Proofs on Encrypted Values in Bilinear Groups and an Application to Anonymity of Signatures

G. Fuchsbauer D. Pointcheval

École normale supérieure

Pairing'09, 13.08.2009

Anonymous Consecutive Delegation of Signing Rights

Fuchsbauer, Pointcheval: Anonymous Proxy Signatures [SCN'08]

Delegation A delegator delegates his signing rights to a proxy signer (or delegatee) who can then sign on the delegator's behalf

Consecutiveness A delegatee may re-delegate the received signing rights
$\Rightarrow$ intermediate delegators

Anonymity All intermediate delegators and the proxy signer remain anonymous

Anonymous Consecutive Delegation of Signing Rights

Fuchsbauer, Pointcheval: Anonymous Proxy Signatures [SCN'08]

Delegation A delegator delegates his signing rights to a proxy signer (or delegatee) who can then sign on the delegator's behalf

Consecutiveness A delegatee may re-delegate the received signing rights $\Rightarrow$ intermediate delegators

Anonymity All intermediate delegators and the proxy signer remain anonymous

Anonymous Consecutive Delegation of Signing Rights

Fuchsbauer, Pointcheval: Anonymous Proxy Signatures [SCN'08]

Delegation	A delegator delegates his signing rights to a proxy signer (or delegatee) who can then sign on the delegator's behalf
Consecutiveness	A delegatee may re-delegate the received signing rights $\Rightarrow$ intermediate delegators
Anonymity	All intermediate delegators and the proxy signer remain anonymous

Anonymous Consecutive Delegation of Signing Rights

Fuchsbauer, Pointcheval: Anonymous Proxy Signatures [SCN'08]

Delegation	A delegator delegates his signing rights to a proxy signer (or delegatee) who can then sign on the delegator's behalf
Consecutiveness	A delegatee may re-delegate the received signing rights $\Rightarrow$ intermediate delegators
Anonymity	All intermediate delegators and the proxy signer remain anonymous

After verifying a proxy signature one knows that someone entitled signed but nothing more.

Application: GRID computing

User authenticates herself and starts process which needs to authenticate to resources / start subprocesses

$\Rightarrow$ Delegation and re-delegation of signing rights

No need to know that it was not the user herself to be authenticated

Relation to Other Primitives

Anonymous proxy signatures are a generalization of

- Proxy signatures (consecutive delegation)
  formalized by [BPW03]
- (Dynamic) Group signatures (anonymity)
  formalized by [BSZ05]

and satisfy the respective security notions.

Application: GRID computing

User authenticates herself and starts process which needs to authenticate to resources / start subprocesses

$\Rightarrow$ Delegation and re-delegation of signing rights

No need to know that it was not the user herself to be authenticated

Relation to Other Primitives

Anonymous proxy signatures are a generalization of

- Proxy signatures (consecutive delegation)
  formalized by [BPW03]
- (Dynamic) Group signatures (anonymity)
  formalized by [BSZ05]

and satisfy the respective security notions.

Application: GRID computing

User authenticates herself and starts process which needs to authenticate to resources / start subprocesses

$\Rightarrow$ Delegation and re-delegation of signing rights

No need to know that it was not the user herself to be authenticated

Relation to Other Primitives

Anonymous proxy signatures are a generalization of

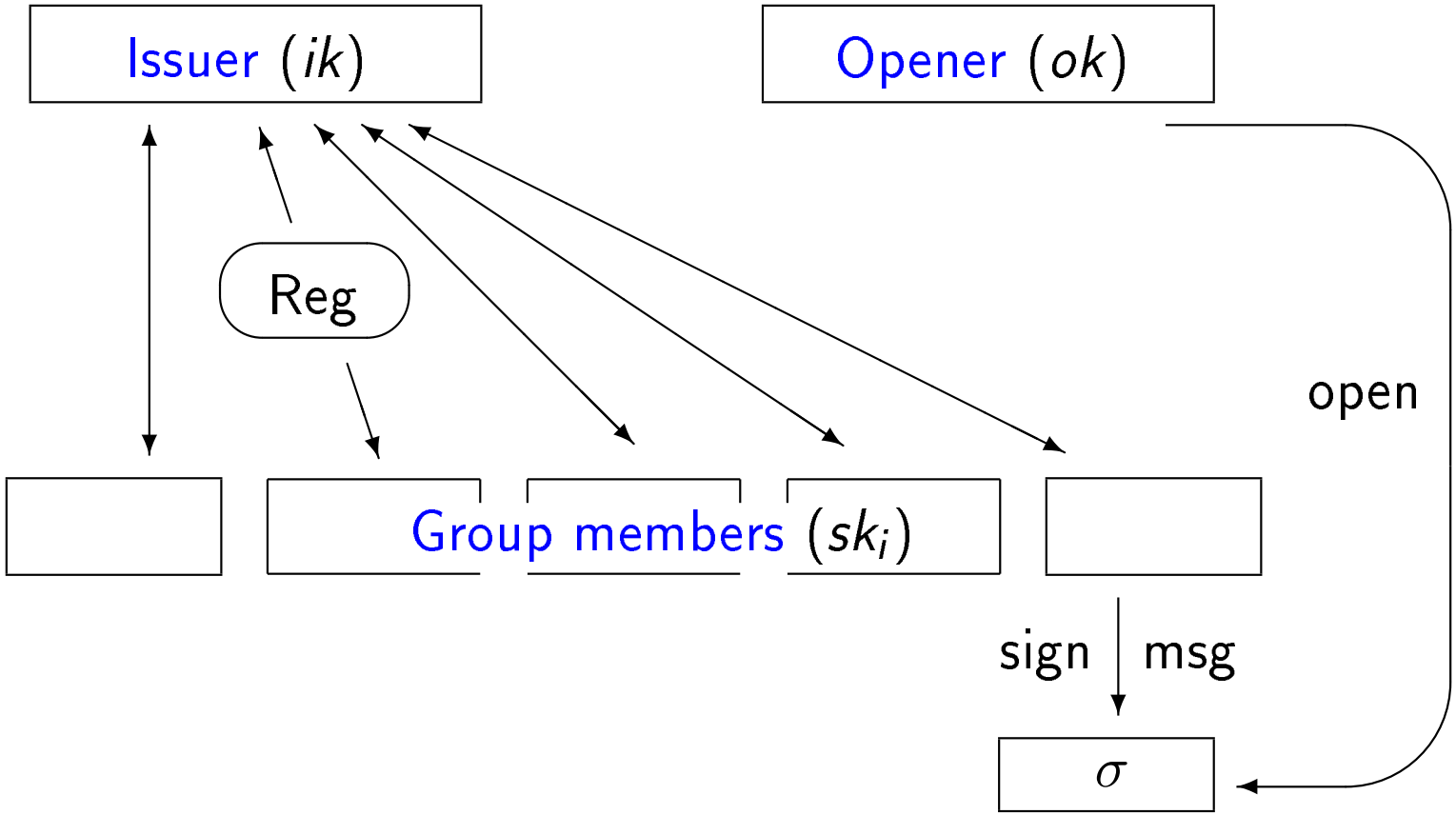
- Proxy signatures (consecutive delegation)
  formalized by [BPW03]
- (Dynamic) Group signatures (anonymity)
  formalized by [BSZ05]

and satisfy the respective security notions.

- recently: Delegatable Anonymous Credentials [BCCKLS09]

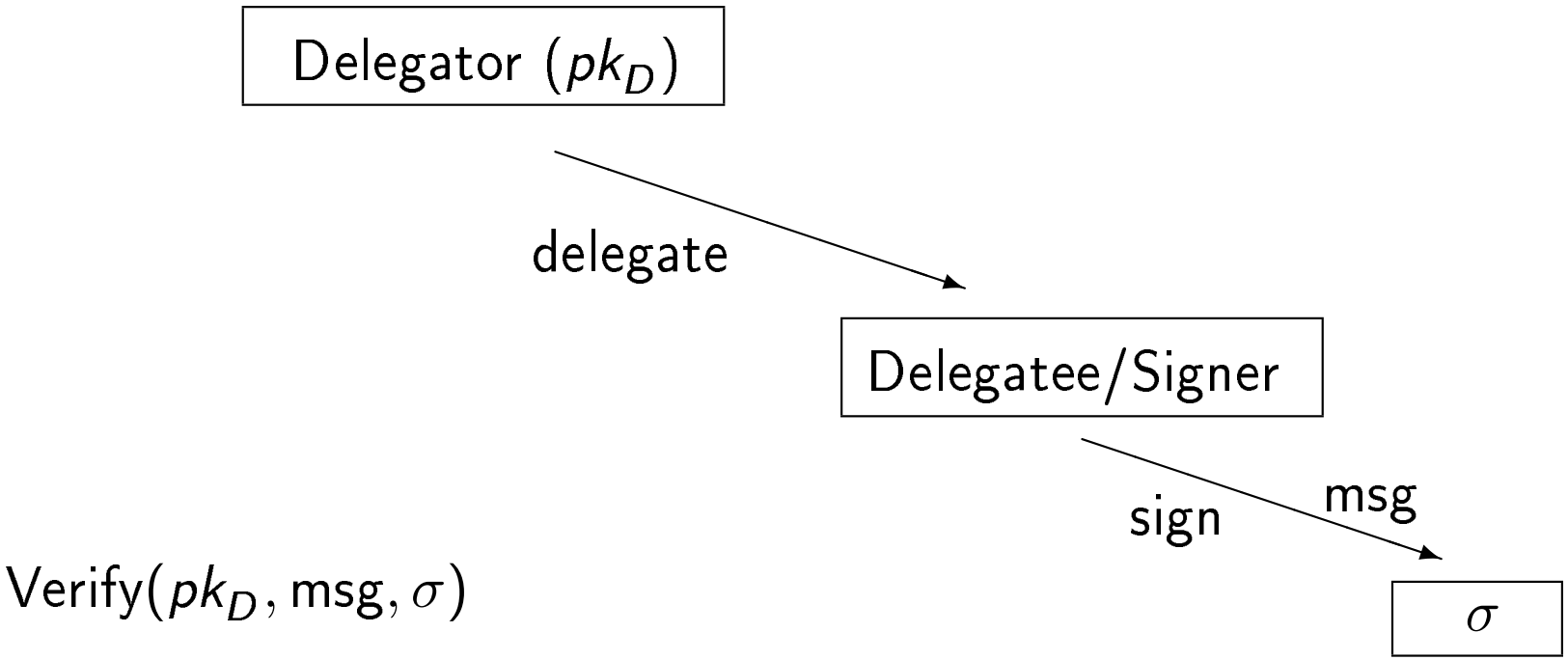
(Dynamic) Group Signatures

Group public key: pk



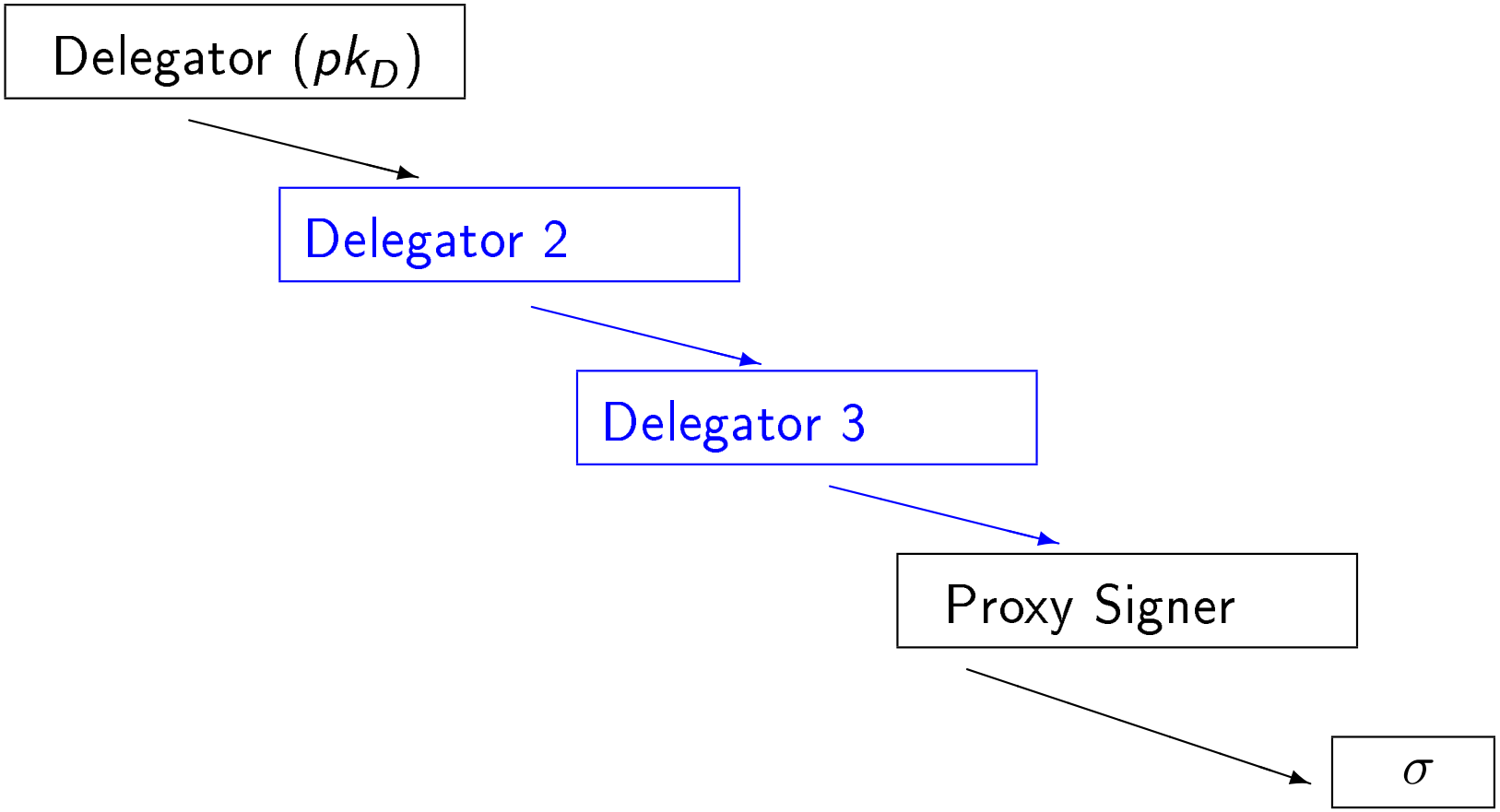
Verification: $\mathsf{Verify}(pk, \mathsf{msg}, \sigma) = 1$

Proxy Signatures

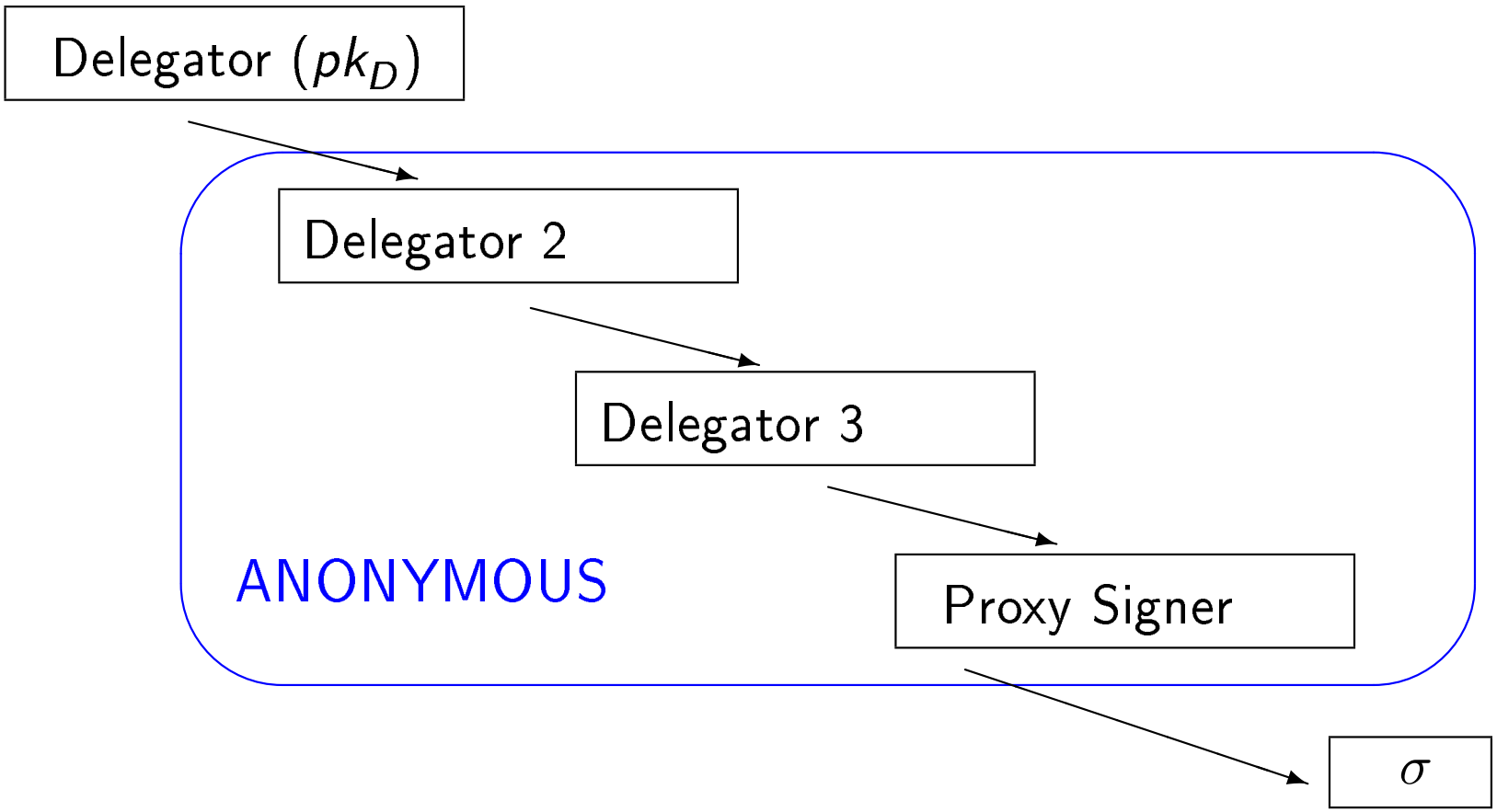


$\text{Verify}(pk_D, \text{msg}, \sigma)$

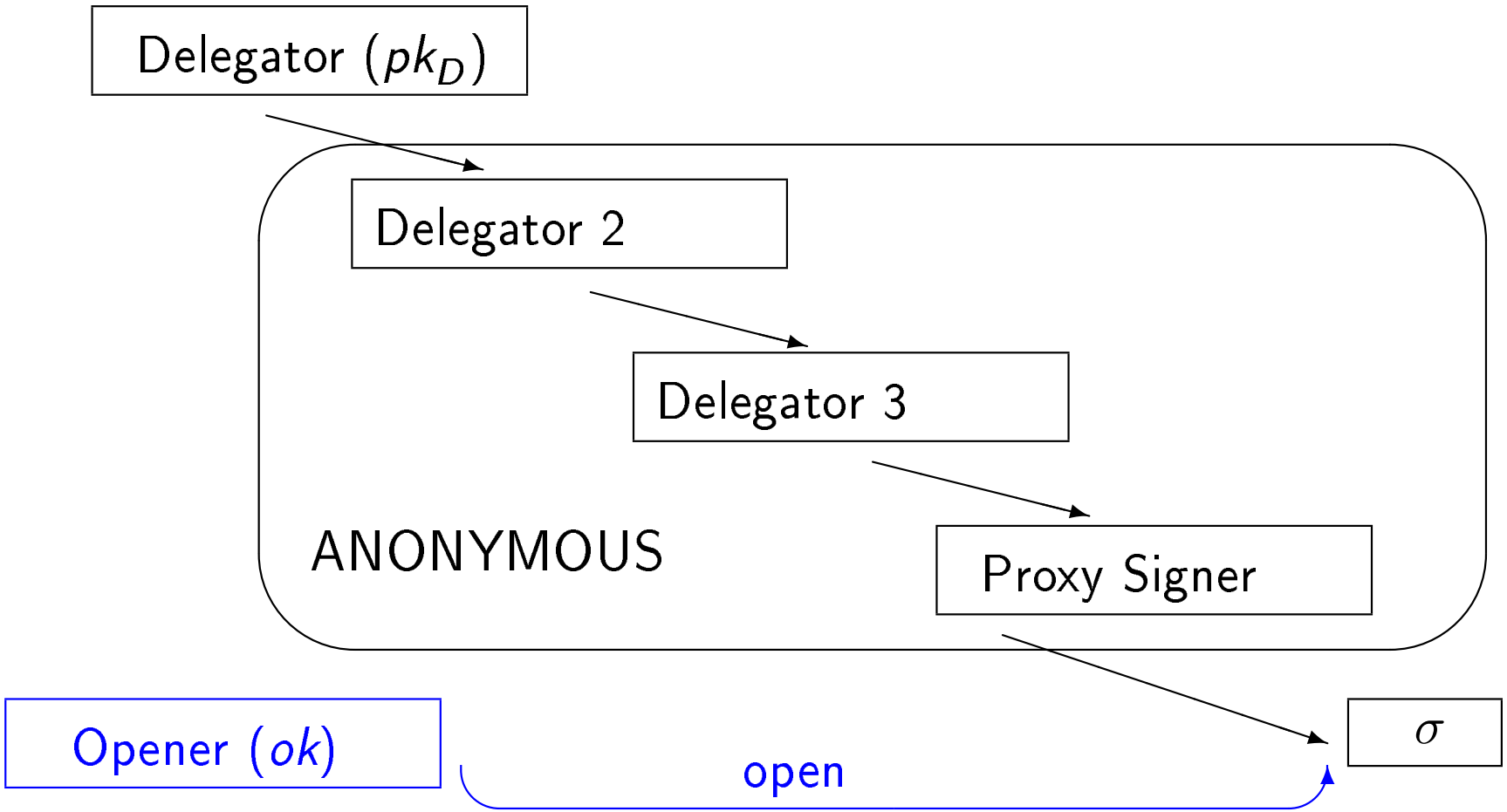
Proxy Signatures, Consecutive Delegations



Proxy Signatures, Consecutive Delegations



Proxy Signatures, Consecutive Delegations



Algorithms of Anonymous Proxy Signature Scheme

$$1^\lambda \quad \longrightarrow \quad \text{Setup} \quad \longrightarrow \quad pp, ik, ok$$

Algorithms of Anonymous Proxy Signature Scheme

$$1^{\lambda} \longrightarrow \text{Setup} \longrightarrow pp, ik, ok$$

Algorithms of Anonymous Proxy Signature Scheme

$$1^{\lambda} \longrightarrow \text{Setup} \longrightarrow pp, ik, ok$$

$$sk_x, [warr_{\rightarrow x},]\ pk_y \longrightarrow \text{Del} \longrightarrow warr_{[\rightarrow]x \rightarrow y}$$

Algorithms of Anonymous Proxy Signature Scheme

$$1^{\lambda} \longrightarrow \text{Setup} \longrightarrow pp, ik, ok$$

$$sk_x, [warr_{\rightarrow x},]\ pk_y \longrightarrow \text{Del} \longrightarrow warr_{[\rightarrow]x \rightarrow y}$$

$$sk_y, warr_{x \rightarrow \ldots \rightarrow y}, M \longrightarrow \text{PSig} \longrightarrow \sigma$$

Algorithms of Anonymous Proxy Signature Scheme

$$1^{\lambda} \longrightarrow \text{Setup} \longrightarrow pp, ik, ok$$

$$sk_x, [warr_{\rightarrow x},]\ pk_y \longrightarrow \text{Del} \longrightarrow warr_{[\rightarrow]x \rightarrow y}$$

$$sk_y, warr_{x \rightarrow \ldots \rightarrow y}, M \longrightarrow \text{PSig} \longrightarrow \sigma$$

$$pk_x, M, \sigma \longrightarrow \text{PVer} \longrightarrow b \in \{0,1\}$$

Algorithms of Anonymous Proxy Signature Scheme

$$1^\lambda \longrightarrow \text{Setup} \longrightarrow pp, ik, ok$$

$$sk_x, [warr_{\rightarrow x},]\ pk_y \longrightarrow \text{Del} \longrightarrow warr_{[\rightarrow]x\rightarrow y}$$

$$sk_y, warr_{x\rightarrow\ldots\rightarrow y}, M \longrightarrow \text{PSig} \longrightarrow \sigma$$

$$pk_x, M, \sigma \longrightarrow \text{PVer} \longrightarrow b \in \{0,1\}$$

$$ok, M, \sigma \longrightarrow \text{Open} \longrightarrow \text{a list of users or } \perp \text{ (failure)}$$

Security for Anonymous Proxy Signatures

Anonymity	intermediate delegators and proxy signer remain anonymous
Traceability	every valid signature can be traced to its intermediate delegators and proxy signer
Non-Frameability	no one can produce a signature that, when opened, wrongfully reveals a delegator or signer

Security for Anonymous Proxy Signatures

Anonymity	intermediate delegators and proxy signer remain anonymous
Traceability	every valid signature can be traced to its intermediate delegators and proxy signer
Non-Frameability	no one can produce a signature that, when opened, wrongfully reveals a delegator or signer

Security for Anonymous Proxy Signatures

Anonymity	intermediate delegators and proxy signer remain anonymous
Traceability	every valid signature can be traced to its intermediate delegators and proxy signer
Non-Frameability	no one can produce a signature that, when opened, wrongfully reveals a delegator or signer

Generic Construction, Ingredients

Generic Construction

using

- Digital signatures (EUF-CMA)
- Public-key encryption (IND-CPA)
- Non-interactive zero-knowledge proofs

Generic Construction, Ingredients

Generic Construction

using

- Digital signatures (EUF-CMA)
- Public-key encryption (IND-CPA)
- Non-interactive zero-knowledge proofs

(existence follows from trapdoor permutations)

Generic Construction, Overview

Setup Generates decryption key for opening authority; signing key for issuer
Parameters: resp. public keys, *crs* for NIZK

Register Issuer signs user's public key $\longrightarrow$ *certificate*

Delegate Sign delegatee's public key $\longrightarrow$ *warrant*
Re-delegate: additionally forward received warrants

Proxy-Sign Sign message, encrypt

- delegators' verification keys and certificates
- warrants
- signature on message

NIZK that plaintext contains valid signatures

Verify Verify NIZK

Open Decrypt ciphertext

Generic Construction, Overview

Setup Generates decryption key for opening authority; signing key for issuer
Parameters: resp. public keys, *crs* for NIZK

Register Issuer signs user's public key $\longrightarrow$ *certificate*

Delegate Sign delegatee's public key $\longrightarrow$ *warrant*
Re-delegate: additionally forward received warrants

Proxy-Sign Sign message, encrypt

- delegators' verification keys and certificates
- warrants
- signature on message

NIZK that plaintext contains valid signatures

Verify Verify NIZK

Open Decrypt ciphertext

Generic Construction, Overview

Setup Generates decryption key for opening authority; signing key for issuer
Parameters: resp. public keys, *crs* for NIZK

Register Issuer signs user's public key $\longrightarrow$ *certificate*

Delegate Sign delegatee's public key $\longrightarrow$ *warrant*
Re-delegate: additionally forward received warrants

Proxy-Sign Sign message, encrypt

- delegators' verification keys and certificates
- warrants
- signature on message

NIZK that plaintext contains valid signatures

Verify Verify NIZK

Open Decrypt ciphertext

Generic Construction, Overview

Setup Generates decryption key for opening authority; signing key for issuer
Parameters: resp. public keys, *crs* for NIZK

Register Issuer signs user's public key $\longrightarrow$ *certificate*

Delegate Sign delegatee's public key $\longrightarrow$ *warrant*
Re-delegate: additionally forward received warrants

Proxy-Sign Sign message, encrypt

- delegators' verification keys and certificates
- warrants
- signature on message

NIZK that plaintext contains valid signatures

Verify Verify NIZK

Open Decrypt ciphertext

Generic Construction, Overview

Setup Generates decryption key for opening authority; signing key for issuer
Parameters: resp. public keys, *crs* for NIZK

Register Issuer signs user's public key $\longrightarrow$ *certificate*

Delegate Sign delegatee's public key $\longrightarrow$ *warrant*
Re-delegate: additionally forward received warrants

Proxy-Sign Sign message, encrypt

- delegators' verification keys and certificates
- warrants
- signature on message

NIZK that plaintext contains valid signatures

Verify Verify NIZK

Open Decrypt ciphertext

Generic Construction, Overview

Setup Generates decryption key for opening authority; signing key for issuer
Parameters: resp. public keys, *crs* for NIZK

Register Issuer signs user's public key $\longrightarrow$ *certificate*

Delegate Sign delegatee's public key $\longrightarrow$ *warrant*
Re-delegate: additionally forward received warrants

Proxy-Sign Sign message, encrypt

- delegators' verification keys and certificates
- warrants
- signature on message

NIZK that plaintext contains valid signatures

Verify Verify NIZK

Open Decrypt ciphertext

The Subgroup Decision Assumption and BGN Encryption

Let $\mathbf{G} = (n, \mathbb{G}, \mathbb{G}_T, e, g)$ be a bilinear group of order $|\mathbb{G}| = n = pq$, where p, q prime.

Subgroup Decision Assumption (SD)

No p.p.t. adversary can distinguish a random element of $\mathbb{G}$ from a random element of $\mathbb{G}_q$, the subgroup of order q.

Boneh-Goh-Nissim (BGN) Encryption [TCC'05]

KeyGen Generate bilinear group $\mathbf{G}$ with $|\mathbb{G}| = n = pq$, choose
$h \leftarrow \mathbb{G}_q$.
$sk = q$.

Encrypt
- Choose $r \leftarrow \mathbb{Z}_n$
- Set $C := g^m h^r$.

Decrypt $C^q = (g^m h^r)^q = (g^q)^m$, thus $\log_{g^q} C^q = m$.

The Subgroup Decision Assumption and BGN Encryption

Let $\mathbf{G} = (n, \mathbb{G}, \mathbb{G}_T, e, g)$ be a bilinear group of order $|\mathbb{G}| = n = pq$, where p, q prime.

Subgroup Decision Assumption (SD)

No p.p.t. adversary can distinguish a random element of $\mathbb{G}$ from a random element of $\mathbb{G}_q$, the subgroup of order q.

Boneh-Goh-Nissim (BGN) Encryption [TCC'05]

KeyGen Generate bilinear group $\mathbf{G}$ with $|\mathbb{G}| = n = pq$, choose $h \leftarrow \mathbb{G}_q$.
$sk = q$.

Encrypt
- Choose $r \leftarrow \mathbb{Z}_n$
- Set $C := g^m h^r$.

Decrypt $C^q = (g^m h^r)^q = (g^q)^m$, thus $\log_{g^q} C^q = m$.

Boyen Waters Group Signatures [PKC'07] I

Hybrid Scheme

Setup

- Choose bilinear group $\mathbf{G}$
- Choose parameters for Waters signatures $(g, g_2, \vec{u}) \in \mathbb{G}^{m+3}$
- Choose $\omega, \alpha \leftarrow \mathbb{Z}_p$,
- Define $pp = (g, g_2, \vec{u}, \Omega = g^{\omega}, A = e(g,g)^{\alpha})$ $mk = (\omega, g^{\alpha})$

Extract

- Choose $s_i \leftarrow \mathbb{Z}_p$
- Return $K_i = ((g^{\alpha})^{\frac{1}{\omega + s_i}}, g^{s_i}, g_2^{s_i})$

Sign

- Choose $r \leftarrow \mathbb{Z}_p$ let $\mathcal{F}(M) := u_0 \prod_{i=1}^{m} u_i^{M_i}$
- Return $S = (K_1, K_2, K_3 \cdot \mathcal{F}(M)^r, g^{-r}) \in \mathbb{G}^4$

Verify

$$e(S_1, S_2\Omega) \cdot A^{-1} \stackrel{?}{=} 1$$

$$e(S_2, g_2)^{-1} \cdot e(g, S_3) \cdot e(S_4, \mathcal{F}(M)) \stackrel{?}{=} 1$$

Boyen Waters Group Signatures [PKC'07] I

Hybrid Scheme

Setup

- Choose bilinear group $\mathbf{G}$
- Choose parameters for Waters signatures $(g, g_2, \vec{u}) \in \mathbb{G}^{m+3}$
- Choose $\omega, \alpha \leftarrow \mathbb{Z}_p$,
- Define $pp = (g, g_2, \vec{u}, \Omega = g^{\omega}, A = e(g,g)^{\alpha})$ $mk = (\omega, g^{\alpha})$

Extract

- Choose $s_i \leftarrow \mathbb{Z}_p$
- Return $K_i = ((g^{\alpha})^{\frac{1}{\omega + s_i}}, g^{s_i}, g_2^{s_i})$

Sign

- Choose $r \leftarrow \mathbb{Z}_p$ let $\mathcal{F}(M) := u_0 \prod_{i=1}^{m} u_i^{M_i}$
- Return $S = (K_1, K_2, K_3 \cdot \mathcal{F}(M)^r, g^{-r}) \in \mathbb{G}^4$

Verify

$$e(S_1, S_2\Omega) \cdot A^{-1} \stackrel{?}{=} 1$$

$$e(S_2, g_2)^{-1} \cdot e(g, S_3) \cdot e(S_4, \mathcal{F}(M)) \stackrel{?}{=} 1$$

Boyen Waters Group Signatures [PKC'07] I

Hybrid Scheme

Setup

- Choose bilinear group $\mathbf{G}$
- Choose parameters for Waters signatures $(g, g_2, \vec{u}) \in \mathbb{G}^{m+3}$
- Choose $\omega, \alpha \leftarrow \mathbb{Z}_p$,
- Define $pp = (g, g_2, \vec{u}, \Omega = g^{\omega}, A = e(g,g)^{\alpha})$ $mk = (\omega, g^{\alpha})$

Extract

- Choose $s_i \leftarrow \mathbb{Z}_p$
- Return $K_i = ((g^{\alpha})^{\frac{1}{\omega + s_i}}, g^{s_i}, g_2^{s_i})$

Sign

- Choose $r \leftarrow \mathbb{Z}_p$ let $\mathcal{F}(M) := u_0 \prod_{i=1}^{m} u_i^{M_i}$
- Return $S = (K_1, K_2, K_3 \cdot \mathcal{F}(M)^r, g^{-r}) \in \mathbb{G}^4$

Verify

$$e(S_1, S_2\Omega) \cdot A^{-1} \stackrel{?}{=} 1$$

$$e(S_2, g_2)^{-1} \cdot e(g, S_3) \cdot e(S_4, \mathcal{F}(M)) \stackrel{?}{=} 1$$

Boyen Waters Group Signatures [PKC'07] I

Hybrid Scheme

Setup

- Choose bilinear group $\mathbf{G}$
- Choose parameters for Waters signatures $(g, g_2, \vec{u}) \in \mathbb{G}^{m+3}$
- Choose $\omega, \alpha \leftarrow \mathbb{Z}_p$,
- Define $pp = (g, g_2, \vec{u}, \Omega = g^{\omega}, A = e(g,g)^{\alpha})$ $mk = (\omega, g^{\alpha})$

Extract

- Choose $s_i \leftarrow \mathbb{Z}_p$
- Return $K_i = ((g^{\alpha})^{\frac{1}{\omega + s_i}}, g^{s_i}, g_2^{s_i})$

Sign

- Choose $r \leftarrow \mathbb{Z}_p$ let $\mathcal{F}(M) := u_0 \prod_{i=1}^{m} u_i^{M_i}$
- Return $S = (K_1, K_2, K_3 \cdot \mathcal{F}(M)^r, g^{-r}) \in \mathbb{G}^4$

Verify

$$e(S_1, S_2\Omega) \cdot A^{-1} \stackrel{?}{=} 1$$

$$e(S_2, g_2)^{-1} \cdot e(g, S_3) \cdot e(S_4, \mathcal{F}(M)) \stackrel{?}{=} 1$$

Boyen Waters Group Signatures [PKC'07] II

Group Signatures

Setup
- Choose a bilinear group $\mathbf{G}$ of order $n = pq$
- Choose keys pp and mk for hybrid scheme
- Publish an additional element $h \in \mathbb{G}_q$
- Set the tracing key as $tk = q \in \mathbb{Z}$

Enroll
- Choose $s_i \leftarrow \mathbb{Z}_p$, s.t. $\omega + s_i \in \mathbb{Z}_n^\times$
- Return $K_i = ((g^\alpha)^{\frac{1}{\omega+s_i}}, g^{s_i}, g_2^{s_i})$

Sign
- Make hybrid signature (S_1, S_2, S_3, S_4)
- BGN-encrypt components: $S_i' := S_i \cdot h^{\rho_i}$ for $\rho_i \leftarrow \mathbb{Z}_n$
- Add proofs π_1, π_2 for each verification relation

Verify

$$e(S_1', S_2'\Omega) \cdot A^{-1} \stackrel{?}{=} e(h, \pi_1)$$

$$e(S_2', g_2)^{-1} \cdot e(g, S_3') \cdot e(S_4', \mathcal{F}(M)) \stackrel{?}{=} e(h, \pi_2)$$

Trace for each s_i, check $(S_2')^q \stackrel{?}{=} (g^{s_i})^q$

Boyen Waters Group Signatures [PKC'07] III

Security of Group Signature

Anonymity.

- Replace h by a random element in $\mathbb{G}$ (indist. by SD).
- Lemma: If $h \leftarrow \mathbb{G}$ then signer id is statistically hidden, i.e. π_1 and π_2 do not leak information.

"Full Traceability". Reduction to unforgeability of hybrid signatures in $\mathbb{G}_p$

- Get parameters, build twin scheme in $\mathbb{G}_q$.
- Parameters of group signature are products of parameters of schemes in $\mathbb{G}_p$ and $\mathbb{G}_q$.
- Forgery (S_1', S_2', S_3', S_4') is projected to $\mathbb{G}_p$ by raising elements to the power of θ with $\theta \equiv 1 \pmod{p}$ and $\theta \equiv 0 \pmod{q}$.

Boyen Waters Group Signatures [PKC'07] III

Security of Group Signature

Anonymity.

- Replace h by a random element in $\mathbb{G}$ (indist. by SD).
- Lemma: If $h \leftarrow \mathbb{G}$ then signer id is statistically hidden, i.e. π_1 and π_2 do not leak information.

"Full Traceability". Reduction to unforgeability of hybrid signatures in $\mathbb{G}_p$

- Get parameters, build twin scheme in $\mathbb{G}_q$.
- Parameters of group signature are products of parameters of schemes in $\mathbb{G}_p$ and $\mathbb{G}_q$.
- Forgery (S_1', S_2', S_3', S_4') is projected to $\mathbb{G}_p$ by raising elements to the power of θ with $\theta \equiv 1 \pmod{p}$ and $\theta \equiv 0 \pmod{q}$.

The Leak-Tightness Lemma I

Let $(n, \mathbb{G}, \mathbb{G}_T, e, g)$ be a bilinear group, and let $a_j, b_j \in \mathbb{G}$, $\delta_{j,i}, \varepsilon_{j,i} \in \mathbb{Z}_n$ for $1 \leq j \leq \ell$, $1 \leq i \leq m$. Let $(X_i)_{i=1}^m \in \mathbb{G}^m$ satisfy a **pairing-product equation** $E_{(a_j,b_j)_j}$ that is

$$E_{(a_j,b_j)_j}(X_1, \ldots, X_m): \quad \prod_{j=1}^{\ell} e\left(a_j \prod_{i=1}^{m} X_i^{\delta_{j,i}},\ b_j \prod_{i=1}^{m} X_i^{\varepsilon_{j,i}}\right) = 1\ .$$

The Leak-Tightness Lemma I

Let $(n, \mathbb{G}, \mathbb{G}_T, e, g)$ be a bilinear group, and let $a_j, b_j \in \mathbb{G}$, $\delta_{j,i}, \varepsilon_{j,i} \in \mathbb{Z}_n$ for $1 \leq j \leq \ell$, $1 \leq i \leq m$. Let $(X_i)_{i=1}^m \in \mathbb{G}^m$ satisfy a **pairing-product equation** $E_{(a_j,b_j)_j}$ that is

$$E_{(a_j,b_j)_j}(X_1, \ldots, X_m): \quad \prod_{j=1}^{\ell} e\left(a_j \prod_{i=1}^m X_i^{\delta_{j,i}},\ b_j \prod_{i=1}^m X_i^{\varepsilon_{j,i}}\right) = 1\ .$$

1. Form of a proof

The Leak-Tightness Lemma I

Let $(n, \mathbb{G}, \mathbb{G}_T, e, g)$ be a bilinear group, and let $a_j, b_j \in \mathbb{G}$, $\delta_{j,i}, \varepsilon_{j,i} \in \mathbb{Z}_n$ for $1 \leq j \leq \ell$, $1 \leq i \leq m$. Let $(X_i)_{i=1}^m \in \mathbb{G}^m$ satisfy a **pairing-product equation** $E_{(a_j,b_j)_j}$ that is

$$E_{(a_j,b_j)_j}(X_1, \ldots, X_m): \quad \prod_{j=1}^{\ell} e\Big(a_j \prod_{i=1}^m X_i^{\delta_{j,i}},\ b_j \prod_{i=1}^m X_i^{\varepsilon_{j,i}}\Big) = 1\ .$$

1. Let $H \in \mathbb{G}$, $(\rho_i)_{i=1}^m \in {\mathbb{Z}_n}^m$. Then $\widetilde{X}_i := X_i H^{\rho_i}$ for $1 \leq i \leq m$ satisfy

$$\prod_j e\Big(a_j \prod_i \widetilde{X}_i^{\delta_{j,i}},\ b_j \prod_i \widetilde{X}_i^{\varepsilon_{j,i}}\Big) = e\Big(H,\ P_E\big((X_i),(\rho_i)\big)\Big)\ , \qquad (\tilde{E})$$

where $P_E\big((X_i),(\rho_i)\big) :=$

$$\prod_j \Big((a_j \prod_i X_i^{\delta_{j,i}})^{\sum \varepsilon_{j,i}\rho_i} (b_j \prod_i X_i^{\varepsilon_{j,i}})^{\sum \delta_{j,i}\rho_i} H^{(\sum \delta_{j,i}\rho_i)(\sum \varepsilon_{j,i}\rho_i)}\Big)\ .$$

The Leak-Tightness Lemma II

2. Proofs do not leak info on plaintexts

The Leak-Tightness Lemma II

2. Given (X_i) and (X'_i) both satisfying E, and (ρ_i), (ρ'_i), s.t. for all $1 \leq i \leq m$: $X_i H^{\rho_i} = X'_i H^{\rho'_i}$, then

$$P_E\big((X_i),(\rho_i)\big) = P_E\big((X'_i),(\rho'_i)\big) \ .$$

The Leak-Tightness Lemma II

2. Given (X_i) and (X'_i) both satisfying E, and (ρ_i), (ρ'_i), s.t. for all $1 \leq i \leq m$: $X_i H^{\rho_i} = X'_i H^{\rho'_i}$, then

$$P_E\big((X_i),(\rho_i)\big) = P_E\big((X'_i),(\rho'_i)\big) \ .$$

3. Simulation

The Leak-Tightness Lemma II

2. Given (X_i) and (X'_i) both satisfying E, and (ρ_i), (ρ'_i), s.t. for all $1 \leq i \leq m$: $X_i H^{\rho_i} = X'_i H^{\rho'_i}$, then

$$P_E((X_i),(\rho_i)) = P_E((X'_i),(\rho'_i)) \ .$$

3. Let $|G| = pq$, let $a_j, b_j, X_i \in \mathbb{G}_p$; $c_j, d_j, Y_i \in \mathbb{G}_q$ for all i, j. If (X_i) satisfy $E_{(a_j,b_j)_j}$ and (Y_i) satisfy $E_{(c_j,d_j)_j}$, then $(X_i Y_i)$ satisfy $E_{(a_j c_j, b_j d_j)_j}$.

The Leak-Tightness Lemma II

2. Given (X_i) and (X'_i) both satisfying E, and (ρ_i), (ρ'_i), s.t. for all $1 \leq i \leq m$: $X_i H^{\rho_i} = X'_i H^{\rho'_i}$, then

$$P_E\big((X_i),(\rho_i)\big) = P_E\big((X'_i),(\rho'_i)\big) \ .$$

3. Let $|G| = pq$, let $a_j, b_j, X_i \in \mathbb{G}_p$; $c_j, d_j, Y_i \in \mathbb{G}_q$ for all i, j. If (X_i) satisfy $E_{(a_j,b_j)_j}$ and (Y_i) satisfy $E_{(c_j,d_j)_j}$, then $(X_i Y_i)$ satisfy $E_{(a_j c_j, b_j d_j)_j}$.

4. Projection

The Leak-Tightness Lemma II

2. Given (X_i) and (X'_i) both satisfying E, and (ρ_i), (ρ'_i), s.t. for all $1 \leq i \leq m$: $X_i H^{\rho_i} = X'_i H^{\rho'_i}$, then

$$P_E\big((X_i),(\rho_i)\big) = P_E\big((X'_i),(\rho'_i)\big) \ .$$

3. Let $|G| = pq$, let $a_j, b_j, X_i \in \mathbb{G}_p$; $c_j, d_j, Y_i \in \mathbb{G}_q$ for all i, j. If (X_i) satisfy $E_{(a_j,b_j)_j}$ and (Y_i) satisfy $E_{(c_j,d_j)_j}$, then $(X_i Y_i)$ satisfy $E_{(a_j c_j, b_j d_j)_j}$.

4. Let furthermore $H \in \mathbb{G}_q$ and $\theta \in \mathbb{N}$ be such that $\theta \equiv 1 \pmod{p}$ and $\theta \equiv 0 \pmod{q}$. If $(\widetilde{X}_i) \in \mathbb{G}$ satisfy $\tilde{E}_{(a_j c_j, b_j d_j)_j}$ for some P_E, then $(\widetilde{X}_i^{\theta})$ satisfy $E_{(a_j,b_j)_j}$.

Re-randomization

$$(X_i) \text{ satisfy } \quad \prod_j e\left(a_j \prod_i X_i^{\delta_{j,i}},\ b_j \prod_i X_i^{\varepsilon_{j,i}}\right) = 1$$

$$(\widetilde{X}_i) \text{ satisfy } \quad \prod_j e\left(a_j \prod_i \widetilde{X}_i^{\delta_{j,i}},\ b_j \prod_i \widetilde{X}_i^{\varepsilon_{j,i}}\right) = e\left(H,\ P_E\left((X_i),(\rho_i)\right)\right)$$

Re-randomization

$$(X_i) \text{ satisfy} \quad \prod_j e\left(a_j \prod_i X_i^{\delta_{j,i}},\ b_j \prod_i X_i^{\varepsilon_{j,i}}\right) = e(H, P')$$

$$(\widetilde{X}_i) \text{ satisfy} \quad \prod_j e\left(a_j \prod_i \widetilde{X}_i^{\delta_{j,i}},\ b_j \prod_i \widetilde{X}_i^{\varepsilon_{j,i}}\right) = e\left(H,\ P' \cdot P_E\left((X_i), (\rho_i)\right)\right)$$

Re-randomization

$$(X_i) \text{ satisfy} \quad \prod_j e\left(a_j \prod_i X_i^{\delta_{j,i}},\ b_j \prod_i X_i^{\varepsilon_{j,i}}\right) = e(H, P')$$

$$(\widetilde{X}_i) \text{ satisfy} \quad \prod_j e\left(a_j \prod_i \widetilde{X}_i^{\delta_{j,i}},\ b_j \prod_i \widetilde{X}_i^{\varepsilon_{j,i}}\right) = e\left(H,\ P' \cdot P_E((X_i), (\rho_i))\right)$$

So, given a proof P for $(\widetilde{X}_i)$ satisfying $\widetilde{E}$, one can re-randomize the $(\widetilde{X}_i)$:

$$\widetilde{\widetilde{X}}_i := \widetilde{X}_i H^{\rho'_i} \qquad \text{with } \rho'_i \leftarrow \mathbb{Z}_p$$

and adapt the proof (without knowledge of the plaintexts!):

$$P_{\text{new}} := P \cdot P_E((\widetilde{X}), (\rho'_i))$$

If $((\widetilde{X}_i), P)$ and $((\widetilde{Y}_i), P')$ both satisfy $\widetilde{E}$, then their re-randomizations are indistinguishable by SD and the Lemma.

Application of Re-randomization

Intermediate Delegator Anonymity

Even the delegatee cannot distinguish warrants from different previous delegators.

Compatible Signature Scheme

A signature scheme with the following properties:

- EUF-CMA secure
- verification keys lie in message space
- messages and signatures are $\mathbb{G}$-elements
- verification by checking pairing-product equations.
- (efficient?)

Compatible Signature Scheme

A signature scheme with the following properties:

- EUF-CMA secure
- verification keys lie in message space
- messages and signatures are $\mathbb{G}$-elements
- verification by checking pairing-product equations.
- (efficient?)

This paper: not quite efficient, based on new kind of assumption.
Efficient implementation:

F: Automorphic Signatures in Bilinear Groups
`http://eprint.iacr.org/2009/320`

Messages and public keys in $\mathbb{G}^2$. Signatures in $\mathbb{G}^5$
Verification: 7 pairing evaluations.
q-SDH-style Assumption

Thank you!