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

1 Introduction
2 Cryptographic Tools
3 Signatures on Randomizable Ciphertexts
4 Blind Signatures

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 → blind signature
- No double-spending: a coin should not be used twice → fair blind signature

Computationally/Fair Blind Signatures

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

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

Blind RSA

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 \(M' = M \cdot r^e \mod n\)
- The signer signs \(M'\) into \(\sigma' = M'^d \mod n\)
- The user unblinds the signature: \(\sigma = \sigma' / r \mod n\)

Indeed,

\[
\sigma = \sigma' / r = M'^d / r = (M \cdot r^e)^d / r = M^d \cdot r / r = M^d \mod n
\]

→ Proven under the One-More RSA Assumption [Bellare, Namprempre, Pointcheval, Semanko, 2001]

→ Perfectly blind signature

Blind Signatures and NIZK

Fischlin Approach

To get a signature on \(m\),

- The user commits \(m\) into \(c\)
- The signer signs \(c\) into \(\sigma\)
- The user generates a NIZK proof of knowledge of \(c\) and \(\sigma\), 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 \(\sigma\): signature on the blinded message
- The user generates a NIZK proof of knowledge of \(c\) and \(\sigma\) → Could we do an unblinding?
**Assumptions: Diffie-Hellman**

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

\[ \mathcal{G} \text{ a cyclic group of prime order } p. \]

The \( \text{CDH} \) assumption in \( \mathcal{G} \) states:

for any generator \( g \overset{s}{\leftarrow} \mathcal{G} \), and any scalars \( a, b \overset{s}{\leftarrow} \mathbb{Z}_p^* \),

given \( (g, g^a, g^b) \), it is hard to compute \( g^{ab} \).

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

\[ \mathcal{G} \text{ a cyclic group of prime order } p. \]

The \( \text{DDH} \) assumption in \( \mathcal{G} \) states:

for any generator \( g \overset{s}{\leftarrow} \mathcal{G} \), and any scalars \( a, b, c \overset{s}{\leftarrow} \mathbb{Z}_p^* \),

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))**

\[ \mathcal{G} \text{ a cyclic group of prime order } p. \]

The \( \text{DLin} \) assumption states:

for any generator \( g \overset{s}{\leftarrow} \mathcal{G} \), and any scalars \( a, b, x, y, c \overset{s}{\leftarrow} \mathbb{Z}_p^* \),

given \( (g, g^x, g^y, g^xb, g^yb, g^c) \),

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 \( c = a + b \)).

**Definition (Signature Scheme)**

\[ S = (\text{Setup}, \text{SKeyGen}, \text{Sign}, \text{Verif}) : \]

\[ \text{Setup}(1^k) \rightarrow \text{global parameters param}; \]

\[ \text{SKeyGen(param)} \rightarrow \text{pair of keys (sk, vk)}; \]

\[ \text{Sign}(sk, m; s) \rightarrow \text{signature } \sigma, \text{using the random coins } s; \]

\[ \text{Verif}(vk, m, \sigma) \rightarrow \text{validity of } \sigma \]

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

In a group $G$ of order $p$, with a generator $g$,

and a bilinear map $e: G \times G \rightarrow G_T$

**Waters Signature**

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 $\tau = (u_0, \ldots, u_k) \leftarrow G^{k+1}$.

For an additional generator $h \leftarrow G$.

- $\text{Sign}(sk = Y, M; s)$, for $M \in \{0, 1\}^k$ and $s \leftarrow \mathbb{Z}_p$,

  $\sigma = (\sigma_1 = Y \cdot \mathcal{F}(M)^s, \sigma_2 = g^{-s})$;

- $\text{Verif}(vk = X, M, \sigma)$ checks whether

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

**Definition (Encryption Scheme)**

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

- $\text{Setup}(1^k) \rightarrow$ global parameters $\text{param}$;
- $\text{EKeyGen}(\text{param}) \rightarrow$ pair of keys $(pk, dk)$;
- $\text{Encrypt}(pk, m; r) \rightarrow$ ciphertext $c$, using the random coins $r$;
- $\text{Decrypt}(dk, c) \rightarrow$ 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

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

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

**Encryption: Example**

In a group $G$ of order $p$, with a generator $g$:

**Linear Encryption**

- $\text{EKeyGen}: dk = (x_1, x_2) \leftarrow \mathbb{Z}_p^2$, $pk = (X_1 = g^{x_1}, X_2 = g^{x_2})$;
- $\text{Encrypt}(pk = (X_1, X_2), m; (r_1, r_2))$, for $m \in G$ and $(r_1, r_2) \leftarrow \mathbb{Z}_p^2$,

  $c = (c_1 = X_1^{r_1}, c_2 = X_2^{r_2}, c_3 = g^{r_1 + r_2} \cdot m)$;
- $\text{Decrypt}(dk = (x_1, x_2), c = (c_1, c_2, c_3)) \rightarrow m = c_3 / c_1^{1/x_1} c_2^{1/x_2}$.

**Homomorphism**

$(\oplus_M = \times, \otimes_C = \times, \odot_R = +)$-homomorphism

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

David Pointcheval – 15/45

David Pointcheval – 14/45
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

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

$$\mathbf{u} = (u_1, 1, g), u_2 = (1, u_2, g), u_3 = (u_3, u_3, u_3) \in (\mathbb{G}^3)^3$$

Initialization

$$u_3 = u_1^\lambda \otimes u_2^\mu = (u_3, 1, u_3, 2, u_3, 3) = g^{\lambda+\mu}$$

with $\lambda, \mu \in \mathbb{Z}_p$, and random elements $u_1, 2, u_2, 2 \in \mathbb{G}$.

It means that $u_3$ is a linear tuple w.r.t. $(u_1, u_2, g)$.

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:

$$C(X) := (1, 1, X) \odot u_1^{s_1} \odot u_2^{s_2} \odot u_3^{s_3} = (u_1^{s_1} \cdot u_2^{s_2} \cdot u_3^{s_3}, X \cdot g^{s_1 + s_2} \cdot u_3^{s_3}).$$

**Scalar Commitment**

To commit a scalar $x \in \mathbb{Z}_p$, one chooses random coins $Y_1, Y_2 \in \mathbb{Z}_p$ and sets:

$$C'(x) := (u_3^x, u_2^x, (u_3, 3 g)^x) \odot u_1 \gamma_1 \odot u_3 \gamma_2 = (u_3^{x+\gamma_2} \cdot u_1^{\gamma_1} \cdot u_3^{x+\gamma_2} \cdot u_3^{x+\gamma_1} \cdot g^{x+\gamma_1}).$$

Groth-Sahai Proofs

- If $u_3$ 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:

$$\prod_{i} e(A_i, X_j)_{\alpha_i} \prod_{i} e(Y_i, B_i)_{\beta_i} \prod_{i,j} e(X_i, Y_j)^{\gamma_{ij}} = t,$$

where the $A_i, B_i, t$ are constant group elements, $\alpha_i, \beta_i, \gamma_{ij}$ 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 $g_1^{\gamma_i}$ or $g_2^{\gamma_i}$, respectively, to be committed.
- The proofs are perfectly sound.
**Groth-Sahai Methodology**

1. **Groth-Sahai Proofs**
   - If $u_3$ a linear tuple, these commitments are perfectly binding
   - The proofs are perfectly sound
   - If $u_3$ 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**

1. **Introduction**
2. **Cryptographic Tools**
3. **Signatures on Randomizable Ciphertexts**
4. **Blind Signatures**

---

**Signatures on Randomizable Ciphertexts**

**New Primitive**

**Example**

**Security Notions**

**Improvement**

**Linear Encryption**

In a group $G$ of order $p$, with a generator $g$, and a bilinear map $e : G \times G \rightarrow G_T$

**Linear Encryption**

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

**Re-Randomization**

- $Randomize(pk = (X_1, X_2), c = (c_1, c_2, c_3); (r'_1, r'_2))$, for $(r'_1, r'_2) \leftarrow \mathbb{Z}_p^2$
  \[ c' = (c'_1 = c_1 \cdot X_1^{r'_1}, c'_2 = c_2 \cdot X_2^{r'_2}, c'_3 = c_3 \cdot g^{r'_1 + r'_2}). \]
**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**

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

For an additional generator $h \in \mathbb{G}$.

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

**Waters Signature on a Linear Ciphertext: Idea**

We define $F = F(M) = u_0 \prod_{i=1}^{k} u_i^{M_i}$, 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 $x \leftarrow \mathbb{Z}_p$
  
- **Sign**($X_1, X_2, Y, c; s$), for $c = (c_1, c_2, c_3)$
  
- **Verif**($X_1, X_2, X, c, \sigma$) checks
  
\[
e(g, \sigma_1) = e(X, h) \cdot e(\sigma_{3,0}, c_3)
\]

\[
e(\sigma_{2,0}, g) = e(c_1, \sigma_{3,0})
\]

\[
e(\sigma_{2,1}, g) = e(c_2, \sigma_{3,0})
\]

\[
e(\sigma_{3,1}, g) = e(X_1, \sigma_{3,0})
\]

\[
e(\sigma_{3,2}, g) = e(X_2, \sigma_{3,0})
\]

\[
\sigma_3 \text{ is needed for ciphertext re-randomization}
\]

**Re-Randomization of Ciphertext**

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

\[
\sigma = (\sigma_1 = Y \cdot c_3^s, \sigma_2 = (c_1^s, c_2^s), \sigma_3 = (g^s, X_1^s, X_2^s))
\]

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

\[
c' = (c_1' = c_1 \cdot X_1^{r_1'}, c_2' = c_2 \cdot X_2^{r_2'}, c_3' = c_3 \cdot g^{r_1' + r_2'})
\]

\[
\sigma' = (\sigma_1' = \sigma_1 \cdot \sigma_{3,0}^{r_1' + r_2'}, \sigma_2' = (\sigma_{2,0} \cdot \sigma_{3,1}^{r_1' + r_2'}, \sigma_{2,1} \cdot \sigma_{3,2}^{r_1' + r_2'}), \sigma_3' = \sigma_3)
\]

Anybody can publicly re-randomize $c$ into $c'$ with additional random coins $(r_1', r_2')$.

and adapt the signature $\sigma$ of $c$ into $\sigma'$ of $c'$
Unforgeability

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
\[ F = c_3 / (c_1^{1/x_1} \cdot c_2^{1/x_2}) \]
\[ \Sigma = (\Sigma_1 = \sigma_1 / (\sigma_2^{1/x_1} \cdot \sigma_2^{1/x_2}), \quad \Sigma_2 = \sigma_3^0) \]
\[ = (Y \cdot F^s, g^s) \]

Security of Waters signature is for a pair \((M, \Sigma)\)
→ needs of a proof of knowledge \(\Pi_M\) 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 \(c\)
→ needs of a proof of knowledge \(\Pi_r\) of \(r_1, r_2\) in \(c\)
based commitment of \(r_1, r_2\) and Groth-Sahai proof

From the random coins \(r_1, r_2\) (and the decryption key):
\[ \sigma = (\sigma_1 = \Sigma_1 \cdot \Sigma_2^{r_1+r_2}, \quad \sigma_2 = (\Sigma_2^{x_1 r_1}, \Sigma_2^{x_2 r_2}), \quad \sigma_3 = (\Sigma_2, \Sigma_2^{r_1}, \Sigma_2^{r_2}) ) \]
\[ = Y \cdot c_3^s, \quad = (c_1^s, c_2^s), \quad = (g^s, X_1^s, X_2^s) \]

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 \(\Pi_M\) of the plaintext \(M\) in \(F = \mathcal{F}(M)\)
- a proof of knowledge \(\Pi_r\) 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 \((C, \sigma)\), where \(C\) encrypts \(M\),
one can generate a Waters signature on \(M\)
**Revisited Waters Signature**

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

**Improved Signature**

- $\text{SKeyGen: } vk = X = g^x, \ sk = Y = h^x, \text{ for } x \leftarrow \mathbb{Z}_p$;
- $\text{Sign}(sk = Y, (M, R_1, R_2, T); s), \text{ if } e(R_1 R_2, X) = e(g, T)$, 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 $T = X^{r_1 + r_2}$
  \[ \sigma = (\sigma_1 = Y \cdot (\mathcal{F}(M) R_1 R_2)^s, \sigma_2 = (g^{-s}, R_1^{-s}, R_2^{-s})) \]
- $\text{Verif}(vk = X, (M, R_1, R_2, T), \sigma = (\sigma_1, \sigma_2))$ checks whether $e(g, \sigma_1) \cdot e(\mathcal{F}(M) R_1 R_2, \sigma_{2,0}) = e(X, h) \quad e(R_1 R_2, X) = e(g, T) \quad e(g, \sigma_{2,1}) = e(\sigma_{2,0}, R_1) \quad e(g, \sigma_{2,2}) = e(\sigma_{2,0}, R_2)$

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

**Efficiency**

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

**Outline**

1. Introduction
2. Cryptographic Tools
3. Signatures on Randomizable Ciphertexts
4. Blind Signatures
   - Extractable Signatures
   - Randomizable Signatures
   - Randomizable Commutative Signature/Encryption

**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

\[ F = \frac{c_3}{(c_1^{1/x_1} c_2^{1/x_2})} \]
\[ \Sigma = (\Sigma_1 = \sigma_1 / (c_2^{1/x_1} \sigma_{2,1}^{1/x_2}), \Sigma_2 = \sigma_{3,0}) \]
\[ = (Y \cdot F^s = g^s) \]

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

**Randomizable Signatures**

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

- **Re-Randomization**
  - $Random_s(vk = X, M, \sigma; s') : \sigma' = (\sigma_1' = \sigma_1 \cdot \mathcal{F}(M)s^0, \sigma_2' = \sigma_2 \cdot g^{-s^0})$
  - this is exactly $Sign(sk = Y, M; s + s')$

**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
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 $\sigma(M)$

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

→ 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.
Extractable Randomizable Signature on Randomizable Ciphertexts

Various Applications

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

Security relies on the \textit{CDH} and the \textit{DLin} assumptions
For an \`\-bit message, ciphertext-signature:
\[ 9^* + 33 \text{ group elements} \]

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