Certificate Based Signature Schemes without Pairings or Random Oracles

Joseph K. Liu, Joonsang Baek, Willy Susilo and Jianying Zhou

Cryptography and Security Department

Institute for Infocomm Research, Singapore

{ksliu, jsbaek, jyzhou}@i2r.a-star.edu.sg

Centre for Computer and Information Security (CCISR)

School of Computer Science and Software Engineering

University of Wollongong, Australia

wsusilo@uow.edu.au

Outline

- Motivation
  - Public Key Cryptography and PKI
  - Identity-based Cryptography
  - Certificate-based Cryptography
  - Certificateless cryptography
- Related Works
- Our Contributions
- CBS Security Model
- CBS Without Pairings
- CBS Without Random Oracles
- Conclusions

Public Key Cryptography

- Users have two keys: public key and private key
- Alice's public key is used for:
  - encryption to Alice by Bob
  - verification of Alice's signatures by Bob
- Alice's private key is used for:
  - decryption by Alice
  - signing by Alice
- An infrastructure linking users and their public keys is needed: Public Key Infrastructure

Public Key Cryptography: Drawbacks

- Certification management adds a heavy overhead in PKIs:
  - Certificate distribution
  - Certificate storage
  - Certificate revocation
  - Cross-certification
- Certificates seem to be a problem. What about removing certificates as much as possible?

Identity-based Cryptography (Shamir 1984)

- Public keys are derived in a public way from users' identities
- Users must contact a trusted authority (TA) prior to decryption/signing. TA identifies the users and compute private keys using a master-key
- Bob can encrypt to Alice w/o consulting a directory
- Alice does not even have to be registered before Bob sends her a message
- Only the system parameters need to be certified (considerably less certificates to be handled)

Identity-based Cryptography: Security Definitions

- In the identity-based scenario, the adversary can corrupt a set of players and learn their private keys. Additionally, it can adaptively ask for encryption (signing) queries for any user
- IBE: chosen-ciphertext security means the adversary can not learn anything about non-queried ciphertexts for non-corrupted users
- IBS: unforgeability means the adversary can not forge non-queried signatures for non-corrupted users
- IBS can be built from *any* signature scheme
- IBE more challenging, and remained an open problem until 2001

IBC: Advantages & Disadvantages

- Advantages:
  - Certificates drastically reduced
  - Human memorizable keys
  - No need for public key directories
- Drawbacks:
  - Key escrow
  - Distributing keys is non-trivial (secure channel)
  - Certificate revocation?
    - Solution: time is added to identifiers (not ideal)

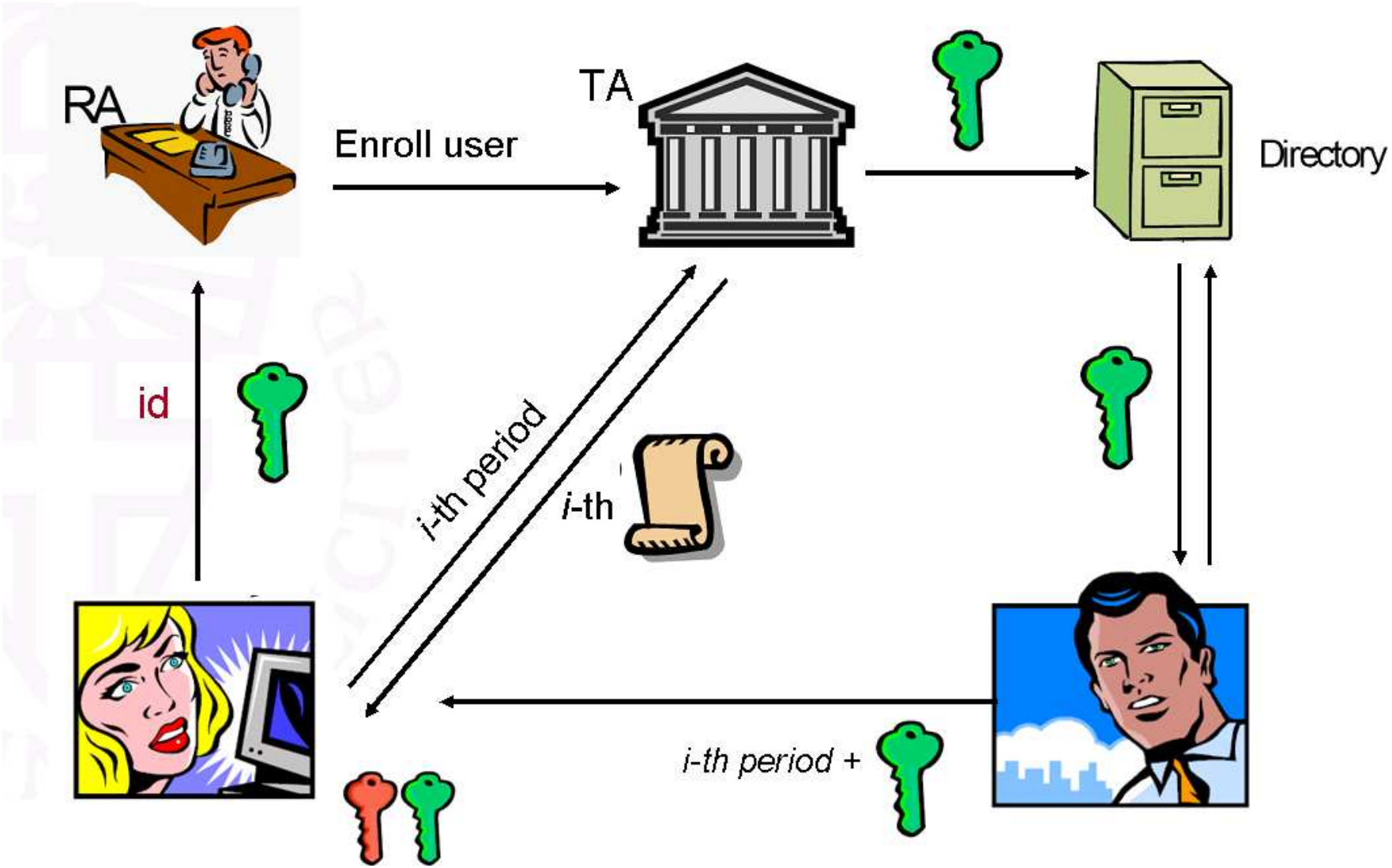
How to remove key escrow?

- Certificate-based cryptography (Gentry 2003)
  - Goal: to remove key escrow from IBC and simplify the certificate revocation strategy
  - Idea:
    - Bob encrypts/verifies adding time periods
    - Alice needs both her secret key and a certificate for each time period to decrypt/sign
    - revocation means the TA stops issuing certificates
    - **no** secure channel
- Certificate-less cryptography (AlRiyami-Patterson 2003)
  - Goal: Bob does not need to check whether Alice's public key is certified (no certificates)
  - Idea:
    - Alice's decryption key is built as a result of the interaction between Alice and the TA secure channel

Certificateless Cryptography

- Pairing based
- *Without pairing* (Baek, Safavi-Naini, Susilo - ISC 2005).

Certificate-based Cryptography



Certificate-based Signature Schemes

- Firstly suggested by Kang, Park and Hahn (CT-RSA 2004).
- Insecure against key replacement attack (Li, Huang, Mu, Susilo and Wu - EuroPKI 2007 & Journal of Computer Security 2008).
- Certificate-based ring signature - Au, Liu, Susilo, Yuen - ISPEC 2007.
- *All* schemes require pairings and random oracles.

Our Contributions

- We propose two *new* CBS schemes:
  - A scheme *without* pairings.
  - A scheme *without* random oracles.
- These schemes fill the gap of the CBS constructions in the literature.

CBS Security Model

A CBS scheme is defined by six algorithms:

- Setup is a probabilistic algorithm taking as input a security parameter. It returns the certifier's master key msk and public parameters $param$. Usually this algorithm is run by the CA.
- UserKeyGen is a probabilistic algorithm that takes $param$ as input. When run by a client, it returns a public key PK and a secret key usk.
- Certify is a probabilistic algorithm that takes as input $(msk, \tau, param, PK, ID)$ where ID is a binary string representing the user information. It returns $Cert'_\tau$ which is sent to the client. Here τ is a string identifying a time period.

CBS Security Model

- Consolidate is a deterministic certificate consolidation algorithm taking as input $(param, \tau, Cert'_\tau)$ and optionally $Cert_{\tau-1}$. It returns $Cert_\tau$, the certificate used by a client in time period τ.
- Sign is a probabilistic algorithm taking as input $(\tau, param, m, Cert_\tau, usk)$ where m is a message. It outputs σ.
- Verify is a deterministic algorithm taking $(param, PK, ID, \sigma)$ as input in time period τ. It returns either valid indicating a valid signature, or the special symbol $\perp$ indicating invalid.

CBS Security Model

We require

$$\text{Verify}_{PK,ID}(\text{Sign}_{\tau,Cert_\tau,usk}(m)) = \text{valid}$$

We also note that a concrete CBS scheme may not involve certificate consolidation. In this situation, algorithm Consolidate will simply output $Cert_\tau = Cert'_\tau$.
In the rest of this paper, for simplicity, we will omit Consolidate and the time identifying string τ in all notations.

CBS Security Model

Two different security games:

- `Game 1`: the adversary models an uncertified entity
- `Game 2`: the adversary models the certifier in possession of the master key msk attacking a fixed entity's public key

We use the enhanced model by Li, Huang, Mu, Susilo and Wu that captures the key replacement attack in `Game 1`.

CBS Without Pairing

Setup. Let $\mathbb{G}$ be a multiplicative group with order q. The PKG selects a random generator $g \in \mathbb{G}$ and randomly chooses $x \in_R \mathbb{Z}_q^*$. It sets $X = g^x$. Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ be a cryptographic hash function. The public parameters param and master secret key msk are given by

$$\text{param} = (\mathbb{G}, q, g, X, H) \qquad msk = x$$

UserKeyGen. User selects a secret value $u \in \mathbb{Z}_q^*$ as his secret key usk, and computes his public key PK as (g^u, X^u, π_u) where π_u is the following non-interactive proof-of-knowledge (PoK):

$$PK\{(u) : U_1 = g^u \wedge U_2 = X^u\}$$

CBS Without Pairing

Certify. Let $\tilde{h} = H(PK, ID)$ for user with public key PK and binary string ID which is used to identify the user. To generate a certificate for this user, the CA randomly selects $r \in_R \mathbb{Z}_q^*$, computes

$$R = g^r \qquad s = r^{-1}(\tilde{h} - xR) \bmod q$$

The certificate is (R, s). Note that a correctly generated certificate should fulfill the following equality:

(1)
$$R^s X^R = g^{\tilde{h}}$$

CBS Without Pairing

Sign. To sign a message $m \in \{0,1\}^*$, the signer with public key PK (and user info ID) , certificate (R, s) and secret key u, randomly selects $y \in_R \mathbb{Z}_q^*$, computes

$$Y = R^{-y} \qquad h = H(Y, R, m) \qquad z = y + h\ s\ u \bmod q$$

and outputs (Y, R, z) as the signature σ.

Verify. Given a signature $\sigma = (Y, R, z)$ for a public key PK on a message m, a verifier first checks whether π_u is a valid PoK. If not, output $\perp$. Otherwise computes $h = H(Y, R, m)$, $\tilde{h} = H(PK, ID)$, and checks whether

(2) $$(g^u)^{h\tilde{h}} \stackrel{?}{=} R^z Y (X^u)^{hR}$$

Output valid if it is equal. Otherwise, output $\perp$.

CBS Without Pairing

Theorem 1 (Unforgeability against Game 1 Adversary) *The CBS scheme without pairing is (ϵ, t)-existential unforgeable against Game 1 adversary with advantage at most ϵ and runs in time at most t, assuming that the (ϵ', t')-DL assumption holds in $\mathbb{G}$, where*

$$\epsilon' = \left(1 - \frac{q_h(q_e + q_s)}{q}\right)\left(1 - \frac{1}{q}\right)\left(\frac{1}{q_h}\right)\epsilon, \qquad t' = t + \mathcal{O}(q_e + q_s)E$$

and q_e, q_s, q_h are the numbers of certification queries, signing queries and hashing queries the adversary is allowed to make and E is the time for an exponentiation operation.

CBS Without Pairing

Theorem 2 (Unforgeability against Game 2 Adversary) *The CBS scheme without pairing is (ϵ, t)-existential unforgeable against Game 2 adversary with advantage at most ϵ and runs in time at most t, assuming that the (ϵ', t')-DL assumption holds in $\mathbb{G}$, where*

$$\epsilon' = \left(1 - \frac{q_h q_s}{q}\right)\left(1 - \frac{1}{q}\right)\left(\frac{1}{q_h}\right)\left(\frac{1}{q_u}\right)\epsilon, \qquad t' = t + \mathcal{O}(q_s)E$$

and q_s, q_h, q_u are the numbers of signing queries, hashing queries and user-key-gen queries the adversary is allowed to make and E is the time for an exponentiation operation.

CBS Without Random Oracles

Our scheme is motivated by Waters IBE. Let $H_u : \{0,1\}^* \rightarrow \{0,1\}^{n_u}$ and $H_m : \{0,1\}^* \rightarrow \{0,1\}^{n_m}$ be two collision-resistant cryptographic hash functions for some $n_u, n_m \in \mathbb{Z}$.

Setup. Select a pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ where the order of $\mathbb{G}$ is p. Let g be a generator of $\mathbb{G}$. Randomly select $\alpha \in_R \mathbb{Z}_p$, $g_2 \in_R \mathbb{G}$ and compute $g_1 = g^\alpha$. Also select randomly the following elements:

$$u', m' \in_R \mathbb{G} \qquad \hat{u}_i \in_R \mathbb{G} \text{ for } i = 1, \ldots, n_u \qquad \hat{m}_i \in_R \mathbb{G} \text{ for } i = 1, \ldots, n_m$$

Let $\hat{U} = \{\hat{u}_i\}, \hat{M} = \{\hat{m}_i\}$. The public parameters param are $(e, \mathbb{G}, \mathbb{G}_T, p, g, g_1, g_2, u', \hat{U}, m', \hat{M})$ and the master secret key msk is g_2^α.

UserKeyGen. User selects a secret value $x \in \mathbb{Z}_p$ as his secret key usk, and computes his public key PK as $(\mathsf{pk}^{(1)}, \mathsf{pk}^{(2)}) = (g^x, g_1^x)$.

CBS Without Random Oracles

Certify. Let $\mathfrak{u} = H_u(PK, ID)$ for user with public key PK and binary string ID which is used to identify the user. Let $\mathfrak{u}[i]$ be the i-th bit of $\mathfrak{u}$. Define $\mathcal{U} \subset \{1, \ldots, n_u\}$ to be the set of indicies such that $\mathfrak{u}[i] = 1$.
To construct the certificate, the CA randomly selects $r_u \in_R \mathbb{Z}_p$ and computes

$$\left(g_2^{\alpha} \left(U\right)^{r_u}, g^{r_u} \right) = (\text{cert}^{(1)}, \text{cert}^{(2)}) \qquad \text{where } U = u' \prod_{i \in \mathcal{U}} \hat{u}_i$$

CBS Without Random Oracles

Sign. To sign a message $m \in \{0,1\}^*$, the signer with identity PK (and user information ID) , certificate $(\text{cert}^{(1)}, \text{cert}^{(2)})$ and secret key usk, compute $\mathfrak{m} = H_m(m)$. Let $\mathfrak{m}[i]$ be the i-th bit of $\mathfrak{m}$ and $\mathcal{M} \subset \{1, \ldots, n_m\}$ be the set of indicies i such that $\mathfrak{m}[i] = 1$. Randomly select $r_\pi, r_\mathfrak{m} \in_R \mathbb{Z}_p$, compute $\mathfrak{u} = H_u(PK, \text{userinfo})$, $U = u' \prod_{i \in \mathcal{U}} \hat{u}_i$ and

$$\sigma = \left(\left(\text{cert}^{(1)}\right)^{usk} \left(U\right)^{r_\pi} \left(m' \prod_{i \in \mathcal{M}} \hat{m}_i\right)^{r_\mathfrak{m}} , \left(\text{cert}^{(2)}\right)^{usk} g^{r_\pi} , g^{r_\mathfrak{m}} \right)$$

$$= (V, R_\pi, R_\mathfrak{m})$$

Verify. Given a signature $\sigma = (V, R_\pi, R_\mathfrak{m})$ for a public key PK and user information ID on a message m, a verifier first checks whether $e(g^x, g_1) = e(g_1^x, g)$. If not, outputs $\perp$. Otherwise computes $\mathfrak{m} = H_m(m)$, $\mathfrak{u} = H_u(PK, ID)$, $U = u' \prod_{i \in \mathcal{U}} \hat{u}_i$ and checks whether

$$e(V, g) \stackrel{?}{=} e(g_2, g_1^x)\, e(U, R_\pi)\, e(m' \prod_{i \in \mathcal{M}} \hat{m}_i, R_\mathfrak{m})$$

CBS Without Random Oracles

Theorem 3 (Unforgeability against Game 1 Adversary) *The CBS scheme without random oracles is (ϵ, t)-existential unforgeable against Game 1 adversary with advantage at most ϵ and runs in time at most t, assuming that the (ϵ', t')-GCDH assumption holds in $\mathbb{G}$, where*

$$\epsilon' \geq \frac{\epsilon}{16(q_e + q_s)(n_u + 1)q_s(n_m + 1)}$$

$$t' = t + O\Big(\big(q_e n_u + q_s(n_u + n_m)\big)\rho + (q_e + q_s)\tau\Big)$$

where q_e is the number of queries made to the Certification Query, q_s is the number of queries made to the Signing Query, and ρ and τ are the time for a multiplication and an exponentiation in $\mathbb{G}$ respectively.

CBS Without Random Oracles

Theorem 4 (Unforgeability against Game 2 Adversary) *The CBS scheme without random oracles is (ϵ, t)-existential unforgeable against Game 2 adversary with advantage at most ϵ and runs in time at most t, assuming that the (ϵ', t')-Many-DH assumption holds in $\mathbb{G}$, where*

$$\epsilon' \geq \frac{\epsilon}{16 q_s (n_u + 1) q_s (n_m + 1) q_k}, \; t' = t + O\Big(\big(q_s (n_u + n_m)\big)\rho + (q_k + q_s)\tau \Big)$$

where q_s is the number of queries made to the Signing Queries, q_k is the number of queries made to the User-key-gen Queries and ρ and τ are the time for a multiplication and an exponentiation in $\mathbb{G}$ respectively.

Conclusions

- We proposed two new CBS schemes.
- The first scheme does not require any pairings.
  - Suitable for some power-constrained devices, such as wireless sensor networks.
- The second scheme does not require any random oracles.
  - Suitable for applications that require a high level of security.

Full Paper

Full version of the paper can be found at
`http://eprint.iacr.org/2008/275`.