiang, ZHOU Junhui and QIN Zhiguang, A Note on Secure Key Issuing in ID-based Cryptography.

[5] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang and Seungjae Yoo, Secure Key Issuing in ID-based Cryptography.

[6] Adi Shamir, Identity-Based Cryptosystems and Signature Schemes *Advances in Cryptology: Proceedings of CRYPTO*, 1984.

[7] Peter Gutmann, PKI: Its Not Dead, Just Resting, August, 2002.

[8] Carl Ellison and Bruce Schneier, Ten Risks of PKI: What Youre not Being Told about Public Key Infrastructure, *Computer Security Journal*, 2000.

[9] F. Vercauteren, Optimal Pairings, *eprint*, 2008.