ACNS 2024: 22nd International Conference on Applied Cryptography and Network Security | Abu Dhabi, UAE

Identity-Based Matchmaking Encryption from Standard Lattice Assumptions

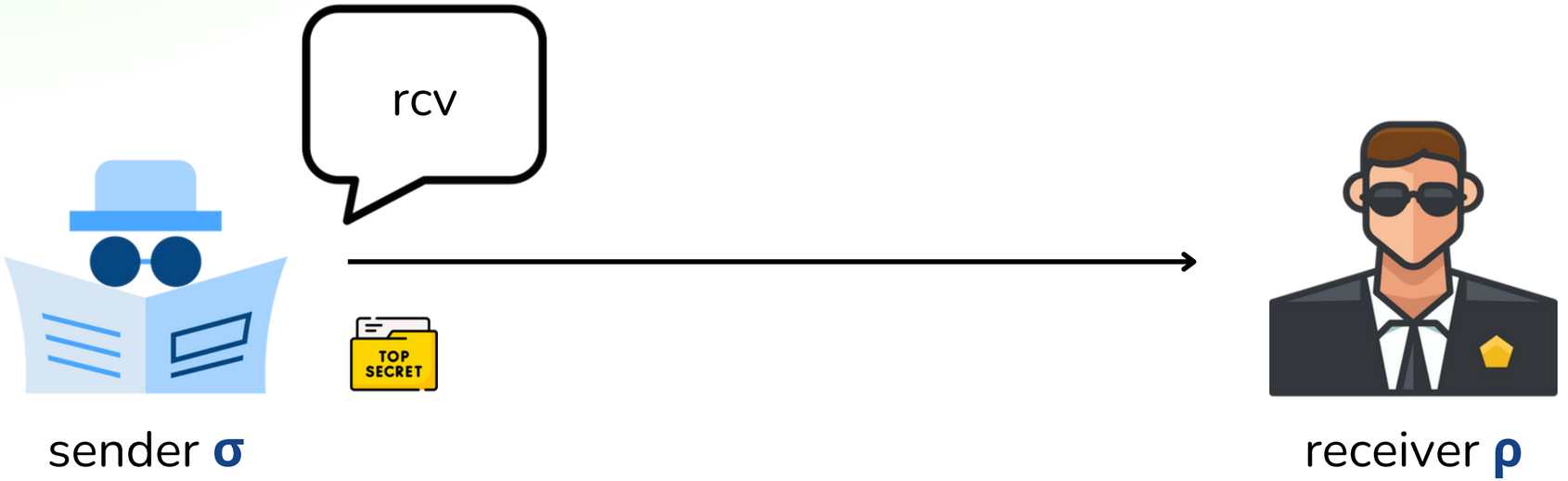
Roberta Cimorelli Belfiore[1], Andrea De Cosmo[2], and Anna Lisa Ferrara[1]

[1]University of Molise, Italy

[2]Leonardo S.p.A. Cyber & Security Solutions Division, Italy

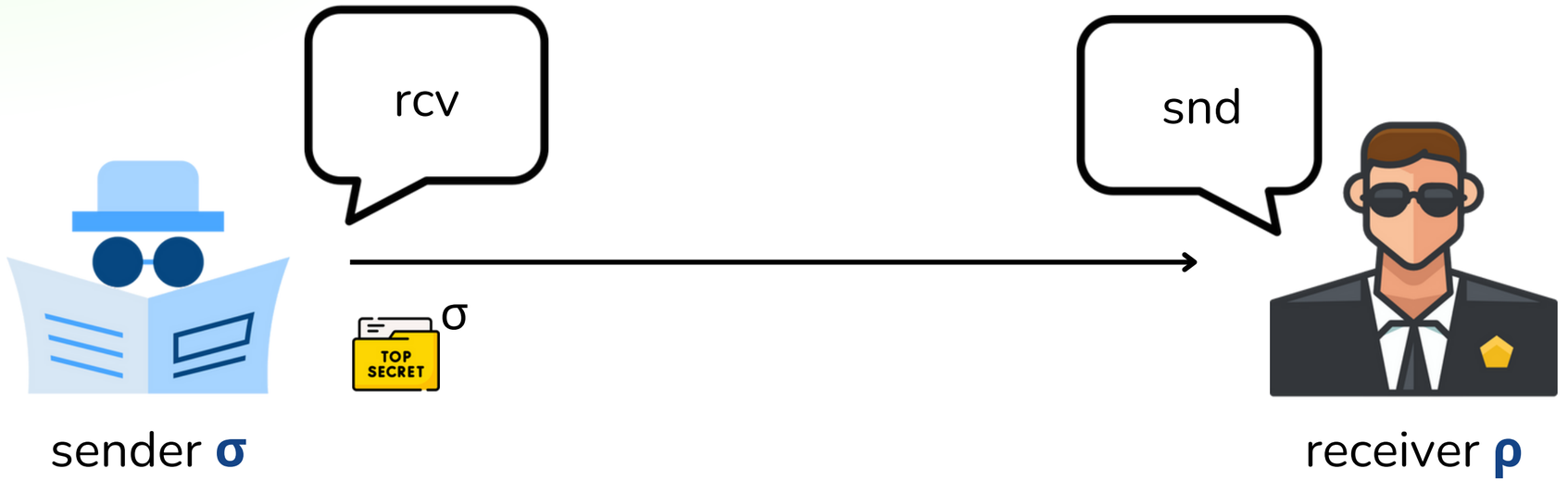
Identity-Based Matchmaking Encryption

In IBE, each user has a public-key that represents their identity.



Identity-Based Matchmaking Encryption

In IB-ME the sender inserts its identity **σ** into the ciphertext and specifies the identity of the recipient **rcv**.

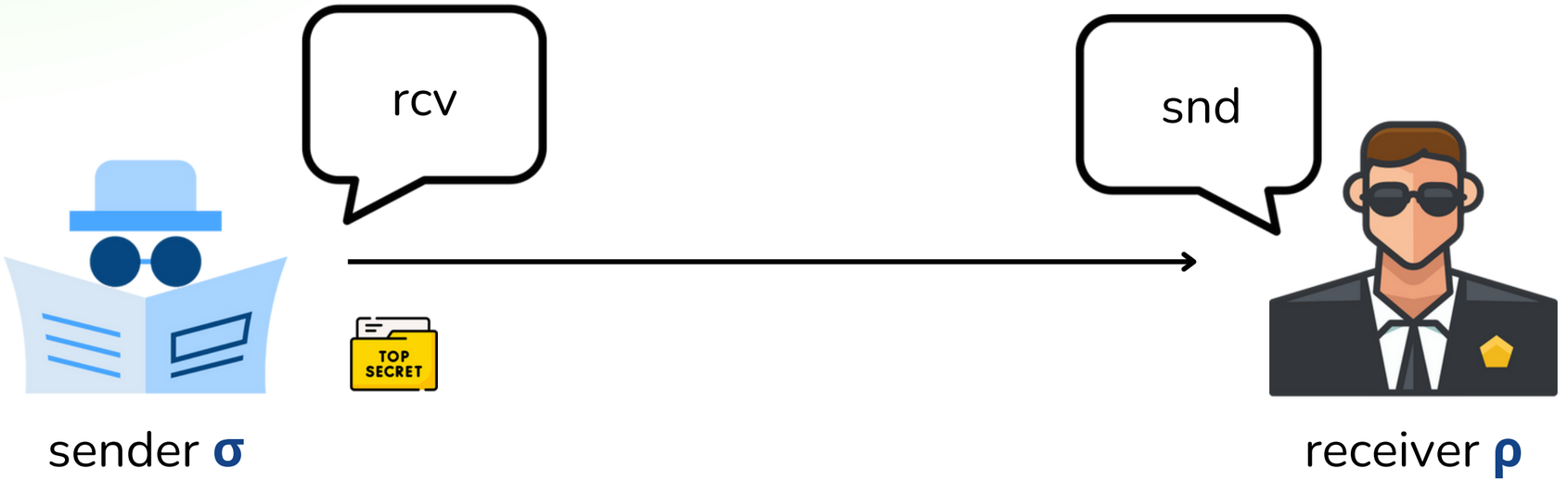


Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. 2019. Match Me if You Can: Matchmaking Encryption and Its Applications – CRYPTO 2019

Identity-Based Matchmaking Encryption

The sender inserts its identity **σ** into the ciphertext and specifies the identity of the recipient **rcv**.

The recipient **ρ** specifies the identity of the target sender snd on the fly.



Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. 2019. Match Me if You Can: Matchmaking Encryption and Its Applications – CRYPTO 2019

Identity-Based Matchmaking Encryption

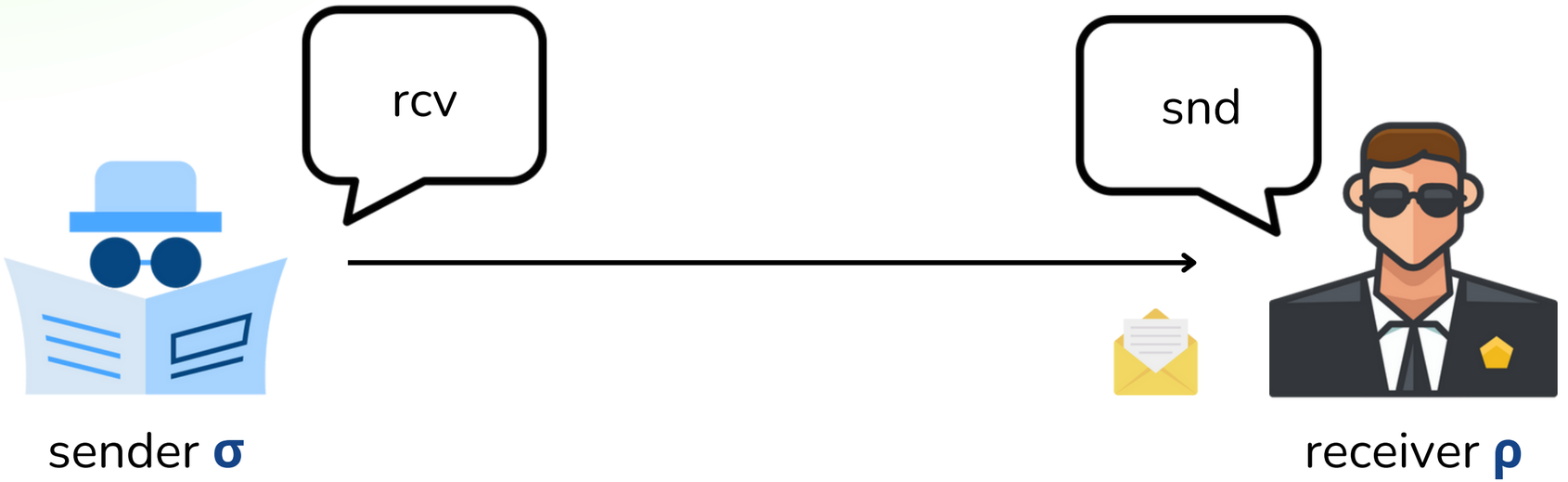
There is a match when **σ=snd and ρ=rcv**

Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. 2019. Match Me if You Can: Matchmaking Encryption and Its Applications – CRYPTO 2019

Identity-Based Matchmaking Encryption

There is a match when **σ=snd and ρ=rcv**

Only in case of a match the original message is **revealed**.



Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. 2019. Match Me if You Can: Matchmaking Encryption and Its Applications – CRYPTO 2019

Identity-Based Matchmaking Encryption

We say that a **mismatch** occurs when **σ ≠ snd** or **ρ ≠ rcv**.

Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. 2019. Match Me if You Can: Matchmaking Encryption and Its Applications – CRYPTO 2019

Identity-Based Matchmaking Encryption

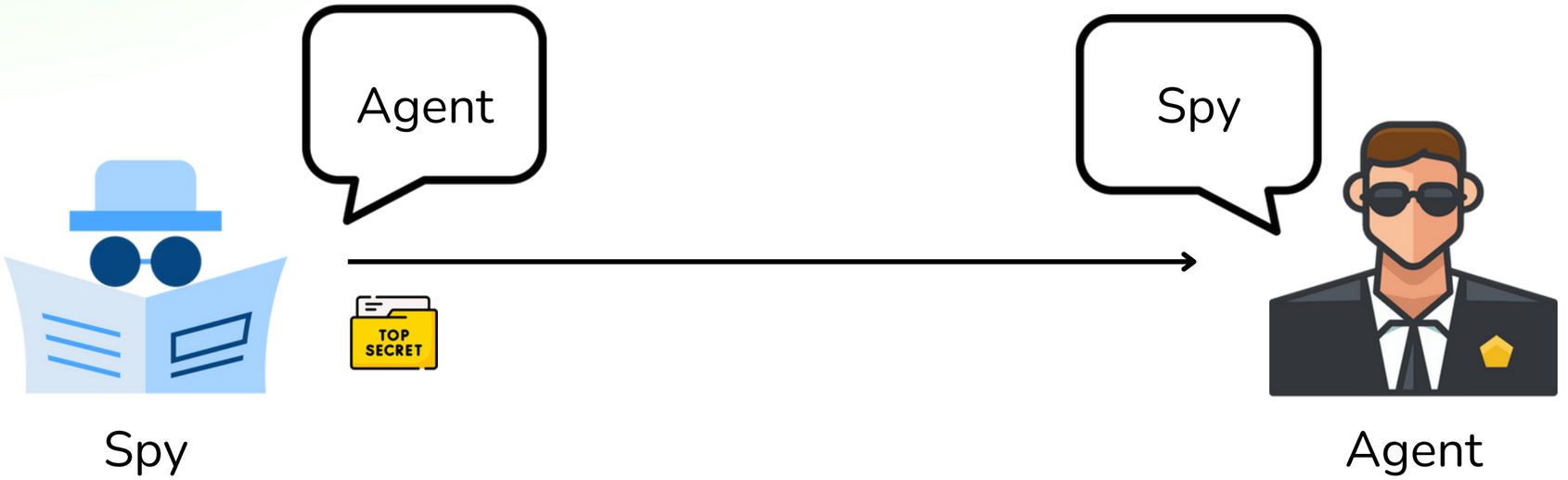
We say that a **mismatch** occurs when **σ ≠ snd or ρ ≠ rcv.**

When a mismatch occurs, **nothing** is leaked.

Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. 2019. Match Me if You Can: Matchmaking Encryption and Its Applications – CRYPTO 2019

Applications Scenario

Identity-Based Matchmaking Encryption enables several interesting applications, such as covert communication services



Related Works

Ateniese et al. (Crypto 2019)

An efficient identity-based scheme for equality policies, with provable security in the random oracle model under the standard BDH assumption.

- Privacy
- Authenticity

Francati et al. (IndoCrypt 2021)

IB-ME construction without random oracle, based on the hardness of the decisional truncated ABDHE assumption.

- **Enhanced Privacy**
- Authenticity

Constructing an IB-ME scheme that offers both **enhanced privacy and authenticity** under **standard** assumptions with particular emphasis on **post quantum** security.

Goal

Formal Definition

Setup(1^λ)	GENERATES MASTER KEYS **(mpk, msk)**.
SKGen(msk, σ)	GENERATES ENCRYPTION KEY **ek_σ** FOR SENDER IDENTITY **σ**.
RKGen(msk, ρ)	GENERATES DECRYPTION KEY **dk_ρ** FOR RECEIVER IDENTITY **ρ**.
Enc(ek_σ, rcv, m)	ENCRYPTS A MESSAGE **m** FOR A SPECIFIC RECEIVER IDENTITY **rcv.**
Dec(dk_ρ, snd, c)	DECRYPTS A CIPHERTEXT **c** SPECIFYING THE SENDER IDENTITY **snd** ON THE FLY. RETURNS THE MESSAGE m OR ⊥ ON FAILURE.

Formal Definition

Setup(1^λ)	GENERATES MASTER KEYS (**mpk, msk**).
SKGen(msk, σ)	GENERATES ENCRYPTION KEY **ek_σ** FOR SENDER IDENTITY **σ**.
RKGen(msk, ρ)	GENERATES DECRYPTION KEY **dk_ρ** FOR RECEIVER IDENTITY **ρ**.
Enc(ek_σ, rcv, m)	ENCRYPTS A MESSAGE **m** FOR A SPECIFIC RECEIVER IDENTITY **rcv.**
Dec(dk_ρ, snd, c)	DECRYPTS A CIPHERTEXT **c** SPECIFYING THE SENDER IDENTITY **snd** ON THE FLY. RETURNS THE MESSAGE m OR ⊥ ON FAILURE.

Formal Definition

Setup(1^λ)	GENERATES MASTER KEYS (**mpk, msk**).
SKGen(msk, σ)	GENERATES ENCRYPTION KEY **ek_σ** FOR SENDER IDENTITY **σ**.
RKGen(msk, ρ)	GENERATES DECRYPTION KEY **dk_ρ** FOR RECEIVER IDENTITY **ρ**.
Enc(ek_σ, rcv, m)	ENCRYPTS A MESSAGE **m** FOR A SPECIFIC RECEIVER IDENTITY **rcv.**
Dec(dk_ρ, snd, c)	DECRYPTS A CIPHERTEXT **c** SPECIFYING THE SENDER IDENTITY **snd** ON THE FLY. RETURNS THE MESSAGE m OR ⊥ ON FAILURE.

Formal Definition

Setup(1^λ)	GENERATES MASTER KEYS (**mpk, msk**).
SKGen(msk, σ)	GENERATES ENCRYPTION KEY **ek_σ** FOR SENDER IDENTITY **σ**.
RKGen(msk, ρ)	GENERATES DECRYPTION KEY **dk_ρ** FOR RECEIVER IDENTITY **ρ**.
Enc(ek_σ, rcv, m)	ENCRYPTS A MESSAGE **m** FOR A SPECIFIC RECEIVER IDENTITY **rcv.**
Dec(dk_ρ, snd, c)	DECRYPTS A CIPHERTEXT **c** SPECIFYING THE SENDER IDENTITY **snd** ON THE FLY. RETURNS THE MESSAGE m OR ⊥ ON FAILURE.

Formal Definition

Setup(1^λ)	GENERATES MASTER KEYS (**mpk, msk**).
SKGen(msk, σ)	GENERATES ENCRYPTION KEY **ek_σ** FOR SENDER IDENTITY **σ**.
RKGen(msk, ρ)	GENERATES DECRYPTION KEY **dk_ρ** FOR RECEIVER IDENTITY **ρ**.
Enc(ek_σ, rcv, m)	ENCRYPTS A MESSAGE **m** FOR A SPECIFIC RECEIVER IDENTITY **rcv.**
Dec(dk_ρ, snd, c)	DECRYPTS A CIPHERTEXT **c** SPECIFYING THE SENDER IDENTITY **snd** ON THE FLY. RETURNS THE MESSAGE m OR ⊥ ON FAILURE.

Security Properties
Enhanced Privacy

Ensure that σ, rcv, and the ciphertext-derived message remain hidden.

$\mathsf{Game}^{\text{ib-priv}^+}_{\Pi,\mathcal{A}}(\lambda)$

1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1) \leftarrow\$ \mathcal{A}_1^{\mathcal{O}_1, \mathcal{O}_2}(1^\lambda, \mathsf{mpk})$
3. $\sigma_0 \leftarrow\$ \mathbf{ID}_0$
4. $\sigma_1 \leftarrow\$ \mathbf{ID}_1$
5. $\mathsf{ek}_{\sigma_0} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_0)$
6. $\mathsf{ek}_{\sigma_1} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_1)$
7. $b \leftarrow\$ \{0, 1\}$
8. $\mathsf{c} \leftarrow\$ \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, \mathsf{m}_b)$
9. $b' \leftarrow\$ \mathcal{A}_2^{\mathcal{O}_1, \mathcal{O}_2, \{\mathcal{O}_3^i\}_{i \in \{0,1\}}}(1^\lambda, \mathsf{c})$
10. **if** $(b' = b)$ **return** 1
11. **else return** 0

Adversary wins if he can **correctly determine** which message was encrypted by the challenger.

Security Properties
Enhanced Privacy

$\mathsf{Game}^{\text{ib-priv}^+}_{\Pi,\mathcal{A}}(\lambda)$

1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1) \leftarrow\$ \mathcal{A}_1^{\mathcal{O}_1, \mathcal{O}_2}(1^\lambda, \mathsf{mpk})$
3. $\sigma_0 \leftarrow\$ \mathbf{ID}_0$
4. $\sigma_1 \leftarrow\$ \mathbf{ID}_1$
5. $\mathsf{ek}_{\sigma_0} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_0)$
6. $\mathsf{ek}_{\sigma_1} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_1)$
7. $b \leftarrow\$ \{0, 1\}$
8. $\mathsf{c} \leftarrow\$ \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, \mathsf{m}_b)$
9. $b' \leftarrow\$ \mathcal{A}_2^{\mathcal{O}_1, \mathcal{O}_2, \{\mathcal{O}_3^i\}_{i \in \{0,1\}}}(1^\lambda, \mathsf{c})$
10. **if** $(b' = b)$ **return** 1
11. **else return** 0

Adversary wins if he can **correctly determine** which message was encrypted by the challenger.

Security Properties
Enhanced Privacy

$\mathsf{Game}^{\text{ib-priv}^+}_{\Pi,\mathcal{A}}(\lambda)$

1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1) \leftarrow\$ \mathcal{A}_1^{\mathcal{O}_1, \mathcal{O}_2}(1^\lambda, \mathsf{mpk})$
3. $\sigma_0 \leftarrow\$ \mathbf{ID}_0$
4. $\sigma_1 \leftarrow\$ \mathbf{ID}_1$
5. $\mathsf{ek}_{\sigma_0} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_0)$
6. $\mathsf{ek}_{\sigma_1} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_1)$
7. $b \leftarrow\$ \{0, 1\}$
8. $\mathsf{c} \leftarrow\$ \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, \mathsf{m}_b)$
9. $b' \leftarrow\$ \mathcal{A}_2^{\mathcal{O}_1, \mathcal{O}_2, \{\mathcal{O}_3^i\}_{i \in \{0,1\}}}(1^\lambda, \mathsf{c})$
10. **if** $(b' = b)$ **return** 1
11. **else return** 0

Adversary wins if he can **correctly determine** which message was encrypted by the challenger.

Security Properties
Enhanced Privacy

$\mathsf{Game}^{\text{ib-priv}^+}_{\Pi,\mathcal{A}}(\lambda)$

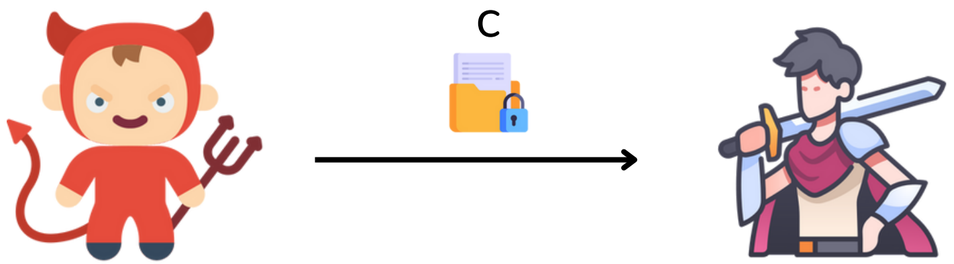
1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1) \leftarrow\$ \mathcal{A}_1^{\mathcal{O}_1, \mathcal{O}_2}(1^\lambda, \mathsf{mpk})$
3. $\sigma_0 \leftarrow\$ \mathbf{ID}_0$
4. $\sigma_1 \leftarrow\$ \mathbf{ID}_1$
5. $\mathsf{ek}_{\sigma_0} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_0)$
6. $\mathsf{ek}_{\sigma_1} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_1)$
7. $b \leftarrow\$ \{0, 1\}$
8. $\mathsf{c} \leftarrow\$ \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, \mathsf{m}_b)$
9. $b' \leftarrow\$ \mathcal{A}_2^{\mathcal{O}_1, \mathcal{O}_2, \{\mathcal{O}_3^i\}_{i \in \{0,1\}}}(1^\lambda, \mathsf{c})$
10. **if** $(b' = b)$ **return** 1
11. **else return** 0

Adversary wins if he can **correctly determine** which message was encrypted by the challenger.

Security Properties
Authenticity

$\mathsf{Game}^{\text{ib-auth}}_{\Pi,\mathcal{A}}(\lambda)$

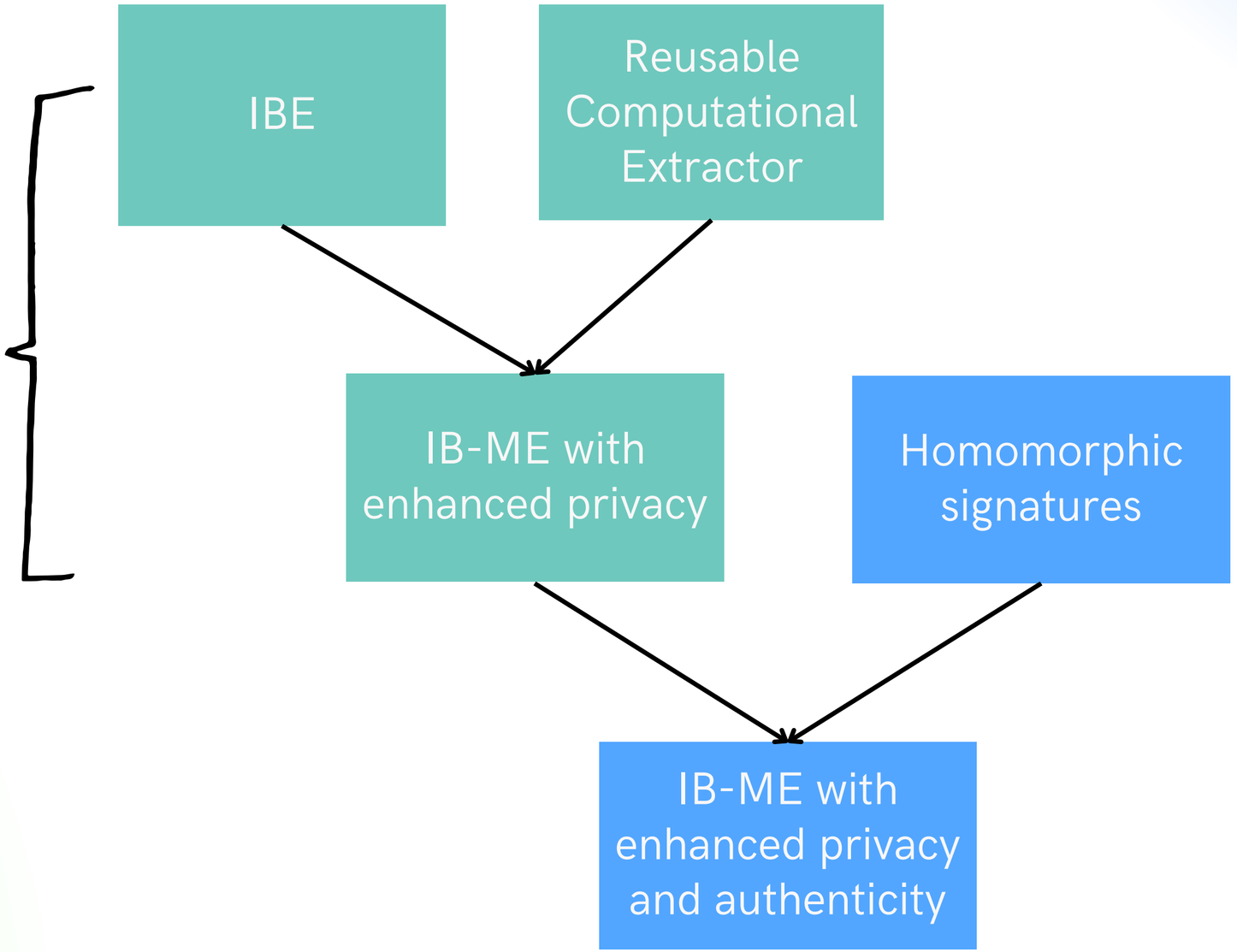
1. $(\mathsf{mpk}, \mathsf{msk}) \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{c}, \rho, \mathsf{snd}) \leftarrow\!\$\ \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2}(1^\lambda, \mathsf{mpk})$
3. $\mathsf{dk}_\rho \leftarrow\!\$\ \mathsf{RKGen}(\mathsf{msk}, \rho)$
4. $\mathsf{m} := \mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{snd}, \mathsf{c})$
5. **if** $\forall \sigma \in Q_{\mathcal{O}_1} : (\sigma \neq \mathsf{snd}) \wedge (\mathsf{m} \neq \bot)$
   return 1
6. **else return** 0



Adversary wins if **c** is a valid ciphertext embedding σ without knowing the encryption key ek_σ

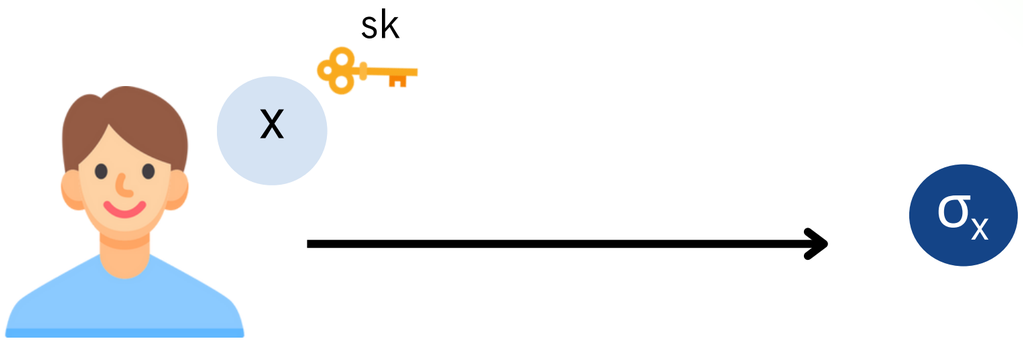
Our Approach

Similar to **Francati** et al approach

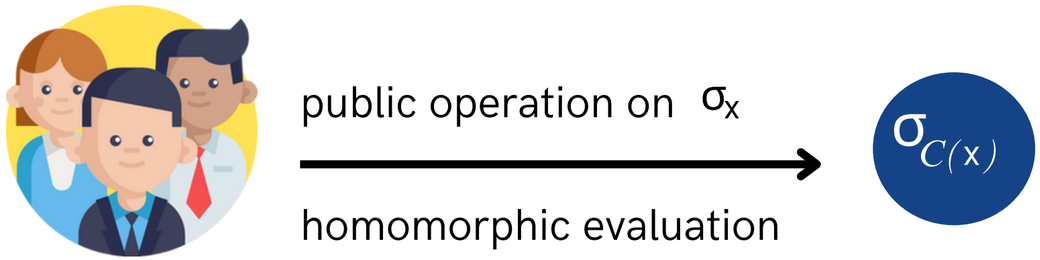


Homomorphic Signatures

Homomorphic signatures
enable computation on **signed** data.



σ_x = Sign(sk,x)

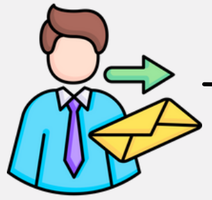


$\sigma_{C(x)}$ = Eval(pk, σ, C)

Our Construction

IB-ME Π = (Setup, SKGen, RKGen, Enc, Dec)
HS Σ = (Setup, Sign, Eval, Verify)

ENCRYPTION



$$\xrightarrow[\Sigma.\mathrm{Eval}(\mathrm{pk},\ \sigma,\ \varsigma_\sigma,\ C);]{\Pi.\mathrm{Enc}(\mathrm{ek}_\sigma,\mathrm{rcv},\mathrm{m})} \mathrm{c} = (\mathrm{c}',\ \varsigma_{C(\sigma)})$$

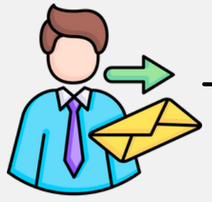
$ek_\sigma := (\sigma, \varsigma_\sigma)$

$\varsigma_\sigma := \Sigma.\mathrm{Sign}\ (\mathrm{sk}, \sigma)$

Our Construction

IB-ME Π = (Setup, SKGen, RKGen, Enc, Dec)
HS Σ = (Setup, Sign, Eval, Verify)

ENCRYPTION



$\Pi.\mathrm{Enc}(ek_\sigma, rcv, m)$

$\Sigma.\mathrm{Eval}(pk, \sigma, \varsigma_\sigma, C);$

$c = (c', \varsigma_{C(\sigma)})$

$ek_\sigma := (\sigma, \varsigma_\sigma)$

$\varsigma_\sigma := \Sigma.\mathrm{Sign}(sk, \sigma)$

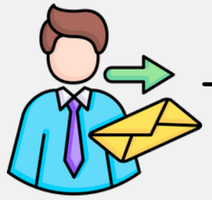
Our Construction

IB-ME Π = (Setup, SKGen, RKGen, Enc, Dec)
HS Σ = (Setup, Sign, Eval, Verify)

m, ρ, snd, c are hird wired in the circuit

$$C(\sigma) := \begin{cases} 1 & \text{iff snd} = \sigma \text{ and } m = \Pi.\mathrm{Dec}(dk_\rho, \text{snd}, c'), \\ & \text{where } dk_\rho \leftarrow\$\ \Pi.\mathrm{RKGen}(msk', \rho); \\ 0 & \text{otherwise} \end{cases}$$

ENCRYPTION



$$\xrightarrow[\Sigma.\mathrm{Eval}(pk, \sigma, \varsigma_\sigma, C);]{\Pi.\mathrm{Enc}(ek_\sigma, rcv, m)} c = (c', \varsigma_{C(\sigma)})$$

$ek_\sigma := (\sigma, \varsigma_\sigma)$

$\varsigma_\sigma := \Sigma.\mathrm{Sign}(sk, \sigma)$

signature on 1

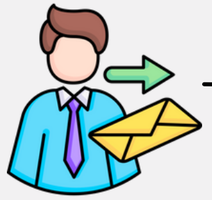
Our Construction

IB-ME Π = (Setup, SKGen, RKGen, Enc, Dec)
HS Σ = (Setup, Sign, Eval, Verify)

m, ρ, snd, c are hird wired in the circuit

$$C(\sigma) := \begin{cases} 1 & \text{iff } \mathrm{snd} = \sigma \text{ and } m = \Pi.\mathrm{Dec}(dk_\rho, \mathrm{snd}, c'), \\ & \text{where } dk_\rho \leftarrow\$\ \Pi.\mathrm{RKGen}(msk', \rho); \\ 0 & \text{otherwise} \end{cases}$$

ENCRYPTION



$$\xrightarrow[\Sigma.\mathrm{Eval}(pk, \sigma, \varsigma_\sigma, C);]{\Pi.\mathrm{Enc}(ek_\sigma, rcv, m)} c = (c', \varsigma_{C(\sigma)})$$

$ek_\sigma := (\sigma, \varsigma_\sigma)$

$\varsigma_\sigma := \Sigma.\mathrm{Sign}(sk, \sigma)$

signature on 1

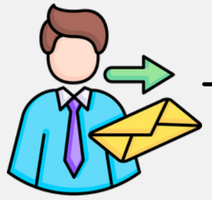
Our Construction

IB-ME Π = (Setup, SKGen, RKGen, Enc, Dec)
HS Σ = (Setup, Sign, Eval, Verify)

m, ρ, snd, c are hird wired in the circuit

$$C(\sigma) := \begin{cases} 1 & \text{iff snd} = \sigma \text{ and } \mathrm{m} = \Pi.\mathrm{Dec}(\mathrm{dk}_\rho, \mathrm{snd}, \mathrm{c}'), \\ & \text{where } \mathrm{dk}_\rho \leftarrow\$\ \Pi.\mathrm{RKGen}(\mathrm{msk}', \rho); \\ 0 & \text{otherwise} \end{cases}$$

ENCRYPTION



$$\xrightarrow[\Sigma.\mathrm{Eval}(\mathrm{pk}, \sigma, \varsigma_\sigma, C);]{\Pi.\mathrm{Enc}(\mathrm{ek}_\sigma, \mathrm{rcv}, \mathrm{m})} \mathrm{c} = (\mathrm{c}', \varsigma_{C(\sigma)})$$

$\mathrm{ek}_\sigma := (\sigma, \varsigma_\sigma)$

$\varsigma_\sigma := \Sigma.\mathrm{Sign}(\mathrm{sk}, \sigma)$

signature on 1

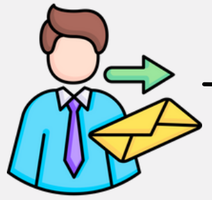
Our Construction

IB-ME Π = (Setup, SKGen, RKGen, Enc, Dec)
HS Σ = (Setup, Sign, Eval, Verify)

m, ρ, snd, c are hird wired in the circuit

$$C(\sigma) := \begin{cases} 1 & \text{iff snd} = \sigma \text{ and } m = \Pi.\mathrm{Dec}(dk_\rho, \text{snd}, c'), \\ & \text{where } dk_\rho \leftarrow\$\ \Pi.\mathrm{RKGen}(msk', \rho); \\ 0 & \text{otherwise} \end{cases}$$

ENCRYPTION



$$\xrightarrow[\Sigma.\mathrm{Eval}(pk, \sigma, \varsigma_\sigma, C);]{\Pi.\mathrm{Enc}(ek_\sigma, rcv, m)} c = (c', \varsigma_{C(\sigma)})$$

$ek_\sigma := (\sigma, \varsigma_\sigma)$

$\varsigma_\sigma := \Sigma.\mathrm{Sign}(sk, \sigma)$

signature on 1

DECRYPTION

dk_ρ

- $m := \Pi.\mathrm{Dec}(dk_\rho, \text{snd}, c')$
- $\Sigma.\mathrm{RVerify}(pk, 1, \varsigma_{C(\sigma)}, C)$

Security of Our Construction

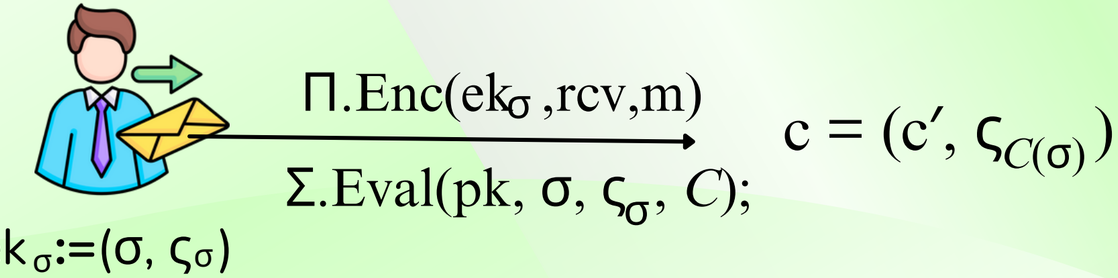
Authenticity:

follows by the unforgeability of the homomorphic signature scheme

Security of Our Construction

Authenticity:

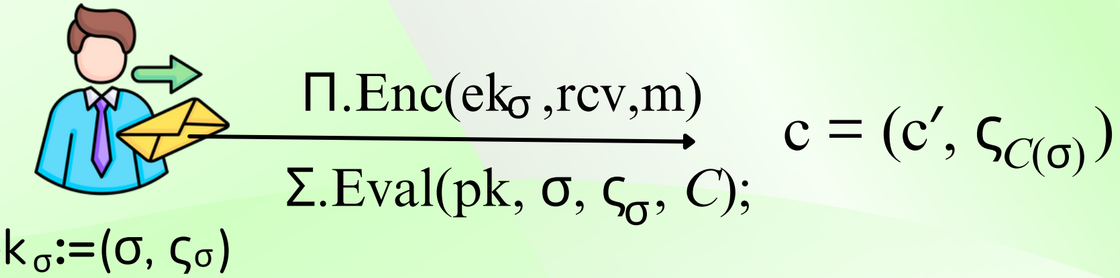
follows by the unforgeability of the homomorphic signature scheme



Security of Our Construction

Authenticity:

follows by the unforgeability of the homomorphic signature scheme



Enhanced privacy:

follows by the privacy property of the underlying IB-ME along with the context-hiding property of the homomorphic signature scheme

Security of Our Construction

Authenticity:

follows by the unforgeability of the homomorphic signature scheme

$\Pi.\mathrm{Enc}(ek_\sigma, rcv, m)$

$\Sigma.\mathrm{Eval}(pk, \sigma, \varsigma_\sigma, C);$

$c = (c', \varsigma_{C(\sigma)})$

$ek_\sigma := (\sigma, \varsigma_\sigma)$

Enhanced privacy:

follows by the privacy property of the underlying IB-ME along with the context-hiding property of the homomorphic signature scheme

privacy of IB-ME

↑

$c = (c', \varsigma_{C(\sigma)})$

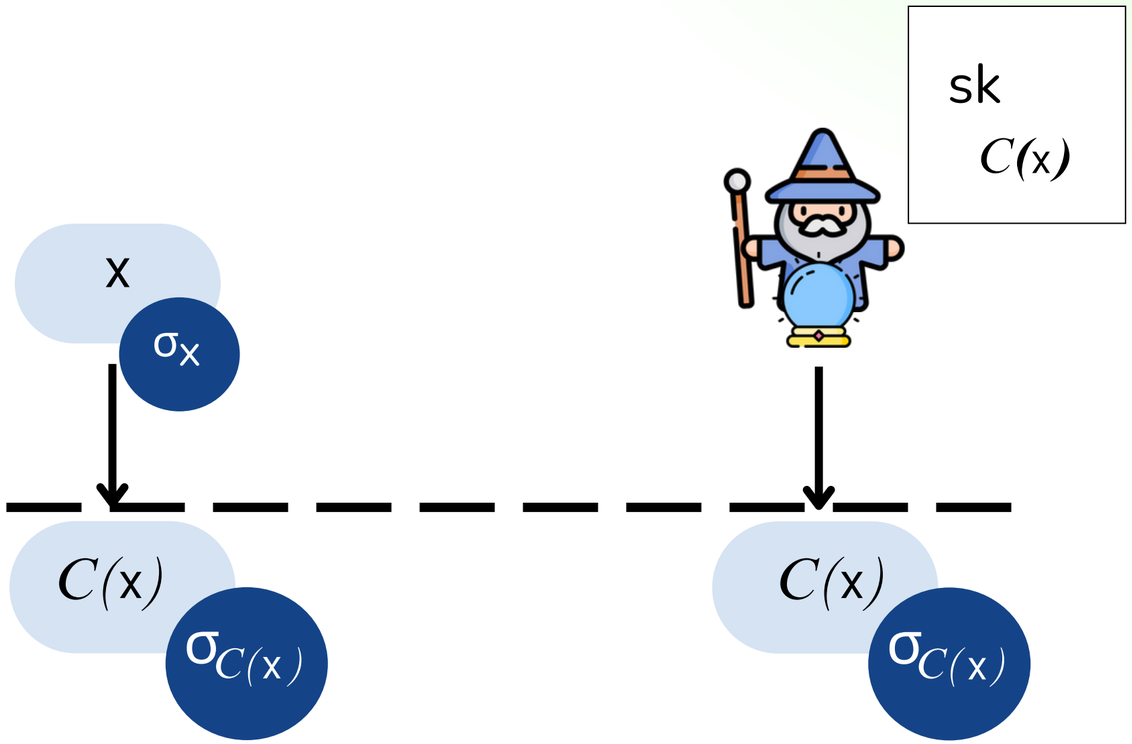
↓

context-hiding of homomorphic signatures

Security of Our Construction

$\sigma_{C(x)}$ hides the original input x up to what is revealed by $C(x)$

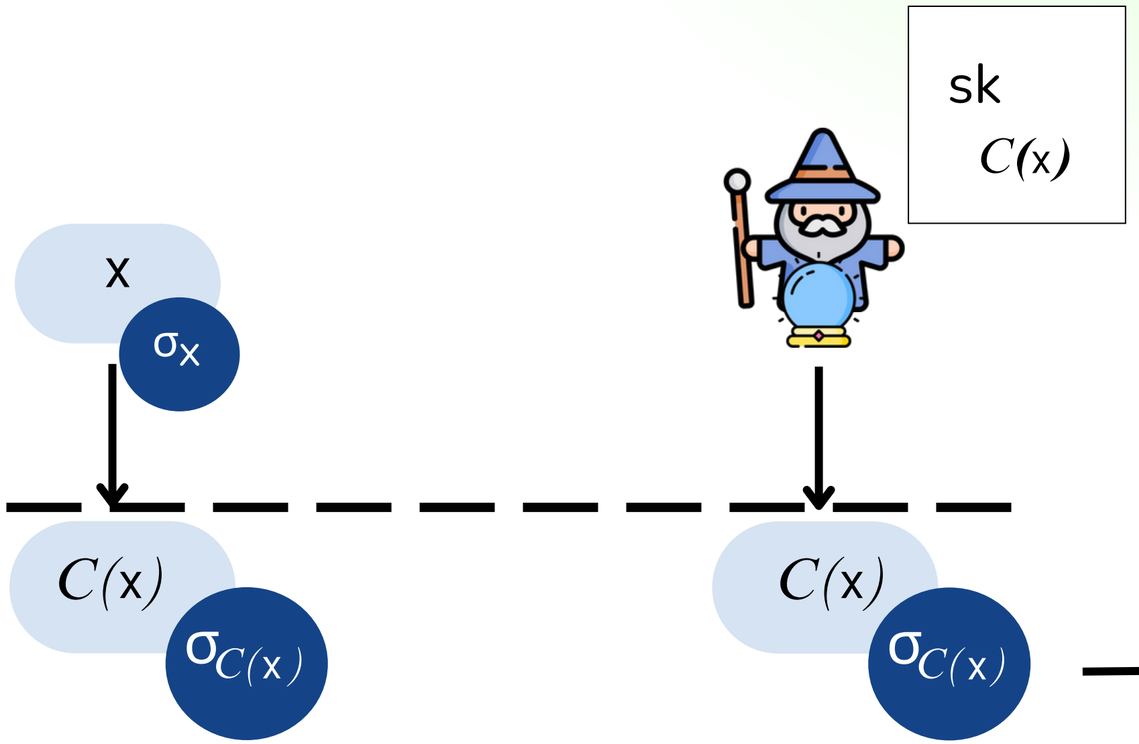
Context-Hiding



Security of Our Construction

$\sigma_{C(x)}$ hides the original input x
up to what is revealed by $C(x)$

Context-Hiding



privacy of IB-ME

$c = (c', \varsigma_{C(\sigma)})$

context-hiding of
homomorphic signatures

Instantiations

IBE schemes applicable to our construction:

Schemes	Assumption	ROM	Quantum Resistant
BF'01–Basicident [15]	DBDH	✓	✗
BW'06 [34]	DLin	✗	✗
DLP'14 [35]	NTRU/RLWE	✓	✓
GPV'08 [32]	LWE	✓	✓
ABB10'10–Select-ID [10]	LWE	✗	✓

Fully Homomorphic Signature schemes applicable to our construction:

Schemes	Assumption	Quantum Resistant
Boyen et al. [16]	SIS	✓
Gorbunov et al. [17]	SIS	✓
Luo et al. [33]	SIS	✓

Conclusions

Generic construction
of an IB-ME scheme with:

- **enhanced privacy**
- **authenticity**
- **standard** assumptions
- **post quantum** security

ACNS 2024: 22nd International Conference on Applied Cryptography and Network Security | Abu Dhabi, UAE

Thanks
for your attention

Roberta Cimorelli Belfiore[1], Andrea De Cosmo[2], and Anna Lisa Ferrara[1]

[1]University of Molise, Italy

[2]Leonardo S.p.A. Cyber & Security Solutions Division, Italy