Certificateless Cryptography I

Kenny Paterson

Information Security Group

Royal Holloway, University of London

kenny.paterson@rhul.ac.uk

June 2007

Overview

This talk and the next:

- Motivating and introducing certificateless (public key) cryptography
- Certificateless Encryption (CLE) and its security
- Specific and generic constructions for CLE
- CLE and Gentry's CBE
- Malicious KGCs and Denial-of-Decryption Attacks
- Mediated certificateless encryption, certificateless signatures (CLS) and further certificateless primitives
- Conclusions and open problems

1 Motivation

What problems does ID-PKC solve?

- Eliminates certificate chains and certificate verification.
- No public keys need be stored/transmitted if context strong enough to define identities (e.g. e-mail address, IP address).
- TA controls issuance of private keys, leading to concept of cryptographic workflow.
- Revocation can be handled by appending time period to identities and including revocation policy in TA's public parameters.
- Suited to "closed" applications where there is a clear choice for the TA, and repudiation is not an issue.

Motivation

What problems does ID-PKC still have?

- Master secret (c.f. CA root signing key) and single point of failure at TA.
- Built-in key escrow: the TA knows all the private keys.
  - Makes non-repudiation of identity-based signatures difficult.
- Sender/verifier needs to obtain authentic TA parameters.
- Still need proper registration procedures prior to private key distribution.
- Delivery of private keys must be over a secure channel.
- Keys may need to be revoked before their natural expiry.
  - Requiring more sophisticated revocation procedure.

Certificateless Public Key Cryptography (CL-PKC)

Is it possible to keep some of the benefits of ID-PKC (no certificates and their associated problems) without introducing key escrow by default?

- Use algebraic properties of pairings and threshold techniques to distribute master key across multiple TAs (Boneh-Franklin, 2001).
- Certificateless Public Key Cryptography (CL-PKC, Al-Riyami–Paterson, 2003).
- Goyal (2007): IBE in which existence of multiple private keys identifies TA misbehaviour.

Introducing Certificateless Public Key Cryptography (CL-PKC)

High-level description:

- Trusted third party called Key Generation Centre (KGC) with master secret and public parameters.
- Users generate their own key-pairs (c.f. ID-PKC).
- Full user private key created from user-generated private key component and KGC-supplied private key component.
- Public key based on user-generated public key and user identity.
- Key-pairs can be used for encryption, signing, key exchange, ...

Slightly More Formally:

- KGC chooses master public key mpk (aka `params`) and master secret msk.
- KGC computes *partial* private key d_{ID} from ID and msk, and delivers d_{ID} securely to correct user.
- User generates a secret value x_{ID} and a corresponding public key $pk_{\mathsf{ID}}\rangle$.
  - This step may be carried out before or after the previous step.
- User combines d_{ID} and x_{ID} to produce the full private key sk_{ID}.
- Any party, in possession of ID, pk_{ID} and mpk can encrypt to user with identity ID/verify signatures from ID, etc.

(Other formulations are possible and will be covered in due course!)

Introducing CL-PKC

- If done properly, public keys of users no longer need support of certificates.
  - Instead, confidentiality guarantees are "implicit".
  - For example, recipient cannot decrypt unless he knows *both* KGC-supplied private component and user-generated private component.
- KGC does not know full private key because of user-generated component.
  - So key escrow removed?
- But lack of certificates means public keys could be *replaced* by an adversary.
- And need for public key of user means no-longer identity-based.

Introducing CL-PKC Adversaries

Generally, there are two types of adversary against CL-PKC schemes:

Type I: Models an outsider adversary, who does not know the master secret, and may replace public keys at will.

Type II: Models an adversarial KGC, who generates mpk/params honestly, knows the master secret, and is trusted not to replace public keys of users.

- A KGC who replaces public key of a user knows all secret information associated with that user.
- Roughly equivalent (but not identical to) trust given to CA in PKI.

(Many variants are possible and will be covered in due course!)

2 Certificateless Encryption (CLE)

- We now focus on the development of certificateless encryption (CLE).
- We formally define the notion of a CLE scheme and its security.
- We then introduce the concrete CLE scheme of Al-Riyami-Paterson (2003).
- This will lead us (eventually) to generic constructions for CLE, different security models for CLE, and so on.

Formal Definition of CLE

A CLE scheme can be defined in terms of 7 algorithms:

Setup:

- Input: security parameter 1^k.
- Output: master secret key pair $\langle msk, mpk \rangle$.
- Run by KGC.
- Assume mpk includes description of key-spaces, plaintexts, ciphertexts, etc.

Formal Definition of CLE

Extract-Partial-Private-Key:

- Input: msk and an identity string $\mathsf{ID} \in \{0,1\}^*$.
- Output: partial private key d_{ID}.
- Run by KGC once for each identity (but users can have more than one identity/identifier).
- Distributed to correct user in a suitably secure manner.

Formal Definition of CLE

Set-Secret-Value:

- Input: mpk and (possibly) ID.
- Output: user secret value x_{ID}.

Set-Private-Key:

- Input: mpk, d_{ID}, x_{ID}.
- Output: full private key sk_{ID}.

Set-Public-Key:

- Input: mpk, x_{ID}.
- Output: user public key pk_{ID}.

These algorithms are run by user, typically once (but user can have more than one secret value and corresponding key-pairs).

Formal Definition of CLE

Encrypt:

- Input: mpk, ID, pk_{ID}, and plaintext M.
- Output: ciphertext C.

Decrypt:

- Input: mpk, sk_{ID}, and ciphertext C.
- Output: plaintext M or an error symbol $\perp$.

We have the obvious consistency requirement that decryption "undoes" encryption.

Formal Definition of CLE

A 5-algorithm formulation is also possible:

- Combine Set-Secret-Value, Set-Private-Key, Set-Public-Key into a single algorithm with input mpk, d_{ID} and output $\langle x_{\mathsf{ID}}, pk_{\mathsf{ID}} \rangle$.
- Provide $\langle x_{\mathsf{ID}}, d_{\mathsf{ID}} \rangle$ as an input to Decrypt in place of sk_{ID}.

The two formulations are equivalent, and which is preferred is largely a matter of taste.

Security for CLE

- Security for CLE is modelled as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.
- Model extends the IBE security game of Boneh-Franklin to include enhanced adversarial capabilities.
- How best to handle decryption queries for users whose public keys have been replaced by the adversary is a contentious question.
- This has led to a variety of different security models being proposed.
- We begin with the original security model of Al-Riyami–Paterson.

Security for CLE

Phase 1: Adversary $\mathcal{A}$ can interleave:

- decryption queries: any ciphertext, any identity;
- private key extract queries: any identity;
- partial private key extraction queries: any identity;
- request public key queries: any identity.
- replace public key queries: any identity.

Phase 2: $\mathcal{A}$ chooses messages M_0, M_1 and identity ID^*.

- $\mathcal{C}$ chooses $b \leftarrow_R \{0,1\}$ and computes C^*, the encryption of M_b using the current public key for ID^* and gives C^* to $\mathcal{A}$.

Phase 3: $\mathcal{A}$ makes more queries and finally outputs a guess b' for b. $\mathcal{A}$ is successful if $b' = b$.

Restrictions on Adversarial Behaviour

We make the following assumptions about Type I/Type II adversaries:

- No adversary can extract the private key for ID^* at any stage.
- No adversary can request the decryption of C^* for the combination of identity and public key used to encrypt M_b.
- Type I adversary cannot both replace public key for ID^* in Phase 1 and extract partial private key for ID^* in some phase.

– These prevent adversary from trivially winning security game.

Further Restrictions on Adversarial Behaviour

- Type II adversary cannot replace any public keys.
- Type II adversary assumed not to make any partial private key extract queries.
  - Because this adversary is meant to model a KGC who is trusted not to replace *any* public keys but knows master secret.
- Type I adversary cannot extract the private key for an identity if the corresponding public key has been changed.
  - Otherwise $\mathcal{C}$ has no hope of responding correctly.

Security for CLE

Given a CLE scheme and an adversary $\mathcal{A}$, we define

$$\mathrm{Adv}(\mathcal{A}) := \Pr(b' = b) - 1/2.$$

We say that the CLE scheme is IND-CCA secure if $\mathrm{Adv}(\mathcal{A})$ is negligible for any polynomial-time adversary of Type I or Type II in the above security game.

- Here "negligible" and "polynomial-time" are relative to the security parameter k used to define the scheme.
- An IND-CPA security notion for CLE follows immediately by removing access to the decryption oracle.

Decryption Queries in CLE

- The model assumes that $\mathcal{C}$ correctly responds to decryption queries for a specified user *even if the public key for that user has been replaced.*
- We refer to the corresponding Type I adversary as a *Strong Type I* adversary.
- This yields a very strong notion of security, and it has proven difficult to show that concrete schemes meet this notion.
- It can be argued that the notion is *too* strong: a user could never be forced into using a private key to which he has no access (after his public key replaced).
- It has even been argued that security against a Strong Type I adversary is not achievable in the standard model.

Decryption Queries in CLE

Weaker alternatives:

- Weak Type Ia: Bentahar *et al.* (2005) – Type I adversary supplies x_{ID} as part of decryption query.
- Weak Type Ib*: Yum-Lee (2004) – decryption queries answered using original private key if public key replaced; no partial private key extract for ID^*.
- Weak Type Ic: Baek-Wang (2006) – Type I adversary makes no public key replace queries.

Notes:

- Naming here based on Dent's survey article on CLE (early version available in eprint report 2006/211).
- All variants have been invoked by various authors.

Strengthening the Type II Adversary

- It has also been suggested to strengthen the Type II model to allow (limited) public key replacements.
- For example, allow public key replacement except on the challenge identity ID^*.
- This is generally easy to handle in proofs, but detracts from the purpose of the Type I model – which was to model a KGC who is assumed to behave honestly with respect to *all* users.
- Still, it is reasonable to seek equivalence between Type I and Type II adversaries.

Which CLE Security Model Should We Use?

- A strong model gives margin of error for security in practice, and increases the theoretical challenge (and fun).
- A weaker model makes it easier to write research papers and may lead to more efficient schemes.

3 Building CLE schemes

Do there even exist CLE schemes meeting these stringent security requirements?

Schemes are generally of two types:

- Specific schemes arising by "tweaking" existing IBE constructions.
- Schemes arising from generic constructions for CLE from other primitives (often combining IBE and PKE in some way).

Here we consider constructions of both types.

Concrete CLE schemes

The first specific CLE construction was obtained in Al-Riyami and Paterson (2003) by modifying the Boneh-Franklin IBE scheme.

Setup:

- Input: security parameter 1^k.
- Output: $\langle msk, mpk \rangle$ where

$$mpk = \langle \mathbb{G}, \mathbb{G}_T, e, p, n, P, P_0 = sP, H_1, \ldots, H_4 \rangle$$

with $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ a pairing on groups of order p, P a generator for $\mathbb{G}$, n the bit-length of plaintexts, and

$$msk = s \leftarrow_R \mathbb{Z}_p.$$

- As in Boneh-Franklin IBE scheme.

Al-Riyami–Paterson CLE

Extract-Partial-Private-Key:

- Input: msk and an identity string $\mathsf{ID} \in \{0,1\}^*$.
- Output: partial private key $d_{\mathsf{ID}} = sH_1(\mathsf{ID})$.
- Just a private key in Boneh-Franklin IBE scheme.

Al-Riyami–Paterson CLE

Set-Secret-Value:

- Input: mpk.
- Output: user secret value $x_{\mathsf{ID}} \leftarrow_R \mathbb{Z}_p$.

Set-Private-Key:

- Input: mpk, d_{ID}, x_{ID}.
- Output: full private key $sk_{\mathsf{ID}} = x_{\mathsf{ID}} d_{\mathsf{ID}} = x_{\mathsf{ID}} s H_1(\mathsf{ID})$.

Set-Public-Key:

- Input: mpk, x_{ID}.
- Output: user public key $pk_{\mathsf{ID}} = \langle x_{\mathsf{ID}} P, x_{\mathsf{ID}} sP \rangle$.

Al-Riyami–Paterson CLE

Encrypt:

- Input: mpk, ID, $pk_{\mathsf{ID}} = \langle X_{\mathsf{ID}}, Y_{\mathsf{ID}} \rangle$, and plaintext M.
- First test if $e(P, Y_{\mathsf{ID}}) = e(P_0, X_{\mathsf{ID}})$, aborting on failure.
- Set $\sigma \leftarrow_R Z_p$ and $r = H_3(\sigma, M)$.
- Output: ciphertext $C = \langle c_1, c_2, c_3 \rangle$ where

$$\begin{aligned} c_1 &= rP \\ c_2 &= \sigma \oplus H_2(e(Y_{\mathsf{ID}}, H_1(\mathsf{ID}))^r) \\ c_3 &= M \oplus H_4(\sigma) \end{aligned}$$

- Use Y_{ID} in place of P_0 in Boneh-Franklin encryption.

Al-Riyami–Paterson CLE

Decrypt:

- Input: mpk, sk_{ID}, and ciphertext $C = \langle c_1, c_2, c_3 \rangle$.
- Retrieve $\sigma' = c_2 \oplus H_2(e(c_1, sk_{\mathsf{ID}}))$.
- Retrieve $M' = c_3 \oplus H_4(\sigma')$.
- Set $r' = H_3(\sigma', M')$.
- Test if $c_1 = r'P$.
- Output: M' if the test passes; $\perp$ if it fails.

Security of Al-Riyami–Paterson CLE

Al-Riyami and Paterson (2003) proved:

Theorem 1 *The above CLE scheme is IND-CCA secure in the random oracle model, provided the* generalised BDH problem *is hard:*

On input $\langle P, aP, bP, cP \rangle$*, output a pair* $\langle Q, e(P,Q)^{abc} \rangle$*.*

- The proof is complicated, involving a delicate extension of Fujisaki-Okamoto knowledge extraction techniques.
- A new hardness assumption is needed; the generalised BDHP is not harder than BDHP.
- Scheme is not that efficient because of need to verify form of public key:

$$e(P, Y_{\mathsf{ID}}) = e(P_0, X_{\mathsf{ID}}).$$

- Can we do better?

Generic Constructions for CLE

Al-Riyami (2004); Yum-Lee (2004): combine an IBE and a PKE scheme in sequential or parallel fashion.

- Partial private key = private key in IBE scheme.
- Secret value = private key in PKE scheme; public key = public key in PKE scheme.
- Private key for CLE scheme = concatenation of IBE and PKE private keys.
- Then:
  1. Encrypt first with PKE scheme, then with IBE scheme; or
  2. Encrypt first with IBE scheme, then with PKE scheme; or
  3. Encrypt with PKE and IBE schemes in parallel.

Generic Constructions for CLE

- Libert-Quisquater (2006): the first generic construction is insecure if partial private key extract queries are allowed, even if component IBE and PKE schemes are IND-CCA secure.
  - Simple attack based on partial private key extraction.
- Similar results for IBE followed by PKE, for IBE followed by IBE, and for parallel composition (Libert-Quisquater, Galindo *et al.*, Dent).
- Similar attacks already existed for normal PKE schemes obtained by multiple encryption (Dodis-Katz, Zhang *et al.*).

Generic Constructions for CLE

Libert-Quisquater (2006) gave a generic conversion from IND-CPA security to IND-CCA security for CLE:

- Let $\mathsf{Encrypt}(M, R, \mathsf{ID})$ and $\mathsf{Decrypt}(C, sk_{\mathsf{ID}})$ be algorithms of an IND-CPA secure CLE scheme.
- Here R denotes randomness used during encryption.
- Define new algorithms $\mathsf{Encrypt}'$, $\mathsf{Decrypt}'$ via:
  - $\mathsf{Encrypt}'(m, \sigma, \mathsf{ID})$=$\mathsf{Encrypt}(M, R, \mathsf{ID})$ where

$$M = m||\sigma, \quad R = H(m||\sigma||pk_{\mathsf{ID}}||\mathsf{ID}).$$

  - $\mathsf{Decrypt}'(C, sk_{\mathsf{ID}}) = m$ if

$$C = \mathsf{Encrypt}(m||\sigma, H(m||\sigma||pk_{\mathsf{ID}}||\mathsf{ID})).$$

Generic Constructions for CLE

- Libert-Quisquater construction works in the Random Oracle Model and yields IND-CCA security in the full model of Al-Riyami–Paterson.
- Generalises Fujisaki-Okamoto technique from PKE to CLE setting.
- Libert-Quisquater showed that generic sequential/parallel constructions of Al-Riyami/Yum-Lee are IND-CPA secure if the PKE and IBE components are.
- This allows easy construction of IND-CCA secure CLE schemes from IND-CPA secure components.

A Second Concrete CLE Scheme

- Al-Riyami–Paterson (2005) gave an efficient variant of their 2003 CLE scheme, with:

$$C = \langle rP, \sigma \oplus H_2(e(P_0, H_1(\mathsf{ID}))^r) \oplus H_5(rY_{\mathsf{ID}}), M \oplus H_4(\sigma) \rangle$$

  where $r = H_3(\sigma, M)$.

- Now no need to check format of public key, security based on hardness of BDHP.

- But Libert-Quisquater (2006) and Zhang-Feng (2005) showed that this scheme is vulnerable to a Strong Type I attacker.

A Second Concrete CLE Scheme

- Because underlying CLE scheme is IND-CPA secure, the generic conversion of Libert-Quisquater (2006) can be used to repair the Al-Riyami–Paterson (2005) scheme, simply by setting:

$$r = H_3(\sigma||M||pk_{\mathsf{ID}}||\mathsf{ID})$$

  when creating randomness.

- IND-CCA security based on hardness of BDHP, and more efficient than original CLE scheme.

- A similar scheme was proposed independently by Cheng-Comley (2005).

A Third Concrete CLE Scheme

- Libert-Quisquater (2006) also gave an efficient IND-CCA secure scheme based on the Sakai-Kasahara ID-based keying technique:

$$d_{\mathsf{ID}} = \frac{1}{s + H(\mathsf{ID})} \cdot P.$$

- Security based on the hardness of q-Bilinear Diffie-Hellman Inversion (q-BDHI) problem:
  - Given $\langle P, xP, x^2P, \ldots, x^qP \rangle$, compute $e(P,P)^{1/x}$.
- Ciphertext contains only two elements of $\mathbb{G}$; encryption is pairing-free.
- Map-to-point hashing is avoided.
- Similar scheme also given by Shi-Li (2005).

Certificateless KEMs

Bentahar *et al.* (2005):

- Introduced notion of Certificateless KEMs as a lightweight way of encapsulating a (symmetric) key.
- Secure CL-KEM + secure DEM $\rightarrow$ secure CLE.
- CLE scheme so obtained only has Weak Type Ia security.
- Generic construction for secure CL-KEM from OW-CPA^{++} secure, verifiable PKE (e.g. textbook RSA) and OW-ID-CCA secure IBE, using ROM.
- Hence reasonable CLE security from weak components in ROM.

CLE in the Standard Model

- Libert-Quisquater (2006) generic construction requires use of random oracles in security analysis.
- Construction of CLE secure in the standard model against Strong Type I attackers an interesting theoretical question.
- Some doubt as to whether achievable at all!
- Dent, Libert and Paterson (2006):
  - Generic construction for Strong Type I and Strong Type II IND-CCA secure CLE from any IND-CPA secure CLE and PKE using NIZK proofs; and
  - Specific, efficient construction for IND-CCA secure CLE from a variant of Waters' IBE using Boyen-Mei-Waters-style ideas.

Coming Up in Part II ...

- CLE and Certificate-Based Encryption
- Malicious KGCs and Denial-of-Decryption Attacks
- Mediated certificateless encryption, certificateless signatures (CLS) and further certificateless primitives
- Conclusions and open problems