Commuting signatures and verifiable encryption

Georg Fuchsbauer

University of Bristol

EUROCRYPT 17.05.2011

Results

New primitive: Commuting signatures (and verifiable encryption)

- New functionality
- Efficient instantiation in pairing groups

Results

New primitive: Commuting signatures (and verifiable encryption)

- New functionality
- Efficient instantiation in pairing groups

Application: Delegatable anonymous credentials

- *Non-interactive* delegation
- Significant efficiency improvements

Results

New primitive: Commuting signatures (and verifiable encryption)

- New functionality
- Efficient instantiation in pairing groups

Application: Delegatable anonymous credentials

- *Non-interactive* delegation
- Significant efficiency improvements

Other results: Groth-Sahai proofs

- Properties of proofs
- Stronger notion of simulation

Outline of this talk

1. **Commuting signatures**
2. **Delegatable anonymous credentials**
3. **Instantiating commuting signatures**

1. **Commuting signatures**
2. Delegatable anonymous credentials
3. Instantiating commuting signatures

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

Commuting signatures and verifiable encryption I

- Signature $\quad M \xrightarrow{sk} \Sigma \quad$ Verification: vk, M, Σ
- Randomizable Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} \longrightarrow \boxed{M}, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

Proof adaptation:

$$\left.\begin{matrix} \widetilde{\pi} \\ \bar{\pi} \end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

Proof adaptation:

$$\left.\begin{matrix} \widetilde{\pi} \\ \bar{\pi} \end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$$

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \Sigma$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

Proof adaptation:

$$\left.\begin{matrix} \widetilde{\pi} \\ \bar{\pi} \end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$$

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \Sigma$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$vk, M, \Sigma \longrightarrow$

$\longrightarrow$	$\boxed{\Sigma}, \widetilde{\pi}$	Verification: $vk, M, \boxed{\Sigma}, \widetilde{\pi}$
$\longrightarrow$	$\boxed{M}, \bar{\pi}$	Verification: $vk, \boxed{M}, \Sigma, \bar{\pi}$
$\longrightarrow$	$\boxed{M}, \boxed{\Sigma}, \pi$	Verification: $vk, \boxed{M}, \boxed{\Sigma}, \pi$
$\longrightarrow$	$\boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi}$	Verification: $\boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi}$

- Commuting signature and verifiable encryption

Proof adaptation: $\left.\begin{matrix}\widetilde{\pi}\\ \bar{\pi}\end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$ Verification: $vk, \boxed{M}, \boxed{\Sigma}, \pi$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

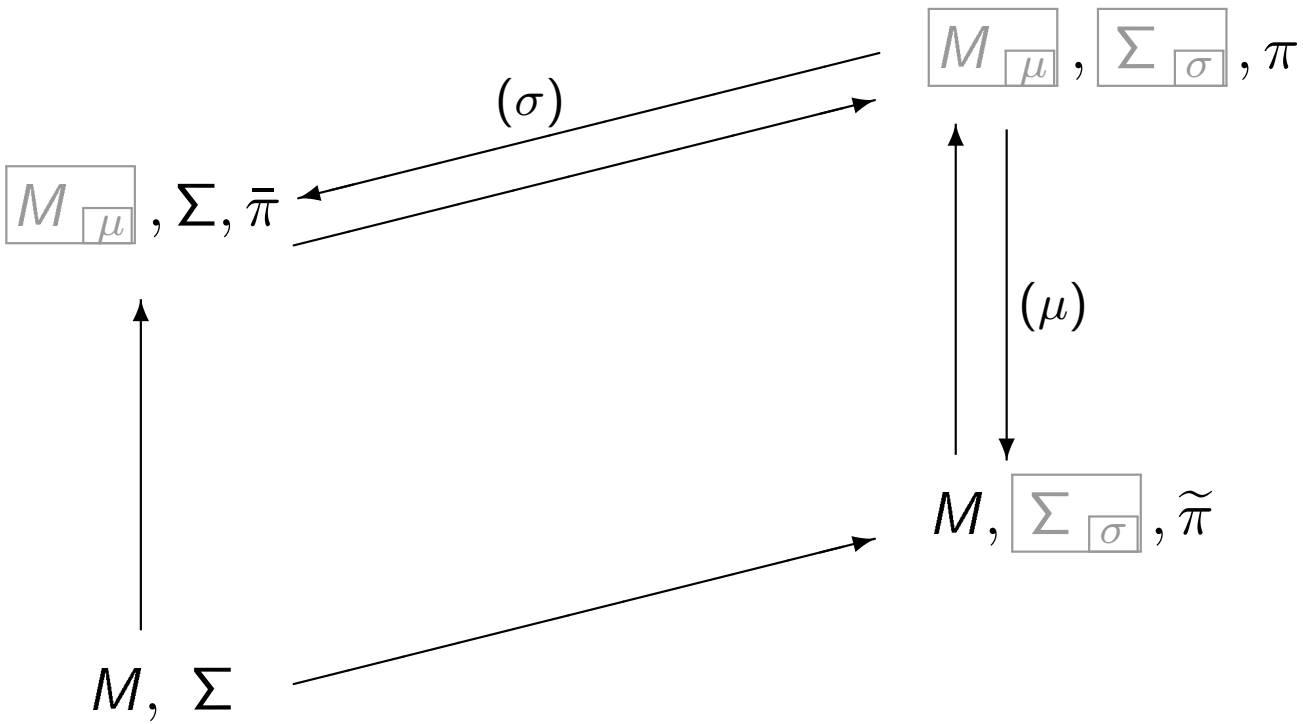
$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

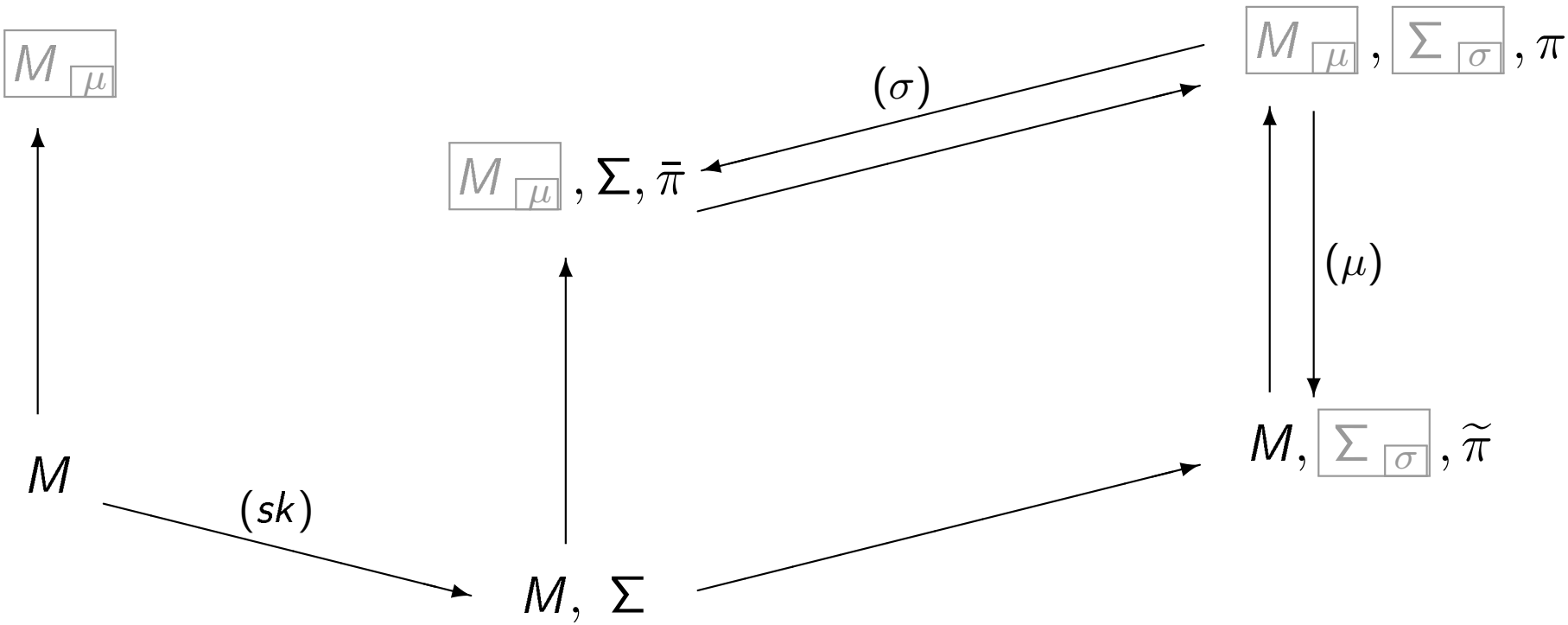
Sign plaintext then encrypt $\iff$ encrypt then sign plaintext

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$ Verification: $vk, \boxed{M}, \boxed{\Sigma}, \pi$

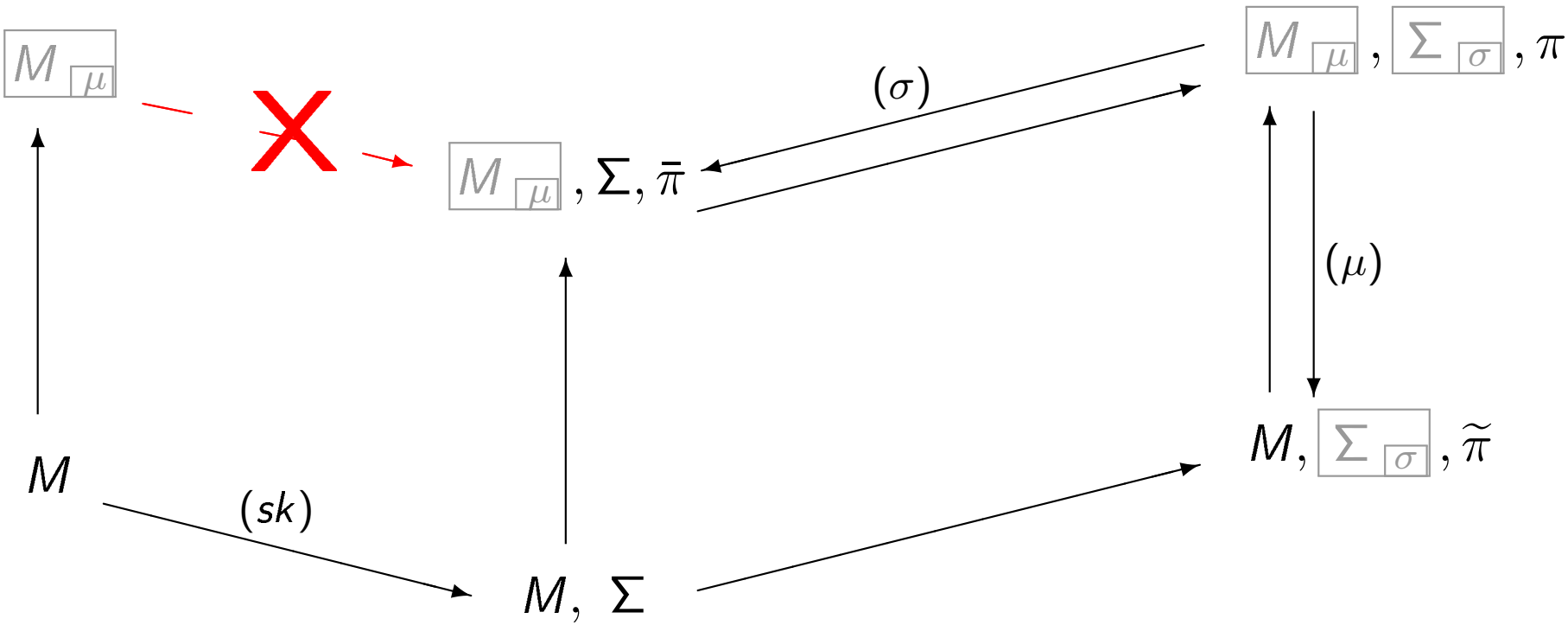
Commuting signatures and verifiable encryption II



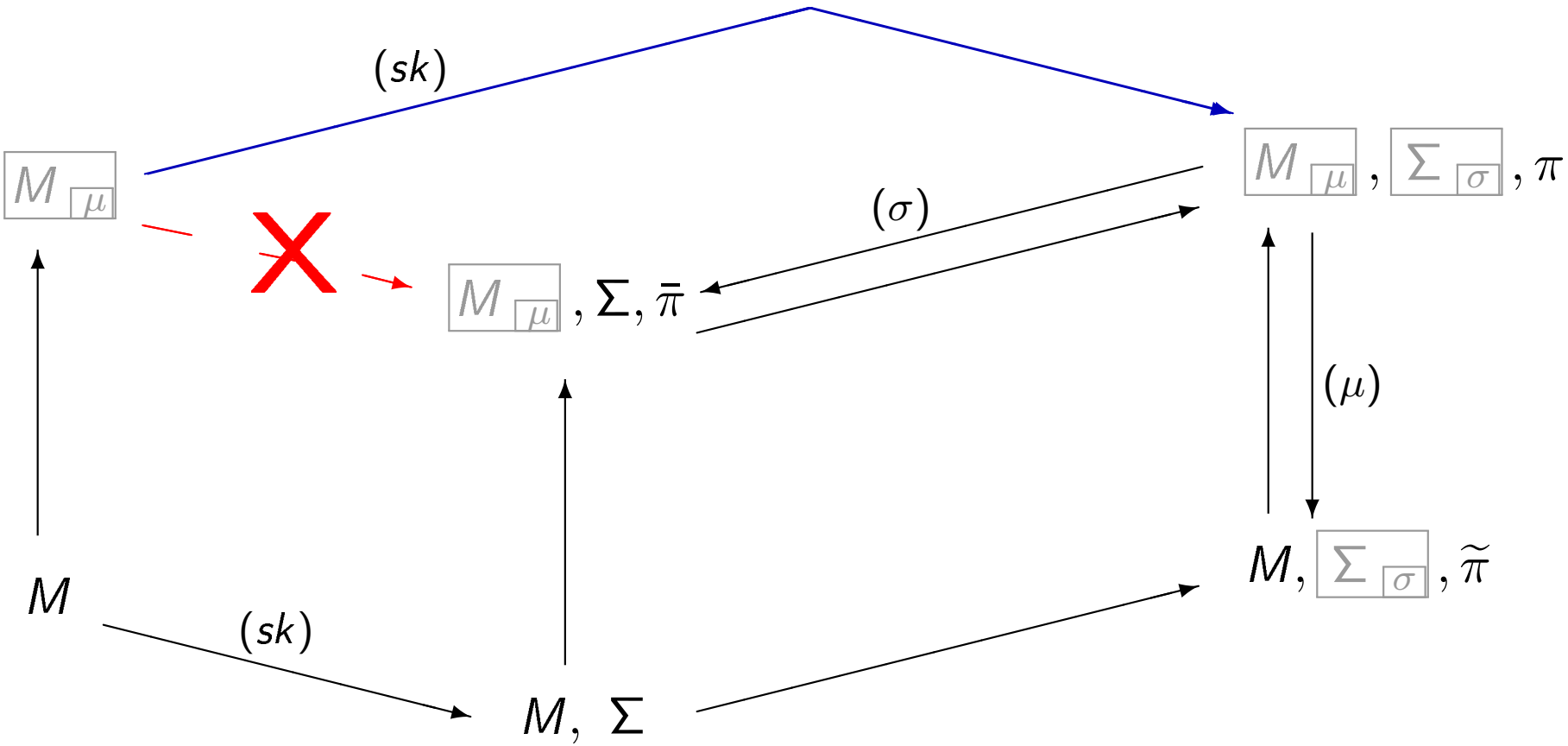
Commuting signatures and verifiable encryption II



Commuting signatures and verifiable encryption II



Commuting signatures and verifiable encryption II



1. Commuting signatures

2. **Delegatable anonymous credentials**

3. Instantiating commuting signatures

Delegatable anonymous credentials [BCCKLS09]

- Users can *prove* to hold credential w/o revealing their identity
- Credentials can be issued/delegated and obtained anonymously

Delegatable anonymous credentials [BCCKLS09]

- Users can *prove* to hold credential w/o revealing their identity
- Credentials can be issued/delegated and obtained anonymously

Model

- Each user holds a secret key and can
- . . . produce arbitrarily many (unlinkable) pseudonyms from it

Delegatable anonymous credentials [BCCKLS09]

- Users can *prove* to hold credential w/o revealing their identity
- Credentials can be issued/delegated and obtained anonymously

Model

- Each user holds a secret key and can
- . . . produce arbitrarily many (unlinkable) pseudonyms from it
- . . . can publish pseudonym as *public key* for a credential
- . . . run *interactive protocol* to issue/delegate credentials to other users

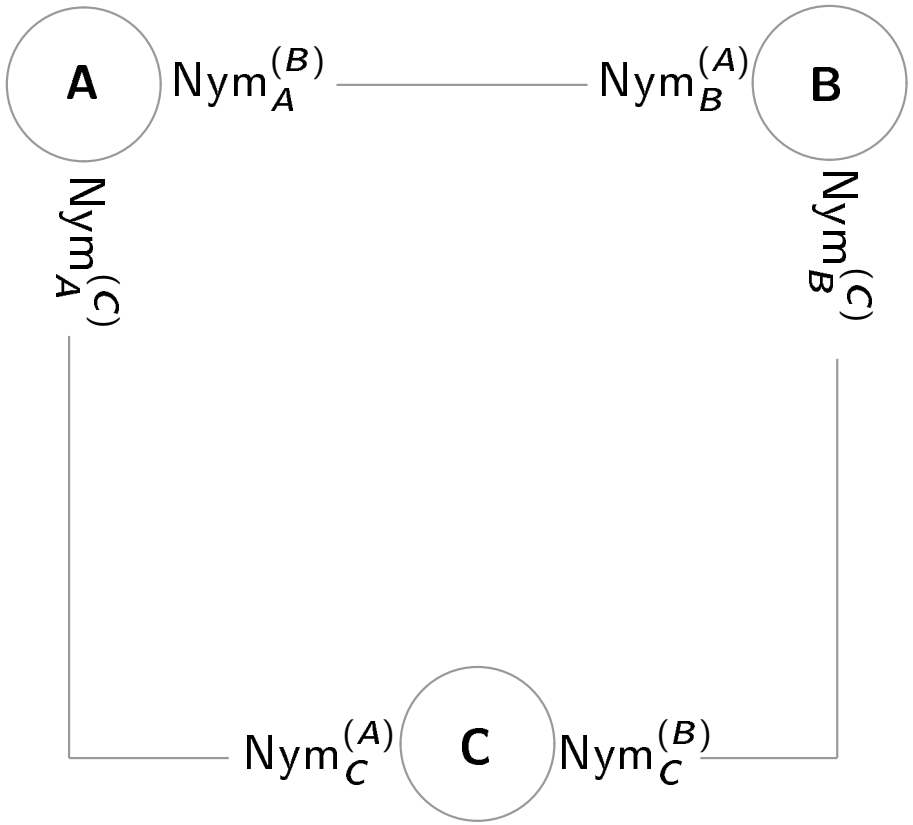
Delegatable anonymous credentials [BCCKLS09]

- Users can *prove* to hold credential w/o revealing their identity
- Credentials can be issued/delegated and obtained anonymously

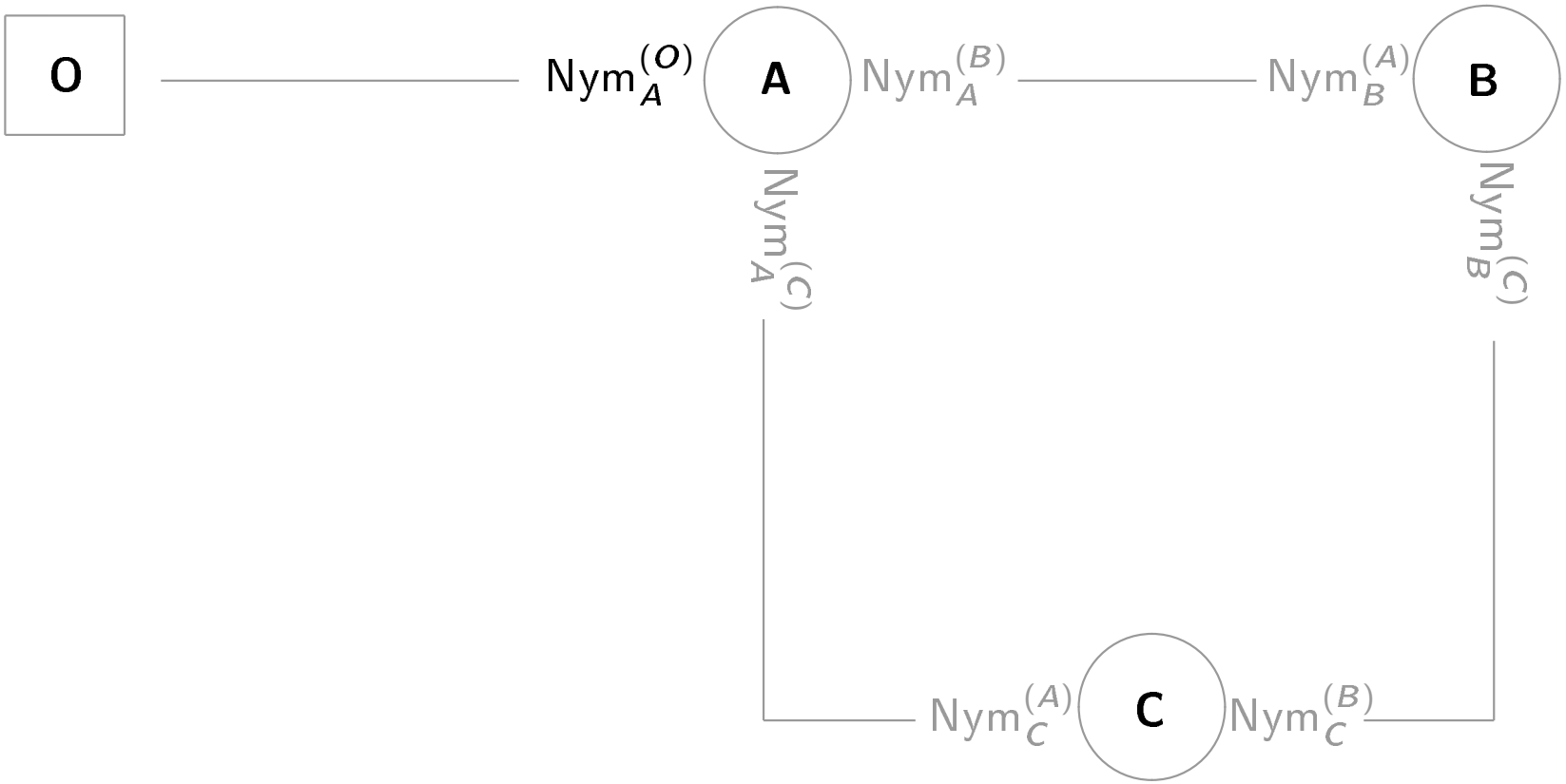
Model

- Each user holds a secret key and can
- . . . produce arbitrarily many (unlinkable) pseudonyms from it
- . . . can publish pseudonym as *public key* for a credential
- . . . run *interactive protocol* to issue/delegate credentials to other users
- . . . prove to hold credentials for every pseudonym

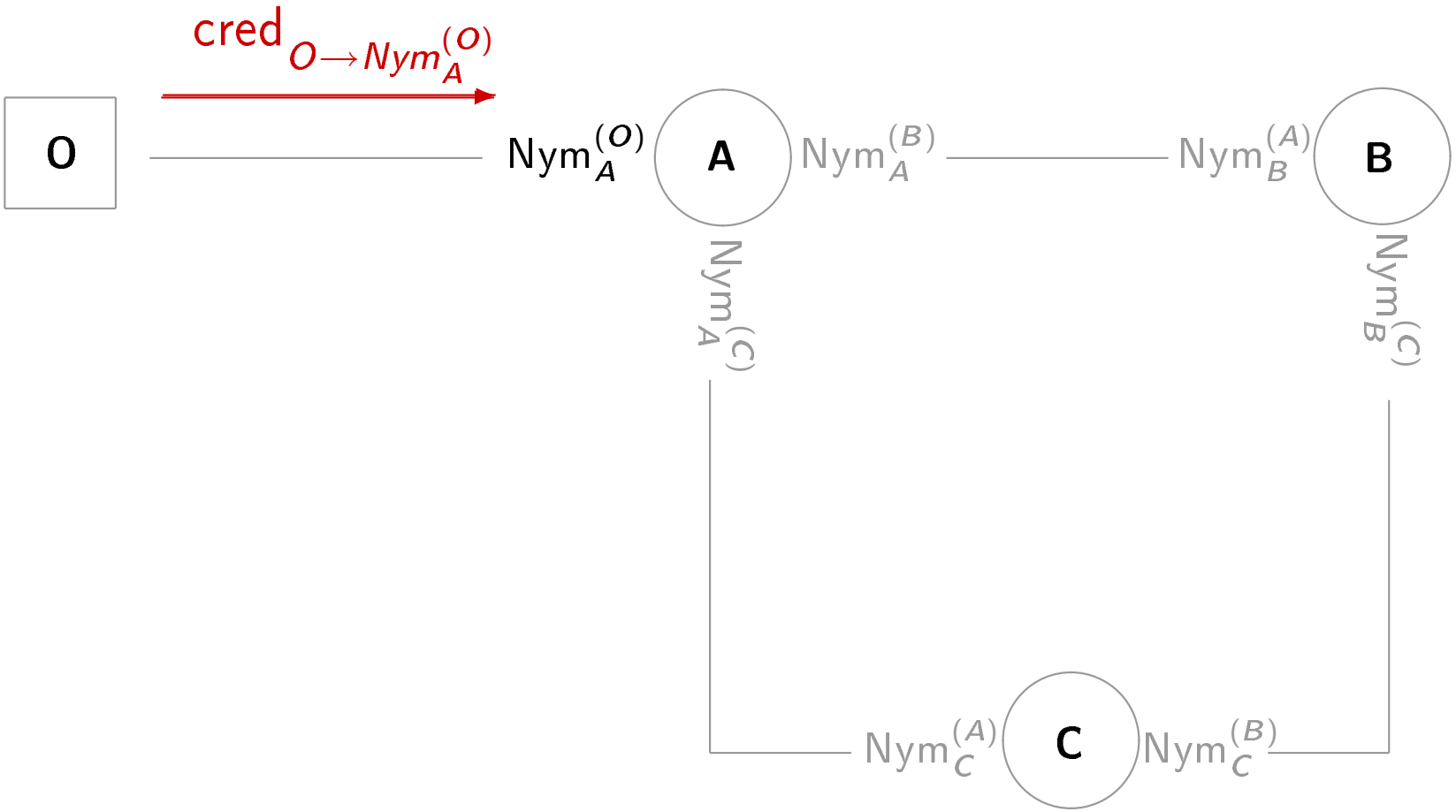
Non-interactively delegatable anonymous credentials



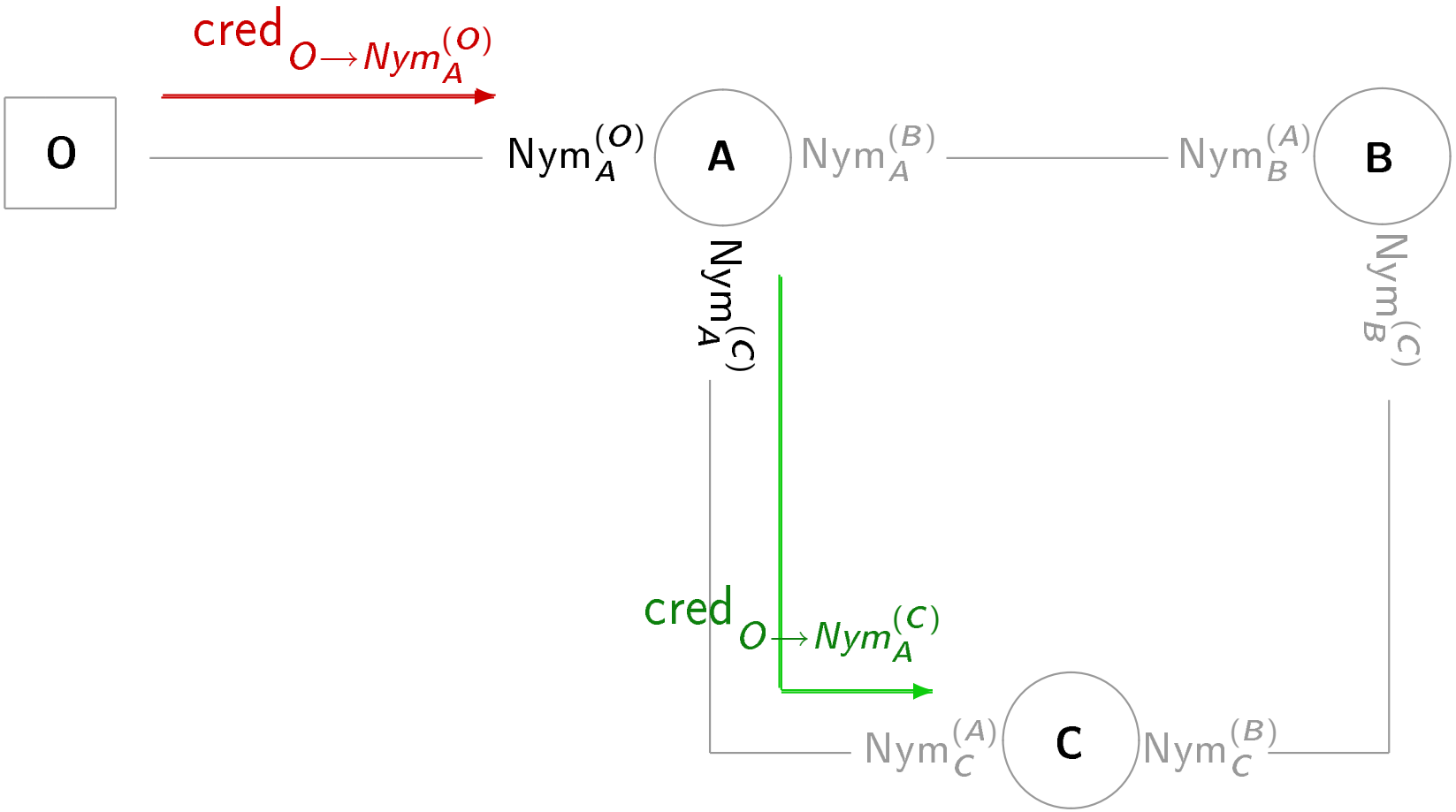
Non-interactively delegatable anonymous credentials



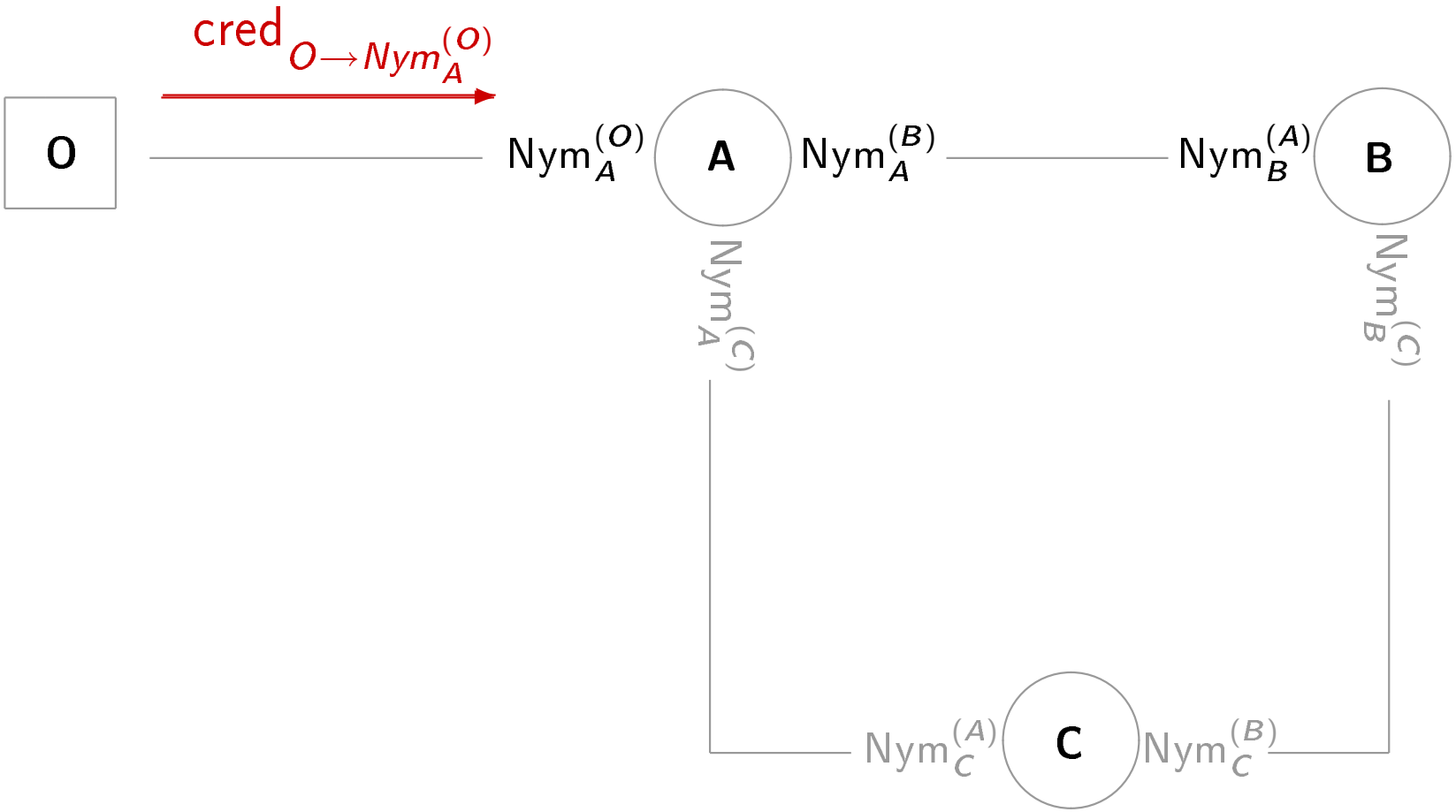
Non-interactively delegatable anonymous credentials



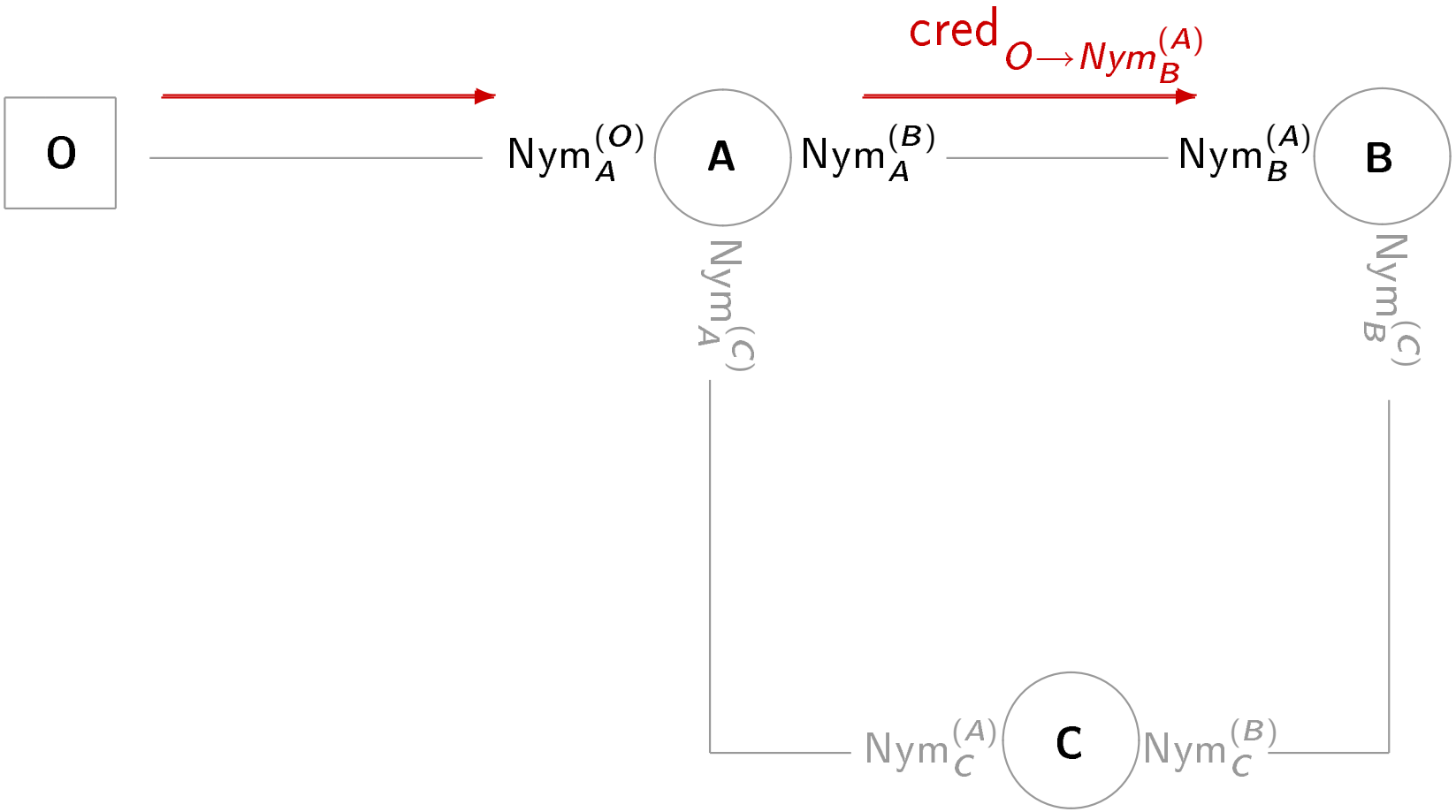
Non-interactively delegatable anonymous credentials



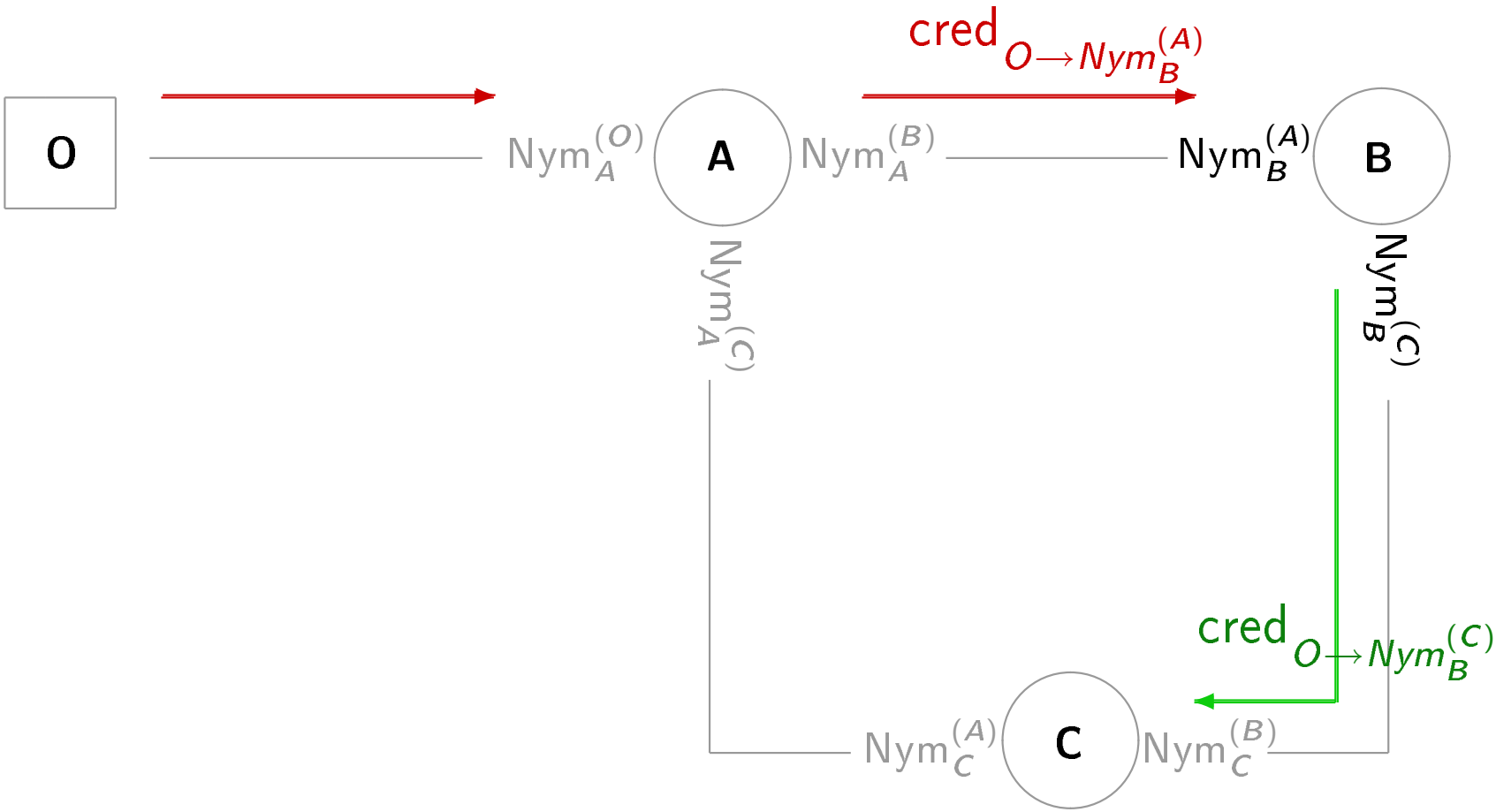
Non-interactively delegatable anonymous credentials



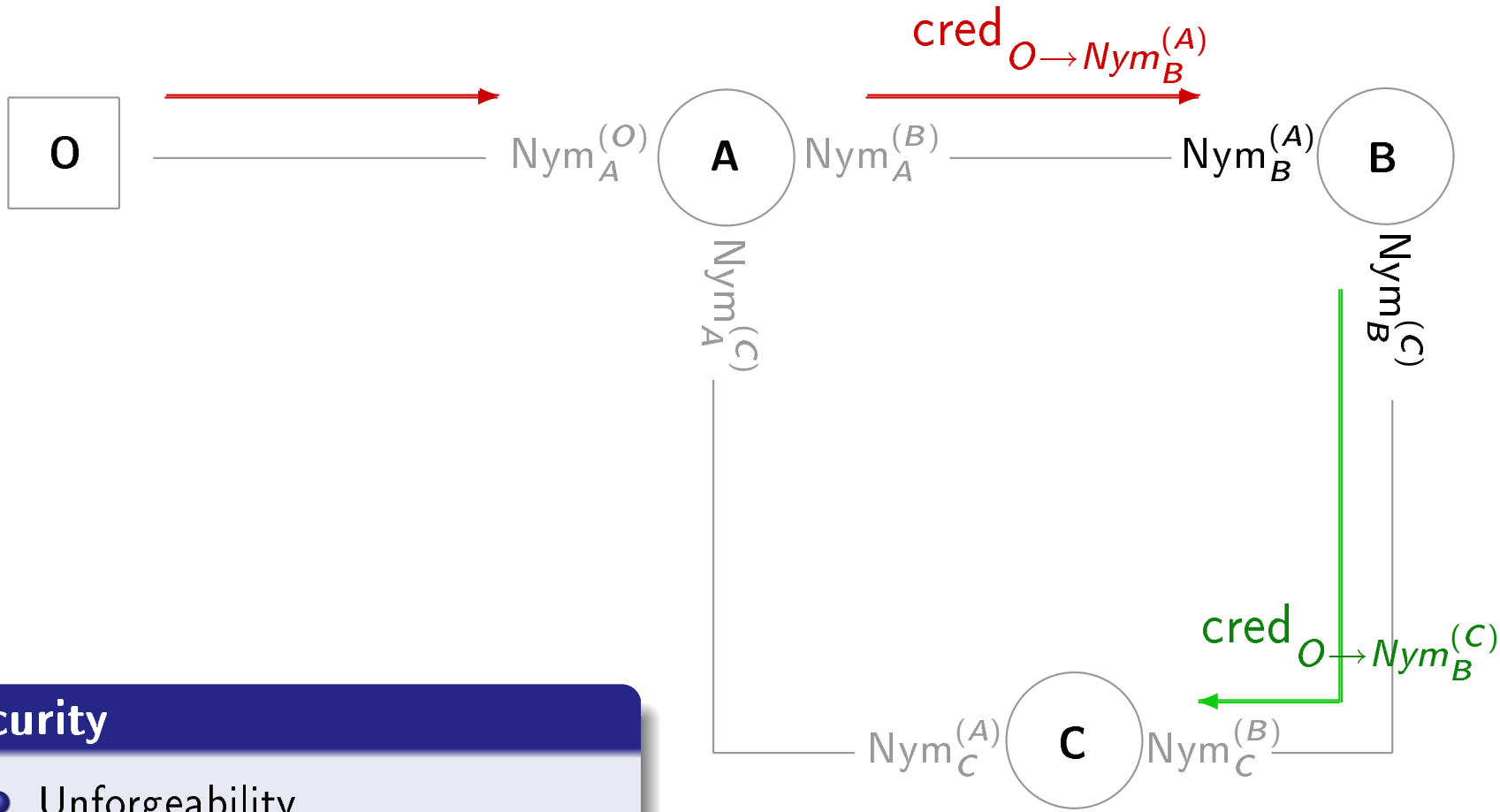
Non-interactively delegatable anonymous credentials



Non-interactively delegatable anonymous credentials



Non-interactively delegatable anonymous credentials



Security

- Unforgeability
- Anonymity (simulation-based)

Black-box instantiation of NIDAC

In a nutshell

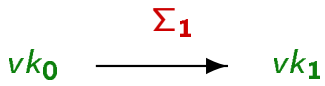
- Pseudonym: encryption of user verification key
- Credential: verifiably encrypted signature
- Non-interactive delegation: commuting signature

Black-box instantiation of NIDAC

- Delegation of signing rights

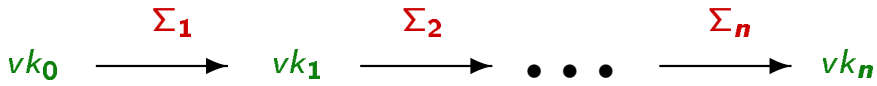
Black-box instantiation of NIDAC

- Delegation of signing rights



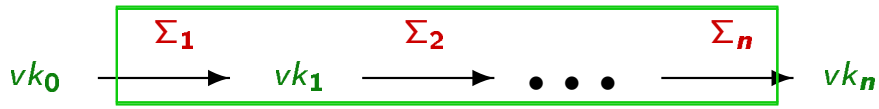
Black-box instantiation of NIDAC

- Delegation of signing rights



Black-box instantiation of NIDAC

- Delegation of signing rights



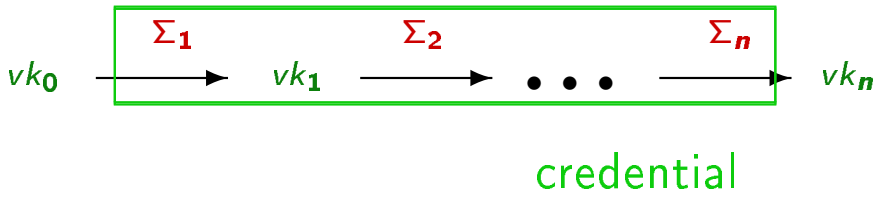
Signatures

credential

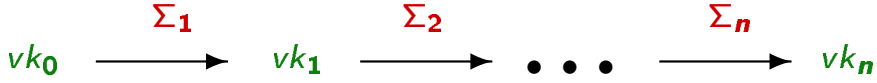
Black-box instantiation of NIDAC

- Delegation of signing rights

  Signatures



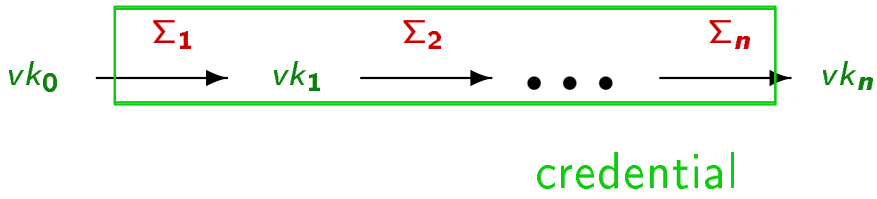
- Anonymous show



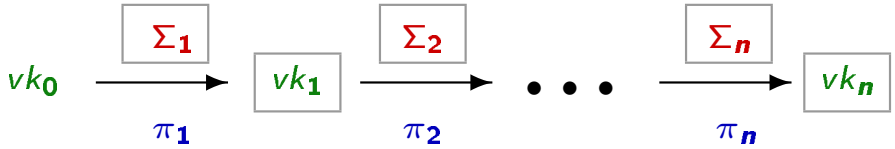
Black-box instantiation of NIDAC

- Delegation of signing rights

  Signatures



- Anonymous show



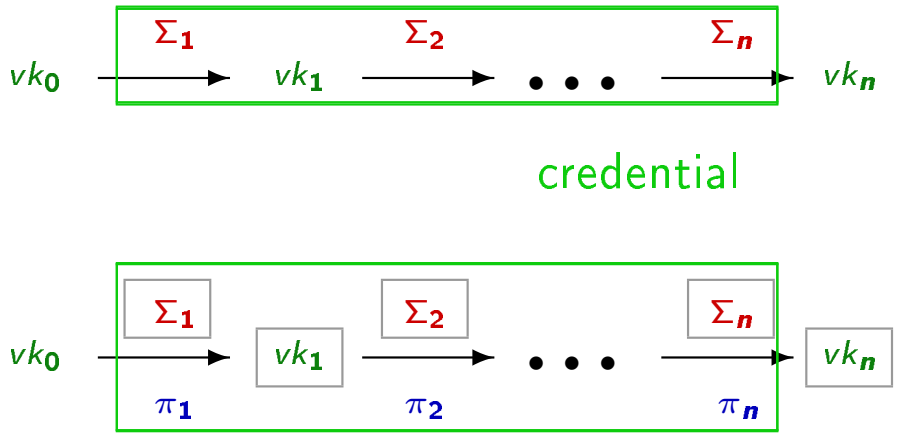
Black-box instantiation of NIDAC

- Delegation of signing rights

  Signatures

- Anonymous show

  Verifiable encryption



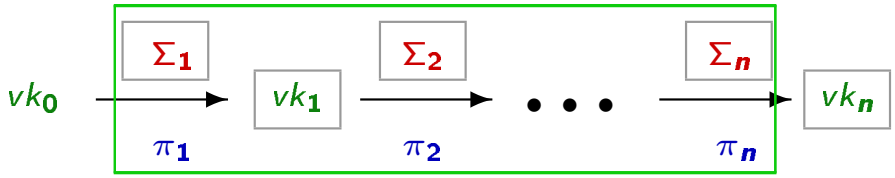
Black-box instantiation of NIDAC

- Delegation of signing rights

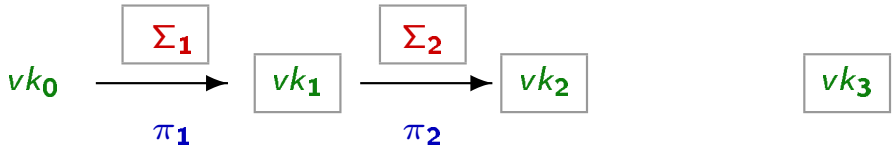
  Signatures

- Anonymous show

  Verifiable encryption



- Anonymous delegation



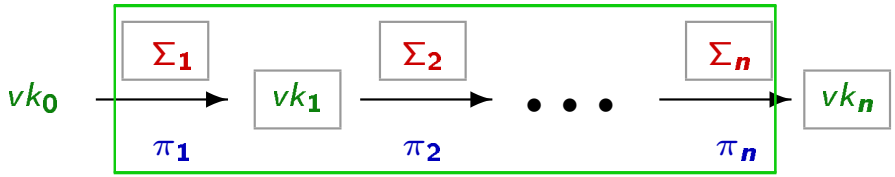
Black-box instantiation of NIDAC

- Delegation of signing rights

Signatures

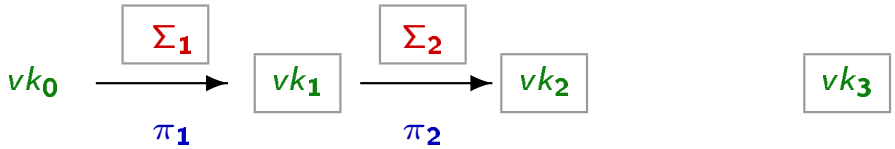
credential

- Anonymous show



Verifiable encryption

- Anonymous delegation



Commuting signatures

- Sign encrypted value $\boxed{vk_3}$ $\Rightarrow (vk_2, \boxed{\Sigma_3}, \boxed{vk_3}, \pi'_3)$

Black-box instantiation of NIDAC

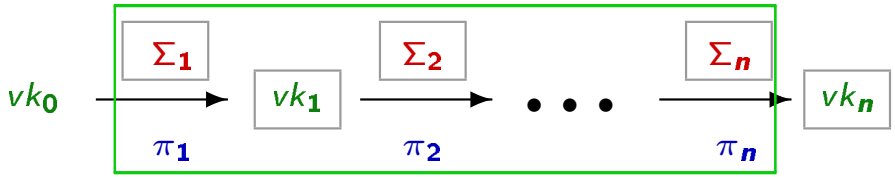
- Delegation of signing rights

  Signatures

credential

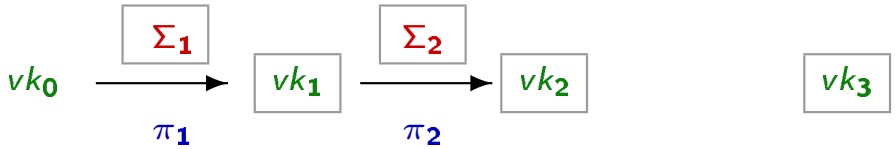
- Anonymous show

  Verifiable encryption



- Anonymous delegation

  Commuting signatures



  - Sign encrypted value $\boxed{vk_3}$ $\Rightarrow (\ vk_2, \boxed{\Sigma_3}, \boxed{vk_3}, \pi'_3\ )$
  - Adapt proof for $\boxed{vk_2}$ $\Rightarrow (\boxed{vk_2}, \boxed{\Sigma_3}, \boxed{vk_3}, \pi_3\ )$

Black-box instantiation of NIDAC

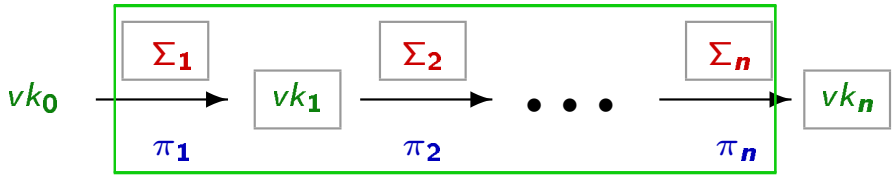
- Delegation of signing rights

  Signatures

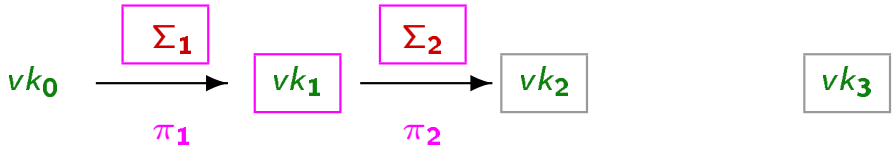
credential

- Anonymous show

  Verifiable encryption



- Anonymous delegation



  Commuting signatures

  - Sign encrypted value $\boxed{vk_3}$ $\Rightarrow (\ vk_2, \boxed{\Sigma_3}, \boxed{vk_3}, \pi'_3\ )$
  - Adapt proof for $\boxed{vk_2}$ $\Rightarrow (\boxed{vk_2}, \boxed{\Sigma_3}, \boxed{vk_3}, \pi_3\ )$
  - Randomize previous encryptions/proofs

Black-box instantiation of NIDAC

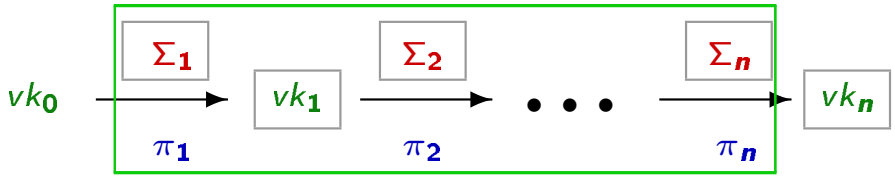
- Delegation of signing rights

  Signatures

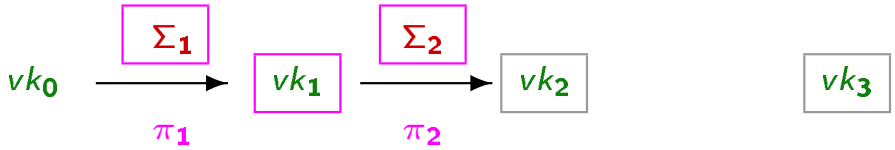
credential

- Anonymous show

  Verifiable encryption



- Anonymous delegation



  Commuting signatures

  - Sign encrypted value $\boxed{vk_3}$ $\Rightarrow (\ vk_2, \boxed{\Sigma_3}, \boxed{vk_3}, \pi'_3\ )$
  - Adapt proof for $\boxed{vk_2}$ $\Rightarrow (\boxed{vk_2}, \boxed{\Sigma_3}, \boxed{vk_3}, \pi_3\ )$
  - Randomize previous encryptions/proofs

Send credential $(\boxed{\Sigma_1}, \pi_1, \boxed{vk_1}, \boxed{\Sigma_2}, \pi_2, \boxed{vk_2}, \boxed{\Sigma_3}, \pi_3)$

1. Commuting signatures
2. Delegatable anonymous credentials
3. **Instantiating commuting signatures**

Building blocks

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

Pairing: $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ bilinear

Building blocks

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

Pairing: $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ bilinear

Pairing-product equation (PPE)

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

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

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

Building blocks

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

Pairing: $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ bilinear

Pairing-product equation (PPE)

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

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

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

Pairing-product equation (PPE)

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

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

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

Pairing-product equation (PPE)

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

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

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

Building blocks

Pairing-product equation (PPE)

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

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

defined 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 [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

Building blocks

Pairing-product equation (PPE)

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

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

defined 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 [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make commitments $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E

   without revealing anything else

Building blocks

Pairing-product equation (PPE)

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

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

defined 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 [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make commitments $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E
   without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Building blocks

Pairing-product equation (PPE)

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

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

defined 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 [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E
   without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Building blocks

Pairing-product equation (PPE)

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

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

defined 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 [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Building blocks

Groth–Sahai proofs [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E
   without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Building blocks

Groth–Sahai proofs [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Building blocks

Groth–Sahai proofs [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E

   without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Automorphic signatures [AFGHO10]

- Messages and signatures are group elements
- Verification by pairing-product equation

Building blocks

Groth–Sahai proofs [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E

   without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Automorphic signatures [AFGHO10]

- Messages and signatures are group elements
- Verification by pairing-product equation

} "structure preserving"

Building blocks

Groth–Sahai proofs [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j
2. Construct proof π that committed values satisfy E

without revealing anything else

Given extraction key, one can extract the committed values X_i, Y_j

Automorphic signatures [AFGHO10]

- Messages and signatures are group elements
- Verification by pairing-product equation
- Verification keys lie in message space

} "structure preserving"

Building blocks

Groth–Sahai proofs [GS08]

Efficient non-interactive zero-knowledge (randomizable [BCCKLS09])

proof of knowledge of $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ satisfying E

1. Make encryptions $\mathbf{c}_i$ to X_i and $\mathbf{d}_j$ to Y_j

Groth-Sahai proofs + structure-pres. signatures
= verifiably encrypted signatures

Automorphic signatures [AFGHO10]

- Messages and signatures are group elements
- Verification by pairing-product equation
- Verification keys lie in message space

} "structure preserving"

Properties of Groth-Sahai proofs

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

Independence

Proofs do not depend on $\mathbf{t}$

Properties of Groth-Sahai proofs

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

Independence

Proofs do not depend on $\mathbf{t}$

Proofs for *linear* equations ($\gamma_{ij} = 0$) do not depend on encrypted values

Properties of Groth-Sahai proofs

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

Independence

Proofs do not depend on $\mathbf{t}$

Proofs for *linear* equations ($\gamma_{ij} = 0$) do not depend on encrypted values

Adapting

Proofs can be adapted when constants are turned into variables or vice versa

Properties of Groth-Sahai proofs

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

Independence

Proofs do not depend on $\mathbf{t}$

Proofs for *linear* equations ($\gamma_{ij} = 0$) do not depend on encrypted values

Adapting

Proofs can be adapted when constants are turned into variables or vice versa

Homomorphic

Let π be proof for E and ciphertexts $(\mathbf{c}_1, \ldots, \mathbf{c}_m, \mathbf{d}_1, \ldots, \mathbf{d}_n)$

Let π' be proof for E' and ciphertexts $(\mathbf{c}'_1, \ldots, \mathbf{c}'_{m'}, \mathbf{d}'_1, \ldots, \mathbf{d}'_{n'})$

Properties of Groth-Sahai proofs

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

Independence

Proofs do not depend on $\mathbf{t}$

Proofs for *linear* equations ($\gamma_{ij} = 0$) do not depend on encrypted values

Adapting

Proofs can be adapted when constants are turned into variables or vice versa

Homomorphic

Let π be proof for E and ciphertexts $(\mathbf{c}_1, \ldots, \mathbf{c}_m, \mathbf{d}_1, \ldots, \mathbf{d}_n)$

Let π' be proof for E' and ciphertexts $(\mathbf{c}'_1, \ldots, \mathbf{c}'_{m'}, \mathbf{d}'_1, \ldots, \mathbf{d}'_{n'})$

Then $\pi \cdot \pi'$ is a proof for $\mathsf{E} \cdot \mathsf{E}'$ and $(\mathbf{c}_1, \ldots, \mathbf{c}_m, \mathbf{c}'_1, \ldots, \mathbf{c}'_{m'}, \mathbf{d}_1, \ldots, \mathbf{d}'_{n'})$

$$\prod e(A_j, Y_j) \prod e(A'_j, Y'_j) \prod e(X_i, B_i) \prod e(X'_i, B'_i) \prod\prod e(X_i, Y_j)^{\gamma_{i,j}} \prod\prod e(X'_i, Y'_j)^{\gamma'_{i,j}} = \mathbf{t} \cdot \mathbf{t}' \qquad (\mathsf{E} \cdot \mathsf{E}')$$

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

Stronger notion of simulatability

- [GS08]: NIZK proof of satisfiability:
  Given E, simulator can produce $(\mathbf{c}_1, \ldots, \mathbf{c}_m, \mathbf{d}_1, \ldots, \mathbf{d}_n)$ and π

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

Stronger notion of simulatability

- [GS08]: NIZK proof of satisfiability:
  Given E, simulator can produce $(\mathbf{c}_1, \ldots, \mathbf{c}_m, \mathbf{d}_1, \ldots, \mathbf{d}_n)$ and π
- Now: Proof for given ciphertexts:
  Given E, $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$, simulator can produce $(\mathbf{d}_1, \ldots, \mathbf{d}_n)$ and π

Properties of Groth-Sahai proofs

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

Stronger notion of simulatability

- [GS08]: NIZK proof of satisfiability:
  Given E, simulator can produce $(\mathbf{c}_1, \ldots, \mathbf{c}_m, \mathbf{d}_1, \ldots, \mathbf{d}_n)$ and π
- Now: Proof for given ciphertexts:
  Given E, $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$, simulator can produce $(\mathbf{d}_1, \ldots, \mathbf{d}_n)$ and π

Application: Given pseudonyms of delegator and delegatee $\Rightarrow$ simulate credential

Instantiating commuting signatures

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)
- **User**, knowing R, turns Σ' into Σ on M

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)
- **User**, knowing R, turns Σ' into Σ on M

Blind signature:

Verif. encryption of Σ: $(M, \boxed{\Sigma}, \widetilde{\pi})$

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)
- **User**, knowing R, turns Σ' into Σ on M

Blind signature:

Verif. encryption of Σ: $(M, \boxed{\Sigma}, \widetilde{\pi})$

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

1. User could produce $(\boxed{M}, \boxed{\Sigma}, \pi)$

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)
- **User**, knowing R, turns Σ' into Σ on M

Blind signature:

Verif. encryption of Σ: $(M, \boxed{\Sigma}, \widetilde{\pi})$

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

1. User could produce $(\boxed{M}, \boxed{\Sigma}, \pi)$
2. Define:

$\overset{def}{=}$ encryption of M

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)
- **User**, knowing R, turns Σ' into Σ on M

Blind signature:

Verif. encryption of Σ: $(M, \boxed{\Sigma}, \widetilde{\pi})$

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

1. User could produce $(\boxed{M}, \boxed{\Sigma}, \pi)$
2. Define: $\widetilde{M} \stackrel{def}{=}$ encryption of M
3. $\widetilde{M} \longrightarrow \Sigma' \longrightarrow \boxed{\Sigma'}$

Instantiating commuting signatures

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)
- **User**, knowing R, turns Σ' into Σ on M

Blind signature:

Verif. encryption of Σ: $(M, \boxed{\Sigma}, \widetilde{\pi})$

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

2. Define:

$\overset{def}{=}$ encryption of M

3. $\widetilde{M} \longrightarrow \Sigma' \longrightarrow \boxed{\Sigma'}$

4. Encryptions homomorphic $\boxed{R}$ and $\boxed{\Sigma'} \longrightarrow \boxed{\Sigma}$

1. User could produce $(\boxed{M}, \boxed{\Sigma}, \pi)$

Instantiating commuting signatures

Round-optimal blind signature

Protocol to sign M from [AFGHO10]:

- **User** sends
  - *Randomization* $\widetilde{M}$ of M
  - *Encryptions* $\boxed{M}$, $\boxed{R}$
  - *Proof* of consistency τ
- **Signer** sends "pre-signature" Σ' (using $\widetilde{M}$)
- **User**, knowing R, turns Σ' into Σ on M

Blind signature:

Verif. encryption of Σ: $(M, \boxed{\Sigma}, \widetilde{\pi})$

Goal: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$

1. User could produce $(\boxed{M}, \boxed{\Sigma}, \pi)$
2. Define: $\stackrel{def}{=}$ encryption of M
3. $\widetilde{M} \longrightarrow \Sigma' \longrightarrow \boxed{\Sigma'}$
4. Encryptions homomorphic $\boxed{R}$ and $\boxed{\Sigma'} \longrightarrow \boxed{\Sigma}$
5. Properties of GS proofs $\tau \longrightarrow \pi$

Conclusion

Commuting signatures imply

- Verifiably encrypted signatures
- Blind signatures
- CL signatures and P-signatures

Conclusion

Commuting signatures imply

- Verifiably encrypted signatures
- Blind signatures
- CL signatures and P-signatures

Applications

- Delegatable anonymous credentials
  - First instantiation with non-interactive issuing/delegation
  - Efficiency improvements:
    - No complex 2-party computation
    - Size of credentials less than half

Conclusion

Commuting signatures imply

- Verifiably encrypted signatures
- Blind signatures
- CL signatures and P-signatures

Applications

- Delegatable anonymous credentials
  - First instantiation with non-interactive issuing/delegation
  - Efficiency improvements:
    - No complex 2-party computation
    - Size of credentials less than half
- Receipt-free e-voting [BFPV11]
- Fully anonymous transferable e-cash [BCFGST11]

Conclusion

Commuting signatures imply

- Verifiably encrypted signatures
- Blind signatures
- CL signatures and P-signatures

Applications

- Delegatable anonymous credentials
  - First instantiation with non-interactive issuing/delegation
  - Efficiency improvements:
    - No complex 2-party computation
    - Size of credentials less than half
- Receipt-free e-voting [BFPV11]
- Fully anonymous transferable e-cash [BCFGST11]

Updated full version in June: `eprint.iacr.org/2010/233`

Thank you! ☺