Round-Optimal Waters Blind Signatures

David Pointcheval

Joint work with Olivier Blazy, Georg Fuchsbauer and Damien Vergnaud

Ecole normale supérieure, CNRS & INRIA

Institute of Advanced Studies of Tsinghua University
Beijing – China – October 18th, 2010

Outline

Outline

Electronic Cash

Electronic Coins [Chaum, 1981]

Expected properties:

- coins are signed by the bank, for unforgeability
- coins must be distinct to detect/avoid double-spending
- the bank should not know to whom it gave a coin, for anonymity

Electronic Cash

The process is the following one:

- Withdrawal: the user gets a coin c from the bank
- Spending: the user spends a coin c in a shop
- Deposit: the shop gives back the money to the bank

Blind Signatures

We thus want:

- Anonymity: the bank cannot link a withdrawal to a deposit to know where a user spent a coin
  $\rightarrow$ blind signature
- No double-spending: a coin should not be used twice
  $\rightarrow$ fair blind signature

Perfectly Blind Signatures

A blind signature allows a user to get a message m signed by an authority into σ so that the authority (even powerful) cannot recognize later the pair (m, σ).

Blind Signatures

We thus want:

- Anonymity: the bank cannot link a withdrawal to a deposit to know where a user spent a coin
  $\rightarrow$ blind signature
- No double-spending: a coin should not be used twice
  $\rightarrow$ fair blind signature

Computationally/Fair Blind Signatures

Unlinkability between the signing process and the pair (m, σ) is either computational, or even revocable (fair blind signatures).

The latter property allows to know/trace the defrauder after double-spending detection.

Blind RSA

[Chaum, 1981]

The easiest way for blind signatures, is to blind the message:
To get an FDH RSA signature on m under RSA public key (n, e),

- The user computes a blind version of the hash value:
  $M = H(m)$ and $M' = M \cdot r^e \bmod n$
- The signer signs M' into $\sigma' = M'^d \bmod n$
- The user unblinds the signature: $\sigma = \sigma'/r \bmod n$

Indeed,

$$\sigma = \sigma'/r = M'^d/r = (M \cdot r^e)^d/r = M^d \cdot r/r = M^d \bmod n$$

$\rightarrow$ Proven under the One-More RSA Assumption

[Bellare, Namprempre, Pointcheval, Semanko, 2001]

$\rightarrow$ Perfectly blind signature

Blind Signatures and NIZK

[Fischlin, 2006]

Fischlin Approach

To get a signature on m,

- The user commits m into c
- The signer signs c into σ
- The user generates a NIZK proof of knowledge of c and σ, valid with respect to m and the signer public key

This can be instantiated within the Groth-Sahai methodology

This method is in the same vein as the Blind RSA:

- The user commits m into c: blinding of the message
- The signer signs c into σ: signature on the blinded message
- The user generates a NIZK proof of knowledge of c and σ
  $\rightarrow$ Could we do an unblinding?

Outline

Assumptions: Diffie-Hellman

Definition (The Computational Diffie-Hellman problem (*CDH*))

$\mathbb{G}$ a cyclic group of prime order p.
The *CDH* assumption in $\mathbb{G}$ states:
for any generator $g \stackrel{\$}{\leftarrow} \mathbb{G}$, and any scalars $a, b \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$,
given (g, g^a, g^b), it is hard to compute g^{ab}.

Definition (The Decisional Diffie-Hellman problem (*DDH*))

$\mathbb{G}$ a cyclic group of prime order p.
The *DDH* assumption in $\mathbb{G}$ states:
for any generator $g \stackrel{\$}{\leftarrow} \mathbb{G}$, and any scalars $a, b, c \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$,
given (g, g^a, g^b, g^c), it is hard to decide whether $c = ab$ or not.

In some pairing-friendly groups, the latter assumption is wrong.

Assumptions: Linear Problem

Definition (Decision Linear Assumption (*DLin*))

$\mathbb{G}$ a cyclic group of prime order p.
The *DLin* assumption states:
for any generator $g \stackrel{\$}{\leftarrow} \mathbb{G}$, and any scalars $a, b, x, y, c \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$,
given $(g, g^x, g^y, g^{xa}, g^{yb}, g^c)$,
it is hard to decide whether $c = a + b$ or not.

Equivalently, given a reference triple $(u = g^x, v = g^y, g)$ and a new triple $(U = u^a = g^{xa}, V = v^b = g^{yb}, T = g^c)$, decide whether $T = g^{a+b}$ or not (that is $c = a + b$).

General Tools: Signature

Definition (Signature Scheme)

$\mathcal{S} = (\mathit{Setup}, \mathit{SKeyGen}, \mathit{Sign}, \mathit{Verif})$:
- $\mathit{Setup}(1^k) \quad \rightarrow$ global parameters *param*;
- $\mathit{SKeyGen}(\mathit{param}) \quad \rightarrow$ pair of keys (sk, vk);
- $\mathit{Sign}(sk, m; s) \quad \rightarrow$ signature σ, using the random coins s;
- $\mathit{Verif}(vk, m, \sigma) \quad \rightarrow$ validity of σ

If one signs $F = \mathcal{F}(M)$, for any function $\mathcal{F}$, one extends the above definitions: $\mathit{Sign}(sk, (\mathcal{F}, F, \Pi_M); s)$ and $\mathit{Verif}(vk, (\mathcal{F}, F, \Pi_M), \sigma)$ where $\mathcal{F}$ details the function that is applied to the message M yielding F, and Π_M is a proof of knowledge of a preimage of F under $\mathcal{F}$.

Signature: Example

In a group $\mathbb{G}$ of order p, with a generator g,
and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$

Waters Signature [Waters, 2005]

For a message $M = (M_1, \ldots, M_k) \in \{0,1\}^k$,
we define $\mathcal{F}(M) = u_0 \prod_{i=1}^{k} u_i^{M_i}$, where $\vec{u} = (u_0, \ldots, u_k) \stackrel{\$}{\leftarrow} \mathbb{G}^{k+1}$.
For an additional generator $h \stackrel{\$}{\leftarrow} \mathbb{G}$.

- *SKeyGen*: $vk = X = g^x$, $sk = Y = h^x$, for $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$;
- *Sign*$(sk = Y, M; s)$, for $M \in \{0,1\}^k$ and $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$
  $\to \quad \sigma = (\sigma_1 = Y \cdot \mathcal{F}(M)^s, \sigma_2 = g^{-s})$;
- *Verif*$(vk = X, M, \sigma = (\sigma_1, \sigma_2))$ checks whether

$$e(g, \sigma_1) \cdot e(\mathcal{F}(M), \sigma_2) = e(X, h).$$

General Tools: Encryption

Definition (Encryption Scheme)

$\mathcal{E} = (\mathit{Setup}, \mathit{EKeyGen}, \mathit{Encrypt}, \mathit{Decrypt})$:

- $\mathit{Setup}(1^k) \quad \to$ global parameters *param*;
- $\mathit{EKeyGen}(\mathit{param}) \quad \to$ pair of keys (pk, dk);
- $\mathit{Encrypt}(pk, m; r) \quad \to$ ciphertext c, using the random coins r;
- $\mathit{Decrypt}(dk, c) \quad \to$ plaintext, or $\perp$ if the ciphertext is invalid.

Homomorphic Encryption

For some group laws: $\oplus$ on the plaintext, $\otimes$ on the ciphertext,
and $\odot$ on the randomness

$$\mathit{Encrypt}(pk, m_1; r_1) \otimes \mathit{Encrypt}(pk, m_2; r_2) = \mathit{Encrypt}(pk, m_1 \oplus m_2; r_1 \odot r_2)$$

$$\mathit{Decrypt}(sk, \mathit{Encrypt}(pk, m_1; r_1) \otimes \mathit{Encrypt}(pk, m_2; r_2)) = m_1 \oplus m_2$$

Encryption: Example

In a group $\mathbb{G}$ of order p, with a generator g:

Linear Encryption [Boneh, Boyen, Shacham, 2004]

- *EKeyGen*: $dk = (x_1, x_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$, $pk = (X_1 = g^{x_1}, X_2 = g^{x_2})$;
- *Encrypt*$(pk = (X_1, X_2), m; (r_1, r_2))$, for $m \in \mathbb{G}$ and $(r_1, r_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$
  $\to \quad c = (c_1 = X_1^{r_1}, c_2 = X_2^{r_2}, c_3 = g^{r_1 + r_2} \cdot m)$;
- *Decrypt*$(dk = (x_1, x_2), c = (c_1, c_2, c_3)) \quad \to \quad m = c_3 / c_1^{1/x_1} c_2^{1/x_2}$.

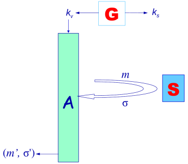
Homomorphism

$(\oplus_M = \times, \otimes_C = \times, \odot_R = +)$-homomorphism
With $m = g^M \quad \to \quad (\oplus_M = +, \otimes_C = \times, \odot_R = +)$-homomorphism

Security Notions: Signature

Signature: EF-CMA

Existential Unforgeability
under Chosen-Message
Attacks
An adversary should not be able to generate a new valid message-signature pair
even if it is allowed to ask signatures on any message of its choice



Impossibility to forge signatures

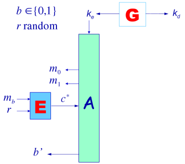
Waters signature reaches EF-CMA under the *CDH* assumption

Security Notions: Encryption

Encryption: IND-CCA

Indistinguishability under Chosen-Plaintext Attacks

An adversary that chooses two messages, and receives the encryption of one of them, should not be able to decide which one has been encrypted



Impossibility to learn any information about the plaintext

The Linear Encryption reaches IND-CPA under the *DLin* assumption

Groth-Sahai Commitments

[Groth, Sahai, 2008]

Under the *DLin* assumption, the commitment key is:

$$(\mathbf{u}_1 = (u_{1,1}, 1, g), \mathbf{u}_2 = (1, u_{2,2}, g), \mathbf{u}_3 = (u_{3,1}, u_{3,2}, u_{3,3})) \in (\mathbb{G}^3)^3$$

Initialization

$$\mathbf{u}_3 = \mathbf{u}_1^{\lambda} \odot \mathbf{u}_2^{\mu} = (u_{3,1} = u_{1,1}^{\lambda}, u_{3,2} = u_{2,2}^{\mu}, u_{3,3} = g^{\lambda+\mu})$$

with $\lambda, \mu \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, and random elements $u_{1,1}, u_{2,2} \stackrel{\$}{\leftarrow} \mathbb{G}$.

It means that $\mathbf{u}_3$ is a linear tuple w.r.t. $(u_{1,1}, u_{2,2}, g)$.

Groth-Sahai Commitments

Group Element Commitment

To commit a group element $X \in \mathbb{G}$,
one chooses random coins $s_1, s_2, s_3 \in \mathbb{Z}_p$ and sets

$$\begin{aligned}\mathcal{C}(X) &:= (1, 1, X) \odot \mathbf{u}_1^{s_1} \odot \mathbf{u}_2^{s_2} \odot \mathbf{u}_3^{s_3} \\ &= (u_{1,1}^{s_1} \cdot u_{3,1}^{s_3}, u_{2,2}^{s_2} \cdot u_{3,2}^{s_3}, X \cdot g^{s_1+s_2} \cdot u_{3,3}^{s_3}).\end{aligned}$$

Scalar Commitment

To commit a scalar $x \in \mathbb{Z}_p$,
one chooses random coins $\gamma_1, \gamma_2 \in \mathbb{Z}_p$ and sets

$$\begin{aligned}\mathcal{C}'(x) &:= (u_{3,1}^{x}, u_{3,2}^{x}, (u_{3,3}g)^{x}) \odot \mathbf{u}_1^{\gamma_1} \odot \mathbf{u}_3^{\gamma_2} \\ &= (u_{3,1}^{x+\gamma_2} \cdot u_{1,1}^{\gamma_1}, u_{3,2}^{x+\gamma_2}, u_{3,3}^{x+\gamma_2} \cdot g^{x+\gamma_1}).\end{aligned}$$

Groth-Sahai Proofs

- If $\mathbf{u}_3$ a linear tuple, these commitments are perfectly binding
- With the initialization parameters, the committed values can even be extracted $\rightarrow$ extractable commitments
- Using pairing product equations, one can make proofs on many relations between scalars and group elements:

$$\prod_j e(A_j, X_j)^{\alpha_j} \prod_i e(Y_i, B_i)^{\beta_i} \prod_{i,j} e(X_i, Y_j)^{\gamma_{i,j}} = t,$$

  where the A_j, B_i, and t are constant group elements,
  α_i, β_j, and $\gamma_{i,j}$ are constant scalars,
  and X_j and Y_i are either group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$,
  or of the form $g_1^{x_j}$ or $g_2^{y_i}$, respectively, to be committed.
- The proofs are perfectly sound

Groth-Sahai Proofs

- If $\mathbf{u}_3$ a linear tuple, these commitments are perfectly binding
- The proofs are perfectly sound

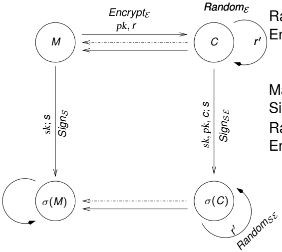
- If $\mathbf{u}_3$ is a random tuple, the commitments are perfectly hiding
- The proofs are perfectly witness hiding

- Under the *DLin* assumption, with a correct initialization, proofs are witness hiding

Can be used for any Pairing Product Equation
If one re-randomizes the commitments, the proof can be adapted

Outline

Signatures on Randomizable Ciphertexts



Randomizable Encryption

Malleable Signature on Randomizable Encryption

Linear Encryption

In a group $\mathbb{G}$ of order p, with a generator g, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$

Linear Encryption [Boneh, Boyen, Shacham, 2004]

- $EKeyGen$: $dk = (x_1, x_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$, $pk = (X_1 = g^{x_1}, X_2 = g^{x_2})$;
- $Encrypt(pk = (X_1, X_2), m; (r_1, r_2))$, for $m \in \mathbb{G}$ and $(r_1, r_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$
  $\to \quad c = (c_1 = X_1^{r_1}, c_2 = X_2^{r_2}, c_3 = g^{r_1+r_2} \cdot m)$;
- $Decrypt(dk = (x_1, x_2), c = (c_1, c_2, c_3)) \quad \to \quad m = c_3 / c_1^{1/x_1} c_2^{1/x_2}$.

Re-Randomization

- $Random_{\mathcal{E}}(pk = (X_1, X_2), c = (c_1, c_2, c_3); (r_1', r_2'))$, for $(r_1', r_2') \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$
  $\to \quad c' = (c_1' = c_1 \cdot X_1^{r_1'}, c_2' = c_2 \cdot X_2^{r_2'}, c_3' = c_3 \cdot g^{r_1'+r_2'})$.

Example

Waters Signature

In a group $\mathbb{G}$ of order p, with a generator g, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$

Waters Signature [Waters, 2005]

For a message $M = (M_1, \ldots, M_k) \in \{0,1\}^k$, we define $F = \mathcal{F}(M) = u_0 \prod_{i=1}^k u_i^{M_i}$, where $\vec{u} = (u_0, \ldots, u_k) \stackrel{\$}{\leftarrow} \mathbb{G}^{k+1}$. For an additional generator $h \stackrel{\$}{\leftarrow} \mathbb{G}$.

- *SKeyGen*: $vk = X = g^x$, $sk = Y = h^x$, for $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$;
- *Sign*$(sk = Y, F; s)$, for $M \in \{0,1\}^k$, $F = \mathcal{F}(M)$, and $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$
  $\to \quad \sigma = (\sigma_1 = Y \cdot F^s, \sigma_2 = g^{-s})$;
- *Verif*$(vk = X, M, \sigma = (\sigma_1, \sigma_2))$ checks whether
  $$e(g, \sigma_1) \cdot e(F, \sigma_2) = e(X, h).$$

Example

Waters Signature on a Linear Ciphertext: Idea

We define $F = \mathcal{F}(M) = u_0 \prod_{i=1}^k u_i^{M_i}$, and encrypt it

$$c = (c_1 = X_1^{r_1}, c_2 = X_2^{r_2}, c_3 = g^{r_1+r_2} \cdot F)$$

- *KeyGen*: $vk = X = g^x$, $sk = Y = h^x$, for $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$
  $dk = (x_1, x_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$, $pk = (X_1 = g^{x_1}, X_2 = g^{x_2})$
- *Sign*$((X_1, X_2), Y, c; s)$, for $c = (c_1, c_2, c_3)$
  $\to \quad \sigma = (\sigma_1 = Y \cdot c_3^s, \sigma_2 = (c_1^s, c_2^s), \sigma_3 = (g^s, X_1^s, X_2^s))$
- *Verif*$((X_1, X_2), X, c, \sigma)$ checks $\quad e(g, \sigma_1) = e(X, h) \cdot e(\sigma_{3,0}, c_3)$
  $$e(\sigma_{2,0}, g) = e(c_1, \sigma_{3,0}) \qquad e(\sigma_{2,1}, g) = e(c_2, \sigma_{3,0})$$
  $$e(\sigma_{3,1}, g) = e(X_1, \sigma_{3,0}) \qquad e(\sigma_{3,2}, g) = e(X_2, \sigma_{3,0})$$

σ_3 is needed for ciphertext re-randomization

Example

Re-Randomization of Ciphertext

$$\begin{array}{llll} c = (c_1 = X_1^{r_1}, & c_2 = X_2^{r_2}, & c_3 = g^{r_1+r_2} \cdot F &) \\ \sigma = (\sigma_1 = Y \cdot c_3^s, & \sigma_2 = (c_1^s, c_2^s), & \sigma_3 = (g^s, X_1^s, X_2^s) &) \end{array}$$

after re-randomization by $\quad (r_1', r_2')$

$$\begin{array}{llll} c' = (c_1' = c_1 \cdot X_1^{r_1'}, & c_2' = c_2 \cdot X_2^{r_2'}, & c_3' = c_3 \cdot g^{r_1'+r_2'} &) \\ \sigma' = (\sigma_1' = \sigma_1 \cdot \sigma_{3,0}^{r_1'+r_2'}, & \sigma_2' = (\sigma_{2,0} \cdot \sigma_{3,1}^{r_1'}, \sigma_{2,1} \cdot \sigma_{3,2}^{r_2'}), & \sigma_3' = \sigma_3 &) \end{array}$$

Anybody can publicly re-randomize c into c'
with additional random coins (r_1', r_2'),
and adapt the signature σ of c into σ' of c'

Security Notions

Unforgeability under Chosen-Ciphertext Attacks

Chosen-Ciphertext Attacks

The adversary is allowed to ask any valid ciphertext of his choice to the signing oracle

Because of the re-randomizability of the ciphertext-signature, we cannot expect resistance to existential forgeries, but we should allow a restricted malleability only:

Forgery

A valid ciphertext-signature pair, so that the plaintext is different from all the plaintexts in the ciphertexts sent to the signing oracle

Unforgeability

From a valid ciphertext-signature pair:

$$\begin{aligned} c &= (c_1 = X_1^{r_1}, c_2 = X_2^{r_2}, c_3 = g^{r_1+r_2} \cdot F) \\ \sigma &= (\sigma_1 = Y \cdot c_3^s, \sigma_2 = (c_1^s, c_2^s), \sigma_3 = (g^s, X_1^s, X_2^s)) \end{aligned}$$

and the decryption key (x_1, x_2), one extracts

$$\begin{aligned} F &= c_3/(c_1^{1/x_1} c_2^{1/x_2}) \\ \Sigma &= (\Sigma_1 = \sigma_1/(\sigma_{2,0}^{1/x_1}\sigma_{2,1}^{1/x_2}), \quad \Sigma_2 = \sigma_{3,0}) \\ &= (\quad = Y \cdot F^s, \quad = g^s) \end{aligned}$$

Security of Waters signature is for a pair (M, Σ)

$\rightarrow$ needs of a proof of knowledge Π_M of M in $F = \mathcal{F}(M)$

bit-by-bit commitment of M and Groth-Sahai proof

Chosen-Message Attacks

From a valid ciphertext $c = (c_1 = X_1^{r_1}, c_2 = X_2^{r_2}, c_3 = g^{r_1+r_2} \cdot F)$,
and the additional proof of knowledge of M,
one extracts M and asks for a Waters signature:

$$\Sigma = (\Sigma_1 = Y \cdot F^s, \Sigma_2 = g^s)$$

In this signature, the random coins s are unknown,
we thus need to know the coins in c

$\rightarrow$ needs of a proof of knowledge Π_r of r_1, r_2 in c

bit-by-bit commitment of r_1, r_2 and Groth-Sahai proof

From the random coins r_1, r_2 (and the decryption key):

$$\begin{aligned} \sigma = (\sigma_1 &= \Sigma_1 \cdot \Sigma_2^{r_1+r_2}, & \sigma_2 &= (\Sigma_2^{x_1 r_1}, \Sigma_2^{x_2 r_2}), & \sigma_3 &= (\Sigma_2, \Sigma_2^{r_1}, \Sigma_2^{r_2})\ ) \\ &= Y \cdot c_3^s, & &= (c_1^s, c_2^s), & &= (g^s, X_1^s, X_2^s) \end{aligned}$$

Security

Chosen-Ciphertext Attacks

A valid ciphertext $C = (c_1, c_2, c_3, \Pi_M, \Pi_r)$ is a

- ciphertext $c = (c_1, c_2, c_3)$
- a proof of knowledge Π_M of the plaintext M in $F = \mathcal{F}(M)$
- a proof of knowledge Π_r of the random coins r_1, r_2

From such a ciphertext and the decryption key (x_1, x_2),
and a Waters signing oracle, one can generate a signature on C

Forgery

From a valid ciphertext-signature pair (C, σ), where C encrypts M,
one can generate a Waters signature on M

Properties

Security Level

Since the Waters signature is EF-CMA under the *CDH* assumption,
our signature on randomizable ciphertext is Unforgeable
against Chosen-Ciphertext Attacks under the *CDH* assumption

Proofs

Since we use the Groth-Sahai methodology for the proofs Π_M and Π_r

- in case of re-randomization of c, one can adapt Π_M and Π_r
- because of the need of M, but also r_1 and r_2 in the simulation,
  we need bit-by-bit commitments: $\rightarrow$ C is large!

Efficiency

We can improve efficiency: shorter signatures

Revisited Waters Signature

In a group $\mathbb{G}$ of order p, with a generator g,
and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$

Improved Signature

- *SKeyGen*: $vk = X = g^x$, $sk = Y = h^x$, for $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$;
- *Sign*$(sk = Y, (M, R_1, R_2, T); s)$, if $e(R_1 R_2, X) = e(g, T)$,
  which guarantees existence of $r_1, r_2 \in \mathbb{Z}_p$ such that
  $R_1 = g^{r_1}$, $R_2 = g^{r_2}$ and $T = X^{r_1+r_2}$
  $\to \quad \sigma = (\sigma_1 = Y \cdot (\mathcal{F}(M) R_1 R_2)^s, \sigma_2 = (g^{-s}, R_1^{-s}, R_2^{-s}))$;
- *Verif*$(vk = X, (M, R_1, R_2, T), \sigma = (\sigma_1, \sigma_2))$ checks whether

$$e(g, \sigma_1) \cdot e(\mathcal{F}(M) R_1 R_2, \sigma_{2,0}) = e(X, h) \quad e(R_1 R_2, X) = e(g, T)$$
$$e(g, \sigma_{2,1}) = e(\sigma_{2,0}, R_1) \quad e(g, \sigma_{2,2}) = e(\sigma_{2,0}, R_2)$$

Properties

Revisited Waters Signature: EF-CMA

Our Waters Signature Variant is EF-CMA under the *CDH* assumption

Signature on a Linear Ciphertext

Ciphertext signatures queries

- still need a proof of knowledge of M (bit-by-bit)
- but only proof of knowledge of $R_1 = g^{r_1}$, $R_2 = g^{r_2}$ and $T = X^{r_1+r_2}$

$\to \quad M$, and $R_1 = g^{r_1}$, $R_2 = g^{r_2}$, $T = X^{r_1+r_2}$ are enough
to simulate signatures on ciphertexts from a signing oracle

Efficiency

For an ℓ-bit message, a pair (C, σ) consists of $9\ell + 33$ group elements

Outline

Extractability

As already noted, from a valid ciphertext-signature pair:

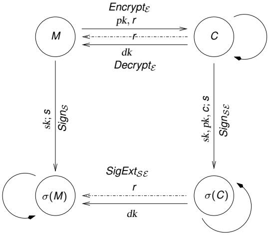
$$c = (c_1 = X_1^{r_1}, c_2 = X_2^{r_2}, c_3 = g^{r_1+r_2} \cdot F)$$
$$\sigma = (\sigma_1 = Y \cdot c_3^s, \sigma_2 = (c_1^s, c_2^s), \sigma_3 = (g^s, X_1^s, X_2^s))$$

and the decryption key (x_1, x_2), one extracts

$$\begin{array}{llll} F = & c_3/(c_1^{1/x_1} c_2^{1/x_2}) & & \\ \Sigma = (& \Sigma_1 = \sigma_1/(\sigma_{2,0}^{1/x_1} \sigma_{2,1}^{1/x_2}), & \Sigma_2 = \sigma_{3,0}) \\ \quad = (& \quad = Y \cdot F^s & \quad = g^s) \end{array}$$

A plain Waters Signature
One can do the same from the random coins (r_1, r_2)

Extractable Signatures



Blind Signatures

A New Approach

To get a signature on M,

- The user commits/encrypts M into C, under random coins r
- The signer signs C into $\sigma(C)$, under random coins s
- The user extracts a signature $\sigma(M)$, granted the random coins r

Weakness

The signer can recognize his signature: the random coins s in $\sigma(M)$

$\rightarrow$ Randomizable Signature

Randomizable Signatures

Waters Signature

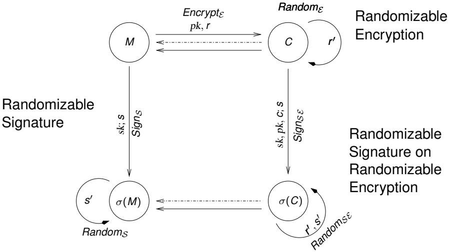
- *SKeyGen*: $vk = X = g^x$, $sk = Y = h^x$, for $x \stackrel{\$}{\leftarrow} \mathbb{Z}_p$;
- *Sign*$(sk = Y, M; s)$, for $M \in \{0,1\}^k$ and $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$
  $\rightarrow \quad \sigma = \left(\sigma_1 = Y \cdot \mathcal{F}(M)^s, \sigma_2 = g^{-s}\right)$;
- *Verif*$(vk = X, M, \sigma = (\sigma_1, \sigma_2))$ checks whether

$$e(g, \sigma_1) \cdot e(\mathcal{F}(M), \sigma_2) = e(X, h).$$

Re-Randomization

$\mathit{Random}_{\mathcal{S}}(vk = X, M, \sigma; s') : \sigma' = \left(\sigma'_1 = \sigma_1 \cdot \mathcal{F}(M)^{s'}, \sigma'_2 = \sigma_2 \cdot g^{-s'}\right)$
this is exactly *Sign*$(sk = Y, M; s + s')$

Randomizable Signatures



Blind Signatures

Our Approach

To get a signature on M,

- The user commits/encrypts M into C, under random coins r
- The signer signs C into $\sigma(C)$, under random coins s
- The user extracts a signature $\sigma(M)$, granted the random coins r
- The user re-randomizes the signature $\sigma(M)$, under additional random coins s'

Security

- encryption hides M
- re-randomization hides $\sigma(M)$

Blind Signatures

Such a primitive can be used for a Waters Blind Signature:

- Unforgeability: one-more forgery would imply a forgery against the signature scheme (*CDH* assumption)
- Blindness: a distinguisher would break indistinguishability of the encryption scheme (*DLin* assumption)

Efficiency

We obtain a plain Waters Signature

$\rightarrow$ Blind Signature: with a real Waters Signature

Fair Blind Signatures

One can even exploit double trapdoor: random coins r and decryption key dk

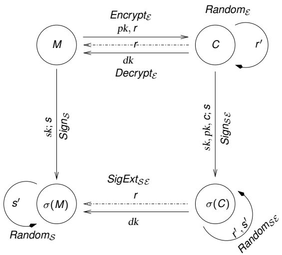
Fair Blind Signatures

To get a signature on M,

- The user encrypts M into C, under random coins r, and the authority encryption key
- The signer signs C into $\sigma(C)$, under random coins s
- The user extracts a signature $\sigma(M)$, granted the random coins r
- The user re-randomizes the signature $\sigma(M)$, under additional random coins s'

Double-spending: the authority can decrypt the ciphertexts C to find the defrauder.

Our New Primitive



Conclusion

Extractable Randomizable Signature on Randomizable Ciphertexts

Various Applications

- non-interactive receipt-free electronic voting scheme
- (fair) blind signature

Security relies on the *CDH* and the *DLin* assumptions
For an ℓ-bit message, ciphertext-signature:
$9\ell + 33$ group elements

A more efficient variant with asymmetric pairing
on the *CDH** and the *SXDH* assumptions
Ciphertext-signature: $6\ell + 15$ group elements in $\mathbb{G}_1$
and $6\ell + 7$ group elements in $\mathbb{G}_2$