# Structure-Preserving Signatures and Commitments to Group Elements

Masayuki Abe [1] Georg Fuchsbauer [2] Jens Groth [3]

Kristiyan Haralambiev [4] Miyako Ohkubo [5]

CRYPTO, 16.08.2010

[1] Information Sharing Platform Laboratories, NTT Corporation, Japan
[2] École Normale Supérieure, CNRS - INRIA, France
[3] University College London, UK
[4] Computer Science Department, New York University, USA
[5] National Institute of Information and Communications Technology, Japan

# Our Contributions

## New commitment and signature schemes in bilinear groups

- Homomorphic trapdoor commitments to group elements
- Signatures on group elements, consisting of group elements (*structure-preserving*)
- Structure-preserving signatures signing their own public keys (*automorphic*)
- Simulatable signatures

# Our Contributions

## New commitment and signature schemes in bilinear groups

- Homomorphic trapdoor commitments to group elements
- Signatures on group elements, consisting of group elements (*structure-preserving*)
- Structure-preserving signatures signing their own public keys (*automorphic*)
- Simulatable signatures

## Applications

- Constant-size trapdoor commitments with sublinear keys
- First efficient round-optimal blind signatures (UC secure)
- First efficient group signatures with concurrent join w/o ROM
- First efficient anonymous proxy signatures

# Outline of the talk

1. Commitments
2. Automorphic Signatures
3. Signatures on Vectors of Group Elements
4. Applications of Our Signatures

# Commitments

- A commitment scheme consists of setup and algorithm Com
- Com takes a *message* and *randomness* and outputs a commitment
- Message and randomness are called opening.

# Commitments

- A commitment scheme consists of setup and algorithm Com
- Com takes a *message* and *randomness* and outputs a commitment
- Message and randomness are called opening. Our scheme is

hiding: a commitment reveals nothing about the message

binding: hard to find a commitment and two openings with *different* messages

# Commitments

- A commitment scheme consists of setup and algorithm Com
- Com takes a *message* and *randomness* and outputs a commitment
- Message and randomness are called opening. Our scheme is

| | |
|---|---|
| hiding: | a commitment reveals nothing about the message |
| binding: | hard to find a commitment and two openings with *different* messages |
| trapdoor: | given a trapdoor, a commitment can be opened to any message |
| homomorphic: | the product of two commitments is a commitment to the product of the messages |
| length-reducing: | a commitment is shorter than the message |

The messages are elements of a bilinear group

# Bilinear Groups and the DP Assumption

Bilinear group: $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$ with

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ cyclic groups of prime order $p$
- $e\colon \mathbb{G}_1 \quad \mathbb{G}_2 \; ! \; \mathbb{G}_T$ bilinear, ie
  $8X\ 2\ \mathbb{G}_1, 8Y\ 2\ \mathbb{G}_2, 8a, b\ 2\ \mathbb{Z}\colon\ e(X^a, Y^b) = e(X, Y)^{ab}$
- $\mathbb{G}_1 = hGi$, $\mathbb{G}_2 = hHi$, $\mathbb{G}_T = he(G, H)i$

# Bilinear Groups and the DP Assumption

**Bilinear group**: $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$ with

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ cyclic groups of prime order $p$
- $e\colon \mathbb{G}_1 \quad \mathbb{G}_2 \;!\; \mathbb{G}_T$ bilinear, ie
  $8X\ 2\ \mathbb{G}_1, 8Y\ 2\ \mathbb{G}_2, 8a, b\ 2\ \mathbb{Z}\colon\ e(X^a, Y^b) = e(X, Y)^{ab}$
- $\mathbb{G}_1 = hGi$, $\mathbb{G}_2 = hHi$, $\mathbb{G}_T = he(G, H)i$

## Double Pairing Assumption

Given random $G_R, G_T\ 2\ \mathbb{G}_1$ it is hard to find non-trivial $R, T\ 2\ \mathbb{G}_2$ satisfying $e(G_R, R)\, e(G_T, T) = 1$

# Bilinear Groups and the DP Assumption

Bilinear group: $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$ with

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ cyclic groups of prime order $p$
- $e\colon \mathbb{G}_1 \quad \mathbb{G}_2 \; ! \; \mathbb{G}_T$ bilinear, ie
  $8X\ 2\ \mathbb{G}_1, 8Y\ 2\ \mathbb{G}_2, 8a, b\ 2\ \mathbb{Z}\colon\ e(X^a, Y^b) = e(X, Y)^{ab}$
- $\mathbb{G}_1 = hGi$, $\mathbb{G}_2 = hHi$, $\mathbb{G}_T = he(G, H)i$

## Double Pairing Assumption

Given random $G_R, G_T\ 2\ \mathbb{G}_1$ it is hard to find non-trivial $R, T\ 2\ \mathbb{G}_2$ satisfying $e(G_R, R)\, e(G_T, T) = 1$

## Lemma

DDH in $\mathbb{G}_1$ implies the double pairing assumption

# Commitment Scheme for $n$ Messages

Setup: Generate $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$.

# Commitment Scheme for $n$ Messages

Setup: Generate $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$.

Key generation: Pick $G_R \quad \mathbb{G}_1^*$ and $x_1, \ldots, x_n \quad \mathbb{Z}_p$. Return

$$ck = (G_R, G_1 = G_R^{x_1}, \ldots, G_n = G_R^{x_n}) \quad \text{and} \quad tk = (x_1, \ldots, x_n).$$

# Commitment Scheme for $n$ Messages

Setup: Generate $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$.

Key generation: Pick $G_R$ $\mathbb{G}_1^*$ and $x_1, \ldots, x_n$ $\mathbb{Z}_p$. Return

$$ck = (G_R, G_1 = G_R^{x_1}, \ldots, G_n = G_R^{x_n}) \quad \text{and} \quad tk = (x_1, \ldots, x_n).$$

Commitment: On input $ck$, $(M_1, \ldots, M_n)$ 2 $\mathbb{G}_2^n$, $R$ 2 $\mathbb{G}_2$, return

$$\mathbf{c} = e(G_R, R) \prod_{i=1}^{n} e(G_i, M_i)$$

# Commitment Scheme for $n$ Messages

Setup: Generate $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$.

Key generation: Pick $G_R$ $\mathbb{G}_1^*$ and $x_1, \ldots, x_n$ $\mathbb{Z}_p$. Return

$$ck = (G_R, G_1 = G_R^{x_1}, \ldots, G_n = G_R^{x_n}) \quad \text{and} \quad tk = (x_1, \ldots, x_n).$$

Commitment: On input $ck$, $(M_1, \ldots, M_n)$ 2 $\mathbb{G}_2^n$, $R$ 2 $\mathbb{G}_2$, return

$$\mathbf{c} = e(G_R, R) \prod_{i=1}^{n} e(G_i, M_i)$$

Trapdoor opening: Given $\mathbf{c}$ for $(M_1, \ldots, M_n)$ and $R$. Open $\mathbf{c}$ to $(M'_1, \ldots, M'_n)$ as $R' = R \prod_{i=1}^{n} (M_i / M'_i)^{x_i}$:

$$e(G_R, R \prod (M_i / M'_i)^{x_i}) \prod e(G_i, M'_i) = e(G_R, R) \prod e(G_i, M_i) = \mathbf{c}$$

# Commitment Scheme for $n$ Messages

Setup: Generate $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$.

Key generation: Pick $G_R$ $\mathbb{G}_1^*$ and $x_1, \ldots, x_n$ $\mathbb{Z}_p$. Return

$$ck = (G_R, G_1 = G_R^{x_1}, \ldots, G_n = G_R^{x_n}) \quad \text{and} \quad tk = (x_1, \ldots, x_n).$$

Commitment: On input $ck$, $(M_1, \ldots, M_n)$ 2 $\mathbb{G}_2^n$, $R$ 2 $\mathbb{G}_2$, return

$$\mathbf{c} = e(G_R, R) \prod_{i=1}^{n} e(G_i, M_i)$$

Trapdoor opening: Given $\mathbf{c}$ for $(M_1, \ldots, M_n)$ and $R$. Open $\mathbf{c}$ to $(M_1', \ldots, M_n')$ as $R' = R \prod_{i=1}^{n} (M_i / M_i')^{x_i}$:

**Theorem**

The scheme above is a homomorphic, perfectly hiding, trapdoor commitment scheme; under the double pairing assumption it is computationally binding.

# Application

## Commitments to Pedersen commitments

Pedersen commitment $C = H^r \prod H_i^{m_i}$ to $(m_1, \ldots, m_k)$ 2 $\mathbb{Z}_p^k$

# Application

## Commitments to Pedersen commitments

Pedersen commitment $C = H^r \prod H_i^{m_i}$ to $(m_1, \ldots, m_k)$ 2 $\mathbb{Z}_p^k$

**c** commitment to $(C_1, \ldots, C_n)$ where $C_i$ commitment to $(m_{i,1}, \ldots, m_{i,k})$

) can commit to $m$ 2 $\mathbb{Z}_p^{n \cdot k}$; key: $n + k + 2$ group elements, **c** 2 $\mathbb{G}_T$

Resulting scheme still homomorphic and trapdoor

# Application

## Commitments to Pedersen commitments

Pedersen commitment $C = H^r \prod H_i^{m_i}$ to $(m_1, \ldots, m_k)$ $2$ $\mathbb{Z}_p^k$

c commitment to $(C_1, \ldots, C_n)$ where $C_i$ commitment to $(m_{i,1}, \ldots, m_{i,k})$

) can commit to $m$ $2$ $\mathbb{Z}_p^{n \cdot k}$; key: $n + k + 2$ group elements, c $2$ $\mathbb{G}_T$

Resulting scheme still homomorphic and trapdoor

## Variant

We give another scheme based on an assumption implied by DLIN

) instantiable in symmetric bilinear groups

# Groth-Sahai Proofs

Pairing-product equation over variables $X_1, \ldots, X_m \in \mathbb{G}_1$, $Y_1, \ldots, Y_n \in \mathbb{G}_2$

$$\prod_{i=1}^{n} e(A_i, Y_i) \prod_{i=1}^{m} e(X_i, B_i) \prod_{i=1}^{m} \prod_{j=1}^{n} e(X_i, Y_j)^{\gamma_{i,j}} = \mathbf{t}\ , \qquad \text{(E)}$$

determined by $A_i \in \mathbb{G}_1, B_i \in \mathbb{G}_2, \gamma_{i,j} \in \mathbb{Z}_p$ and $\mathbf{t} \in \mathbb{G}_T$

# Groth-Sahai Proofs

Pairing-product equation over variables $X_1, \ldots, X_m \in \mathbb{G}_1$, $Y_1, \ldots, Y_n \in \mathbb{G}_2$

$$\prod_{i=1}^{n} e(A_i, Y_i) \prod_{i=1}^{m} e(X_i, B_i) \prod_{i=1}^{m} \prod_{j=1}^{n} e(X_i, Y_j)^{\gamma_{i,j}} = \mathbf{t}\,, \qquad \text{(E)}$$

determined by $A_i \in \mathbb{G}_1, B_i \in \mathbb{G}_2, \gamma_{i,j} \in \mathbb{Z}_p$ and $\mathbf{t} \in \mathbb{G}_T$

Groth, Sahai [GS08]: Non-interactive witness-indistinguishable (and NIZK) proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

(Given a *trapdoor* for CRS, one can extract the witness)

# Motivation

**Structure-preserving signatures**

- Messages, signatures and verification keys are in $\mathbb{G}_1$ and $\mathbb{G}_2$
- Verification: evaluate PPEs on message, signature and key
- Unforgeable (under chosen-message attack)

Combined with Groth-Sahai proofs:

- Prove knowledge of a valid signature (and message)

# Motivation

## Structure-preserving signatures

- Messages, signatures and verification keys are in $\mathbb{G}_1$ and $\mathbb{G}_2$
- Verification: evaluate PPEs on message, signature and key
- Unforgeable (under chosen-message attack)

Combined with Groth-Sahai proofs:

- Prove knowledge of a valid signature (and message)

## Automorphic signatures

- Structure-preserving
- Verification keys lie in the message space

- Prove knowledge of chain of keys and certificates

# A Variant of SDH and a Variant of CDH

The strong Diffie-Hellman (SDH) assumption [BB04] implies hardness of

Given $G, G^x$ and $q$ $\quad$ 1 pairs $(G^{\frac{1}{x+c_i}}, c_i)$, output a new pair $(G^{\frac{1}{x+c}}, c)$

# A Variant of SDH and a Variant of CDH

The strong Diffie-Hellman (SDH) assumption [BB04] implies hardness of

Given $G, K, G^x, \left((K \quad G^{v_i})^{\frac{1}{x+c_i}}, c_i, v_i\right)_{i=1}^{q-1}$, output a new $((K \quad G^v)^{\frac{1}{x+c}}, c, v)$

# A Variant of SDH and a Variant of CDH

The strong Diffie-Hellman (SDH) assumption [BB04] implies hardness of

Given $G, K, G^x, \left((K \quad G^{v_i})^{\frac{1}{x+c_i}}, c_i, v_i\right)_{i=1}^{q-1}$, output a new $((K \quad G^v)^{\frac{1}{x+c}}, c, v)$

Analogously to [BW07] we define a hidden variant

**$q$ - Asymm. Double Hidden SDH**

Given $G, F, K, X = G^x \; 2 \; \mathbb{G}_1$, $H, Y = H^x \; 2 \; \mathbb{G}_2$ and $q \quad 1$ tuples

$$((K \quad G^{v_i})^{\frac{1}{x+c_i}}, \; F^{c_i}, \; H^{c_i}, \; G^{v_i}, \; H^{v_i})$$

it is hard to output $((K \quad G^v)^{\frac{1}{x+c}}, F^c, H^c, G^v, H^v)$ with $(c, v) \; 6= \; (c_i, v_i)$

# A Variant of SDH and a Variant of CDH

The strong Diffie-Hellman (SDH) assumption [BB04] implies hardness of

Given $G, K, G^x, \left((K \quad G^{v_i})^{\frac{1}{x+c_i}}, c_i, v_i\right)_{i=1}^{q-1}$, output a new $((K \quad G^v)^{\frac{1}{x+c}}, c, v)$

Analogously to [BW07] we define a hidden variant

## $q$ - Asymm. Double Hidden SDH

Given $G, F, K, X = G^x \; 2 \; \mathbb{G}_1$, $H, Y = H^x \; 2 \; \mathbb{G}_2$ and $q \quad 1$ tuples

$$((K \quad G^{v_i})^{\frac{1}{x+c_i}}, \; F^{c_i}, \; H^{c_i}, \; G^{v_i}, \; H^{v_i})$$

it is hard to output $((K \quad G^v)^{\frac{1}{x+c}}, F^c, H^c, G^v, H^v)$ with $(c, v) \; 6= \; (c_i, v_i)$

## Asymm. Weak Flexible CDH

Given $G, G^a$ and $H$ it is hard to output $(G^r, G^{ar}, H^r, H^{ar})$ with $r \; 6= \; 0$

# Automorphic Signatures: Instantiation

Setup: Choose $G, K, F, T \quad \mathbb{G}_1$, $H \quad \mathbb{G}_2$

Message space: $DH := f(G^m, H^m) \, j \, m \; 2 \; \mathbb{Z}_p g$,

# Automorphic Signatures: Instantiation

Setup: Choose $G, K, F, T \quad \mathbb{G}_1$, $H \quad \mathbb{G}_2$

Message space: $DH := f(G^m, H^m) \, j \, m \; 2 \; \mathbb{Z}_p g$,

KeyGen: Secret key $x \quad \mathbb{Z}_p$, public key $(X := G^x, Y := H^x)$

# Automorphic Signatures: Instantiation

Setup: Choose $G, K, F, T \quad \mathbb{G}_1$, $H \quad \mathbb{G}_2$

Message space: $DH := f(G^m, H^m) \, j \, m \; 2 \; \mathbb{Z}_p g$,

KeyGen: Secret key $x \quad \mathbb{Z}_p$, public key $(X := G^x, Y := H^x)$

Sign$(x, (M, N))$: Choose $c, r \quad \mathbb{Z}_p$, return

$$((K \quad T^r \quad M)^{\frac{1}{x+c}}, F^c, H^c, G^r, H^r)$$

# Automorphic Signatures: Instantiation

Setup: Choose $G, K, F, T \quad \mathbb{G}_1$, $H \quad \mathbb{G}_2$

Message space: $DH := f(G^m, H^m) \, j \, m \; 2 \; \mathbb{Z}_p g$,

KeyGen: Secret key $x \quad \mathbb{Z}_p$, public key $(X := G^x, Y := H^x)$

Sign$(x, (M, N))$: Choose $c, r \quad \mathbb{Z}_p$, return

$$((K \quad T^r \quad M)^{\frac{1}{x+c}}, F^c, H^c, G^r, H^r)$$

Ver$((X, Y), (M, N), (A, C, D, R, S))$: Return 1 if

$$e(A, Y \quad D) = e(K \quad M, H) \; e(T, S) \qquad \begin{array}{l} e(C, H) = e(F, D) \\ e(R, H) = e(G, S) \end{array}$$

# Automorphic Signatures: Instantiation

Setup: Choose $G, K, F, T \quad \mathbb{G}_1$, $H \quad \mathbb{G}_2$
Message space: $DH := f(G^m, H^m) \, j \, m \; 2 \; \mathbb{Z}_p g$,

KeyGen: Secret key $x \quad \mathbb{Z}_p$, public key $(X := G^x, Y := H^x)$

Sign$(x, (M, N))$: Choose $c, r \quad \mathbb{Z}_p$, return

$$((K \quad T^r \quad M)^{\frac{1}{x+c}}, F^c, H^c, G^r, H^r)$$

Ver$((X, Y), (M, N), (A, C, D, R, S))$: Return 1 if

$$e(A, Y \quad D) = e(K \quad M, H) \; e(T, S) \qquad e(C, H) = e(F, D)$$

**Theorem**

The scheme is strongly unforgeable under ADH-SDH and AWF-CDH.

# A Variant of the Double Pairing Assumption

**Double Pairing problem**: find non-trivial $Z, R$ s.t. $1 = e(G_Z, Z)\, e(G_R, R)$

is malleable: one solution $\Rightarrow$ multiple solutions

# A Variant of the Double Pairing Assumption

**Double Pairing problem**: find non-trivial $Z, R$ s.t. $1 = e(G_Z, Z)\, e(G_R, R)$

is malleable: one solution $\Rightarrow$ multiple solutions

- Make 2 simultaneous equations with common element $Z$

$\Rightarrow$ implied by DLIN

# A Variant of the Double Pairing Assumption

Double Pairing problem: find non-trivial $Z, R$ s.t. $1 = e(G_Z, Z)\, e(G_R, R)$

is malleable: one solution $\Rightarrow$ multiple solutions

- Make 2 simultaneous equations with common element $Z$
  $\Rightarrow$ implied by DLIN
- Multiply random pairings to both sides of equation (*flexible*)
  $\Rightarrow$ non-malleable

# A Variant of the Double Pairing Assumption

**Double Pairing problem**: find non-trivial $Z, R$ s.t. $1 = e(G_Z, Z)\, e(G_R, R)$

is malleable: one solution $\Rightarrow$ multiple solutions

- Make 2 simultaneous equations with common element $Z$

  $\Rightarrow$ implied by DLIN

- Multiply random pairings to both sides of equation (*flexible*)

  $\Rightarrow$ non-malleable

## $q$ - Simultaneous Flexible Pairing assumption (SFP)

Given $G_Z, F_Z, G_R, F_U, A, B \in \mathbb{G}_1$ and $\tilde{A}, \tilde{B} \in \mathbb{G}_2$ and $q$ tuples $(Z_i, R_i, S_i, T_i, U_i, V_i, W_i)$ s.t.

$$e(A, \tilde{A}) = e(G_Z, Z_i)\, e(G_R, R_i)\, e(S_i, T_i)$$
$$e(B, \tilde{B}) = e(F_Z, Z_i)\, e(F_U, U_i)\, e(V_i, W_i)$$

it is hard to find such a tuple $(Z, R, S, T, U, V, W)$ with $Z \neq 1$ and $Z \neq Z_i$ for all $i$

# A Variant of the Double Pairing Assumption

## $q$ - Simultaneous Flexible Pairing assumption (SFP)

Given $G_Z, F_Z, G_R, F_U, A, B \in \mathbb{G}_1$ and $\tilde{A}, \tilde{B} \in \mathbb{G}_2$ and $q$ tuples $(Z_i, R_i, S_i, T_i, U_i, V_i, W_i)$ s.t.

$$e(A, \tilde{A}) = e(G_Z, Z_i)\, e(G_R, R_i)\, e(S_i, T_i)$$
$$e(B, \tilde{B}) = e(F_Z, Z_i)\, e(F_U, U_i)\, e(V_i, W_i)$$

it is hard to find such a tuple $(Z, R, S, T, U, V, W)$ with $Z \neq 1$ and $Z \neq Z_i$ for all $i$

# A Variant of the Double Pairing Assumption

## $q$ - Simultaneous Flexible Pairing assumption (SFP)

Given $G_Z, F_Z, G_R, F_U, A, B \in \mathbb{G}_1$ and $\tilde{A}, \tilde{B} \in \mathbb{G}_2$ and $q$ tuples $(Z_i, R_i, S_i, T_i, U_i, V_i, W_i)$ s.t.

$$e(A, \tilde{A}) = e(G_Z, Z_i)\, e(G_R, R_i)\, e(S_i, T_i)$$
$$e(B, \tilde{B}) = e(F_Z, Z_i)\, e(F_U, U_i)\, e(V_i, W_i)$$

it is hard to find such a tuple $(Z, R, S, T, U, V, W)$ with $Z \neq 1$ and $Z \neq Z_i$ for all $i$

## Theorem

For a generic algorithm the probability of breaking SFP with $\ell$ operations is bounded by $O(q^2 + \ell^2)/p$

# Scheme Signing $k$ $\mathbb{G}_2$ Elements at Once

`Setup`: Choose a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$

# Scheme Signing $k$ $\mathbb{G}_2$ Elements at Once

Setup: Choose a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$

KeyGen: Message Space: $\mathbb{G}_2^k$

Choose secret key $(\alpha, \beta, \gamma_Z, \delta_Z, \gamma_1, \delta_1, \ldots, \gamma_k, \delta_k)$ $(\mathbb{Z}_p^*)^{2k+4}$

Public key: $G_R$ $\mathbb{G}_1^*, G_Z = G_R^{\gamma_Z}, fG_i = G_R^{\gamma_i} g_{i=1}^k, \mathbf{a} = e(G_R, H^\alpha)$

$F_U$ $\mathbb{G}_1^*, F_Z = F_U^{\delta_Z}, fF_i = F_U^{\delta_i} g_{i=1}^k, \mathbf{b} = e(F_U, H^\beta)$

# Scheme Signing $k$ $\mathbb{G}_2$ Elements at Once

Setup: Choose a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$

KeyGen: Message Space: $\mathbb{G}_2^k$

Choose secret key $(\alpha, \beta, \gamma_Z, \delta_Z, \gamma_1, \delta_1, \ldots, \gamma_k, \delta_k)$ $(\mathbb{Z}_p^*)^{2k+4}$

Public key: $G_R$ $\mathbb{G}_1^*, G_Z = G_R^{\gamma_Z}, fG_i = G_R^{\gamma_i} g_{i=1}^k, \mathbf{a} = e(G_R, H^\alpha)$

$F_U$ $\mathbb{G}_1^*, F_Z = F_U^{\delta_Z}, fF_i = F_U^{\delta_i} g_{i=1}^k, \mathbf{b} = e(F_U, H^\beta)$

Sign$(sk, (M_1, \ldots, M_k))$: Choose $\zeta, \rho, \tau, \varphi, \omega$ $\mathbb{Z}_p^*$, return

$Z = H^\zeta$ $R = H^{\rho - \gamma_Z \zeta} \prod_{i=1}^k M_i^{-\gamma_i}$ $S = G_R{}^\tau$ $T = H^{(\alpha-\rho)/\tau}$

$U = H^{\varphi - \delta_Z \zeta} \prod_{i=1}^k M_i^{-\delta_i}$ $V = F_U{}^\omega$ $W = H^{(\beta-\varphi)/\omega}$

# Scheme Signing $k$ $\mathbb{G}_2$ Elements at Once

Setup: Choose a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$

KeyGen: Message Space: $\mathbb{G}_2^k$

Choose secret key $(\alpha, \beta, \gamma_Z, \delta_Z, \gamma_1, \delta_1, \ldots, \gamma_k, \delta_k)$ $(\mathbb{Z}_p^*)^{2k+4}$

Public key: $G_R$ $\mathbb{G}_1^*, G_Z = G_R^{\gamma_Z}, fG_i = G_R^{\gamma_i} g_{i=1}^k, \mathbf{a} = e(G_R, H^\alpha)$

$F_U$ $\mathbb{G}_1^*, F_Z = F_U^{\delta_Z}, fF_i = F_U^{\delta_i} g_{i=1}^k, \mathbf{b} = e(F_U, H^\beta)$

Sign$(sk, (M_1, \ldots, M_k))$: Choose $\zeta, \rho, \tau, \varphi, \omega$ $\mathbb{Z}_p^*$, return

$Z = H^\zeta$ $R = H^{\rho - \gamma_Z \zeta} \prod_{i=1}^k M_i^{-\gamma_i}$ $S = G_R{}^\tau$ $T = H^{(\alpha-\rho)/\tau}$

$U = H^{\varphi - \delta_Z \zeta} \prod_{i=1}^k M_i^{-\delta_i}$ $V = F_U{}^\omega$ $W = H^{(\beta-\varphi)/\omega}$

Ver$(vk, (M_1, \ldots, M_k), (Z, R, S, T, U, V, W))$: Return 1 if

$$\mathbf{a} = e(G_Z, Z)\, e(G_R, R)\, e(S, T) \prod_{i=1}^k e(G_i, M_i)$$

$$\mathbf{b} = e(F_Z, Z)\, e(F_U, U)\, e(V, W) \prod_{i=1}^k e(F_i, M_i)$$

# Scheme Signing $k$ $\mathbb{G}_2$ Elements at Once

Setup: Choose a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$

KeyGen: Message Space: $\mathbb{G}_2^k$

Choose secret key $(\alpha, \beta, \gamma_Z, \delta_Z, \gamma_1, \delta_1, \ldots, \gamma_k, \delta_k)$ $(\mathbb{Z}_p^*)^{2k+4}$

Public key: $G_R$ $\mathbb{G}_1^*, G_Z = G_R^{\gamma_Z}, fG_i = G_R^{\gamma_i} g_{i=1}^k, \mathbf{a} = e(G_R, H^\alpha)$

$F_U$ $\mathbb{G}_1^*, F_Z = F_U^{\delta_Z}, fF_i = F_U^{\delta_i} g_{i=1}^k, \mathbf{b} = e(F_U, H^\beta)$

Sign$(sk, (M_1, \ldots, M_k))$: Choose $\zeta, \rho, \tau, \varphi, \omega$ $\mathbb{Z}_p^*$, return

$Z = H^\zeta$ $R = H^{\rho - \gamma_Z \zeta} \prod_{i=1}^k M_i^{-\gamma_i}$ $S = G_R{}^\tau$ $T = H^{(\alpha-\rho)/\tau}$

$U = H^{\varphi - \delta_Z \zeta} \prod_{i=1}^k M_i^{-\delta_i}$ $V = F_U{}^\omega$ $W = H^{(\beta-\varphi)/\omega}$

Ver$(vk, (M_1, \ldots, M_k), (Z, R, S, T, U, V, W))$: Return 1 if

**Theorem**

The scheme is existentially unforgeable under the SFP assumption

# Variants of the Scheme

- Given $(Z, R, S, T, U, V, W)$, we can randomise $(R, S, T, U, V, W)$
- Replace $\mathbf{a}$ by random $A_0, \tilde{A}_0, A_1, \tilde{A}_1$ with $\mathbf{a} = e(A_0, \tilde{A}_0)\, e(A_1, \tilde{A}_1)$ and $\mathbf{b}$ analogously

  ) Verification key from $\mathbb{G}_1$ and $\mathbb{G}_2$ ) structure preserving

# Variants of the Scheme

- Given $(Z, R, S, T, U, V, W)$, we can randomise $(R, S, T, U, V, W)$
- Replace $\mathbf{a}$ by random $A_0, \tilde{A}_0, A_1, \tilde{A}_1$ with $\mathbf{a} = e(A_0, \tilde{A}_0)\, e(A_1, \tilde{A}_1)$ and $\mathbf{b}$ analogously

  ) Verification key from $\mathbb{G}_1$ and $\mathbb{G}_2$ ) structure preserving
- Dual scheme for signing messages in $\mathbb{G}_1^k$

  ) combine both schemes to sign messages in $\mathbb{G}_1^{k_1}$ $\mathbb{G}_2^{k_2}$

# Variants of the Scheme

- Given $(Z, R, S, T, U, V, W)$, we can randomise $(R, S, T, U, V, W)$
- Replace $\mathbf{a}$ by random $A_0, \tilde{A}_0, A_1, \tilde{A}_1$ with $\mathbf{a} = e(A_0, \tilde{A}_0)\, e(A_1, \tilde{A}_1)$ and $\mathbf{b}$ analogously

  ) Verification key from $\mathbb{G}_1$ and $\mathbb{G}_2$ ) structure preserving
- Dual scheme for signing messages in $\mathbb{G}_1^k$

  ) combine both schemes to sign messages in $\mathbb{G}_1^{k_1}$ $\mathbb{G}_2^{k_2}$
- Chaining signatures to sign unbounded messages ) automorphic

# Variants of the Scheme

- Given $(Z, R, S, T, U, V, W)$, we can randomise $(R, S, T, U, V, W)$
- Replace $\mathbf{a}$ by random $A_0, \tilde{A}_0, A_1, \tilde{A}_1$ with $\mathbf{a} = e(A_0, \tilde{A}_0)\, e(A_1, \tilde{A}_1)$ and $\mathbf{b}$ analogously

  ) Verification key from $\mathbb{G}_1$ and $\mathbb{G}_2$ ) structure preserving
- Dual scheme for signing messages in $\mathbb{G}_1^k$

  ) combine both schemes to sign messages in $\mathbb{G}_1^{k_1}$ $\mathbb{G}_2^{k_2}$
- Chaining signatures to sign unbounded messages ) automorphic

**Simulatable Signatures**

- Signature scheme in the common reference string (CRS) model
- Trapdoor for CRS allows making signatures for any public key

Can use WI instead of ZK proofs, since signatures can be simulated directly

1. Commitments
2. Automorphic Signatures
3. Signatures on Vectors of Group Elements
4. Applications of Our Signatures

# Round-Optimal Blind Signatures

A blind signature scheme allows a *user U* to obtain a signature on a message hidden from the *signer S*

Round optimal: Signature issuing: $m \;!\; U \;\;!\; S$

$U \;\; S$

.

$\Sigma$

# Round-Optimal Blind Signatures

A blind signature scheme allows a *user* $U$ to obtain a signature on a message hidden from the *signer* $S$

Round optimal: Signature issuing: $m$ ! $U$ ! $S$

$U$ $S$

.

$\Sigma$

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message $m$
- Signer makes signature $\sigma$ on $C$
- Blind signature: proof of knowledge (PoK) of

$C$ $\sigma$ an opening of $C$ to $m$

# Round-Optimal Blind Signatures

A blind signature scheme allows a *user* $U$ to obtain a signature on a message hidden from the *signer* $S$

Round optimal: Signature issuing: $m$ ! $U$ ! $S$

$U$ $S$

.

$\Sigma$

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message $m$ (Pedersen)
- Signer makes signature $\sigma$ on $C$ (structure-preserving)
- Blind signature: proof of knowledge (PoK) of (Groth-Sahai)

$C$ $\sigma$ an opening of $C$ to $m$

# Round-Optimal Blind Signatures

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message $m$
- Signer makes signature $\sigma$ on $C$
- Blind signature: proof of knowledge (PoK) of

$$C \qquad \sigma \qquad \text{an opening of } C \text{ to } m$$

# Round-Optimal Blind Signatures

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message $m$
- Signer makes signature $\sigma$ on $C$
- Blind signature: proof of knowledge (PoK) of

$$C \qquad \sigma \qquad \text{an opening of } C \text{ to } m$$

**Variant I** Round-opt. automorphic blind signature

- Message from group, user gets signature *on message*

# Round-Optimal Blind Signatures

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message ~~$m$~~ $M$
- Signer makes ~~signature $\sigma$ on $C$~~ *pre-signature*; User recovers $\sigma$ on $M$
- Blind signature: proof of knowledge (PoK) of

~~$C$~~ $\sigma$ ~~an opening of $C$ to $m$~~

**Variant I** Round-opt. automorphic blind signature

- Message from group, user gets signature *on message*

# Round-Optimal Blind Signatures

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message $m$
- Signer makes signature $\sigma$ on $C$
- Blind signature: proof of knowledge (PoK) of

$$C \qquad \sigma \qquad \text{an opening of } C \text{ to } m$$

**Variant I** Round-opt. automorphic blind signature

- Message from group, user gets signature *on message*

# Round-Optimal Blind Signatures

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message $m$
- Signer makes signature $\sigma$ on $C$
- Blind signature: proof of knowledge (PoK) of

$$C \qquad \sigma \qquad \text{an opening of } C \text{ to } m$$

**Variant I** Round-opt. automorphic blind signature

- Message from group, user gets signature *on message*

**Variant II** Universally composable round-opt. blind signature

- Use *simulatable* signature

# Round-Optimal Blind Signatures

## Sketch of the scheme [Fis06]

- User makes a commitment $C$ to the message $m$
- Signer makes signature $\sigma$ on $C$ (simulatable!)
- Blind signature: proof of knowledge (PoK) of

$$C \qquad \sigma \qquad \text{an opening of } C \text{ to } m$$

**Variant I** Round-opt. automorphic blind signature

- Message from group, user gets signature *on message*

**Variant II** Universally composable round-opt. blind signature

- Use *simulatable* signature

# Group Signatures

A group signature scheme lets a *group manager* enrol *users* who can then sign on behalf of the group anonymously. The anonymity is revocable by an *opener*

# Group Signatures

A group signature scheme lets a *group manager* enrol *users* who can then sign on behalf of the group anonymously. The anonymity is revocable by an *opener*

Automorphic signatures enable efficient instantiation of the following (satisfying model from [BSZ05])

## Group signatures with concurrent join

- *Opener* generates CRS for proof system, keeps trapdoor
- *Group manager* (GM) generates verification key, keeps signing key

# Group Signatures

A group signature scheme lets a *group manager* enrol *users* who can then sign on behalf of the group anonymously. The anonymity is revocable by an *opener*

Automorphic signatures enable efficient instantiation of the following (satisfying model from [BSZ05])

## Group signatures with concurrent join

- *Opener* generates CRS for proof system, keeps trapdoor
- *Group manager* (GM) generates verification key, keeps signing key
- Enrol: User creates signature key pair $(uvk, usk)$, GM signs $uvk$

# Group Signatures

A group signature scheme lets a *group manager* enrol *users* who can then sign on behalf of the group anonymously. The anonymity is revocable by an *opener*

Automorphic signatures enable efficient instantiation of the following (satisfying model from [BSZ05])

## Group signatures with concurrent join

- *Opener* generates CRS for proof system, keeps trapdoor
- *Group manager* (GM) generates verification key, keeps signing key
- Enrol: User creates signature key pair $(uvk, usk)$, GM signs $uvk$
- Group signature on M: Make signature $\sigma$ on $M$ with $usk$, and PoK of

$$uvk \qquad \text{signature on } uvk \text{ by GM} \qquad \sigma$$

# Group Signatures

A group signature scheme lets a *group manager* enrol *users* who can then sign on behalf of the group anonymously. The anonymity is revocable by an *opener*

Automorphic signatures enable efficient instantiation of the following (satisfying model from [BSZ05])

## Group signatures with concurrent join

- *Opener* generates CRS for proof system, keeps trapdoor
- *Group manager* (GM) generates verification key, keeps signing key
- Enrol: User creates signature key pair $(uvk, usk)$, GM signs $uvk$
- Group signature on M: Make signature $\sigma$ on $M$ with $usk$, and PoK of

$$uvk \qquad \text{signature on } uvk \text{ by GM} \qquad \sigma$$

- Open: Opener extracts $uvk$ and $\sigma$

# Anonymous Proxy Signatures

Anonymous proxy signatures [FP08]

- Generalisation of *group signatures* and *proxy signatures*
- Users hold signature key pairs
- Users can *delegate* signing rights to other users
- Users can *re-delegate* and make *proxy signatures* anonymously
- Anonymity revocable by openers

# Anonymous Proxy Signatures

Anonymous proxy signatures [FP08]

- Generalisation of *group signatures* and *proxy signatures*
- Users hold signature key pairs
- Users can *delegate* signing rights to other users
- Users can *re-delegate* and make *proxy signatures* anonymously
- Anonymity revocable by openers

## Instantiation

- Automorphic signatures ) delegation by signing public keys
- GS proof ) proxy signature is PoK of delegation chain

# Conclusion

## Commitments

- First homomorphic trapdoor commitments to group elements
- Used them to construct more efficient schemes

## Signatures

- First signature schemes that are fully "Groth-Sahai compatible"
- Various extensions
- Exemplified their usefulness

Combined with Groth-Sahai proofs, structure-preserving signatures lead to modular instantiations of more complex primitives

# Conclusion

## Commitments

- First homomorphic trapdoor commitments to group elements
- Used them to construct more efficient schemes

## Signatures

- First signature schemes that are fully "Groth-Sahai compatible"
- Various extensions
- Exemplified their usefulness

Combined with Groth-Sahai proofs, structure-preserving signatures lead to modular instantiations of more complex primitives

Thank you! ☺