Matchmaking Encryption. A quick overview

Luigi Russo

June 11, 2021

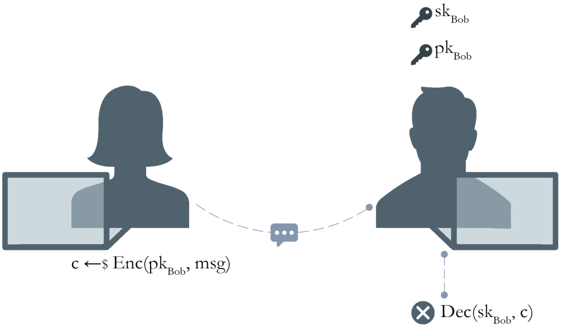
Road Map

- Introduction
- Matchmaking Encryption
- Showcase Application
- Enhanced Security *
- Beyond ME

1. Introduction

Public Key Encryption

- Bob generates a pair of keys (pk, *sk*) and <u>publishes</u> pk
- Alice uses pk to encrypt the message
- Bob uses *sk* to decrypt



Identity-Based Encryption

IBE is a type of PKE, proposed by Shamir in 1984, in which:

- public keys are identities (email address, phone numbers, etc.)
- a trusted party distributes the decryption keys to the receivers

The first construction of IBE was by Boneh and Franklin in 2001.

Attribute-Based Encryption

ABE is a type of PKE, proposed by Sahai and Waters in 2004, in which:

- secret keys and ciphertext depend upon user attributes
- the decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext
- the same ciphertext may be potentially decrypted by different users
- a trusted party distributes the keys

Types of ABE

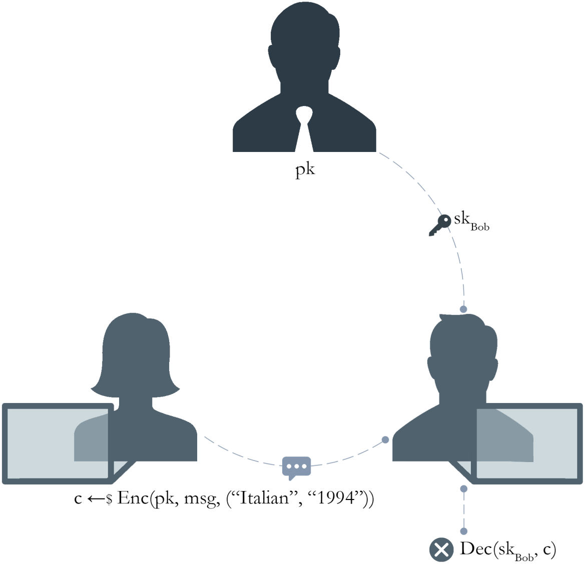
Ciphertext-Policy ABE

- the decryption key is associated with receiver's attributes
- the policy is embedded in the ciphertext

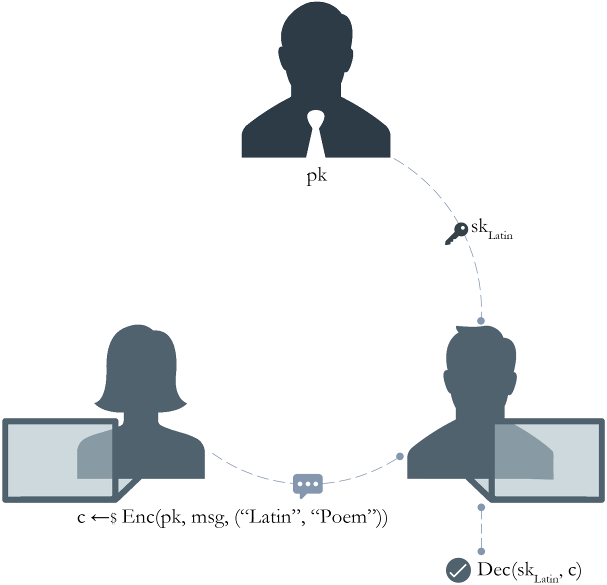
Key-Policy ABE

- the decryption key is associated with a policy
- the ciphertext is annotated with attribute sets

CP-ABE: an example


pk
sk_{Bob}
$c \leftarrow_{\$} Enc(pk, msg, (\text{“Italian”}, \text{“1994”}))$
$Dec(sk_{Bob}, c)$

KP-ABE: an example

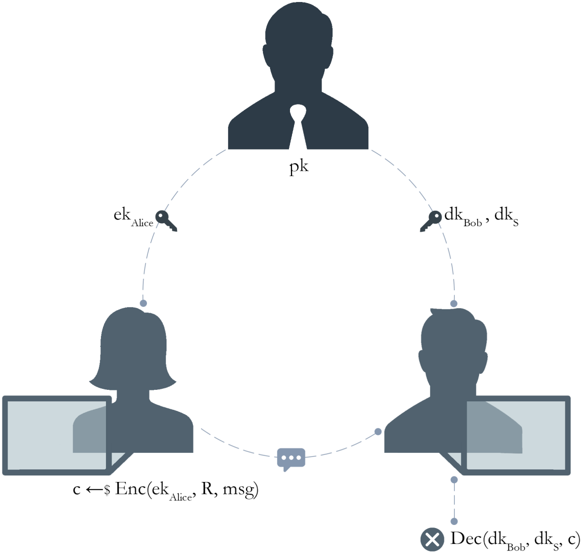

pk
sk_{Latin}
$c \leftarrow_{\$} \text{Enc}(pk, msg, (\text{“Latin”}, \text{“Poem”}))$
✓ $\text{Dec}(sk_{\text{Latin}}, c)$

2. Matchmaking Encryption

Matchmaking Encryption

- The sender σ can specify a policy R for the receiver
- The receiver ρ can specify a policy S for the sender
- The policies are represented as circuits $\mathsf{C} : \{0, 1\}^* \rightarrow \{0, 1\}$
- If both the policies are matched, i.e. $\mathsf{S}(\sigma) = \mathsf{R}(\rho) = 1$, then the decryption is successful. Otherwise nothing is leaked.
- A trusted party stores the secrets

ME: an example


pk
$\mathsf{ek}_{\text{Alice}}$
$\mathsf{dk}_{\text{Bob}}, \mathsf{dk}_{S}$
$c \leftarrow_{\$} \mathsf{Enc}(\mathsf{ek}_{\text{Alice}}, R, \mathsf{msg})$
✗ $\mathsf{Dec}(\mathsf{dk}_{\text{Bob}}, \mathsf{dk}_{S}, c)$

ME vs ABE

ME $\Rightarrow$ CP-ABE

- "Ignore" the policy S set by the receiver (S is a tautology)
- Match condition becomes: $\mathsf{R}(\rho) \wedge 1 \stackrel{?}{=} 1$

ME $\Rightarrow$ KP-ABE

- "Ignore" the policy R set by the sender (R is a tautology)
- Choose sender attributes on the fly
- Match condition becomes: $1 \wedge S(\sigma) \stackrel{?}{=} 1$

ME from ABE?

It could be possible to build ME as a 2-layers ABE.

The sender

- encrypts the message with a KP-ABE scheme (1st layer)
- then, encrypts this ciphertext using a CP-ABE (2nd layer)

The receiver

- decrypts the outer layer with a key associated with his attributes (CP-ABE)
- then, decrypts using a key associated with a policy (KP-ABE)

The receiver knows which policy (if any) does not match.

Functional Encryption

A type of PKE where:

- secret keys are associated with some function F
- we can learn a function of what the ciphertext is encrypting

$$\mathsf{Dec}(sk_F, \mathsf{Enc}(pk, x)) = F(x)$$

Randomized FE (rFE)

- F accepts some fresh randomness

$$\mathsf{Dec}(sk_F, \mathsf{Enc}(pk, x)) = F(x; r)$$

ME: Construction

ME uses a combination of rFe and FE schemes.

The inner layer is computed with FE (needs sk_S to decrypt).

$$F_S(\sigma, x) = \begin{cases} x, \text{ if } S(\sigma) = 1 \\ \perp, \text{ else} \end{cases}$$

The outer layer is computed with rFE (needs sk_ρ to decrypt).

$$F_\rho(\sigma, R, x; r) = \begin{cases} \mathsf{Enc}_{\mathsf{FE}}(\sigma, m), \text{ if } R(\rho) = 1 \\ \mathsf{Enc}_{\mathsf{FE}}(\perp, \perp), \text{ else} \end{cases}$$

To encrypt run $\mathsf{Enc}_{\mathsf{rFE}}(\sigma, R, x))$

To decrypt run $\mathsf{Dec}_{\mathsf{FE}}(\mathsf{sk}_S, \mathsf{Dec}_{\mathsf{rFE}}(\mathsf{sk}_\rho, c))$

Add Authenticity

- The previous construction lacks authenticity: anyone can encrypt using arbitrary sender attributes σ^*

- To solve this issue, the authority generates secret encryption keys ek_σ, i.e., a signature on σ.

- The sender proves in zero-knowledge that he knows a valid signature of the attributes σ used to produce the ciphertext, and attaches this proof π to the ciphertext. The receiver verifies π during the decryption.

3. Applications

Applications

- Social matchmaking
- Encrypting bids
- Encrypting votes
- Liability exemption
- ...

An Anonymous Bulletin Board

- A user that wants to post a message to the bulletin board IB-ME encrypts a message, specifying the policy rcv for the intended receiver

- Uploads the ciphertext to the web server through the Tor network

- The webserver makes these ciphertexts publicly available through a REST API

- A receiver can download all the ciphertexts and try to decrypt each of them, using his decryption keys

Distribution of the Keys

Need of a service that automatically converts email addresses (or phone numbers) into keys.

- out-of-band channel, external to TOR, but <u>trusted</u>
- another TOR hidden service
- integrated with an existing HSDir (already assumed to be trusted because downloaded from legitimate servers)
- ...

Space costs of IB-ME elements

Element	Cost	Bitsize
Encryption Key	$\|\mathsf{G}\|$	512
Decryption Key	$3\|\mathsf{G}\|$	1536
Message	$\|\mathsf{G}_T\|$	1024
Ciphertext	$2\|\mathsf{G}\| + \mathtt{padding}$	2129

Implementation: Charm's curve SS512

Security Guarantees

ME Privacy

- Messages are disclosed if and only if a match occurs

Plus

- IP addresses remain secret
- The connection between the client and the server remains hidden

4. CCA Security

Chosen-Ciphertext Attacks

- The attacker can gather information by obtaining the decryptions of chosen ciphertexts.
- From these pieces of information can attempt to break the security (e.g., recover the secret key)
- Fundamental in many practical situations

[Ateniese et al. 2019] did not consider CCA.

Thesis Contributions

1. Extend the security definitions to handle chosen-ciphertext attacks

    - CCA Privacy
    - CCA Authenticity

2. Provide blackbox constructions from CPA to CCA to obtain:

    - Privacy only
    - Authenticity only
    - Privacy and Authenticity

We have made this for both general and arranged settings.

Privacy

CPA setting

- captures secrecy of sender's inputs in presence of malicious receivers.
- in case of mismatch, the reason why the match did not occur is not revealed.

CCA extension

- the adversary may now additionally have access to a decryption oracle that answers decryption queries
- enforce privacy even in the presence of attackers who can obtain the decryption of possibly mauled ciphertexts.

Authenticity

CPA setting

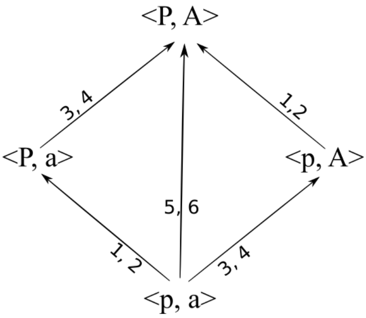
- captures security against malicious senders
- no adversary can spoof the sender identity

CCA extension

- the attacker is also given access to an encryption oracle
- captures authenticity in the presence of attackers that try to generate valid ciphertexts by mauling previously obtained ciphertexts

CCA Transformation Lattice

- P/p: CCA/CPA Privacy
- A/a: CCA/CPA Authenticity
- 1-6: lemmas



Digital Signatures

A digital signature $\Pi = (KGen, Sign, Ver)$ is a scheme for verifying the authenticity of digital messages.

The recipient knows that the message:

- was created by a known sender (**authentication**)
- was not altered in transit (**integrity**)

Non-Interactive Zero-Knowledge

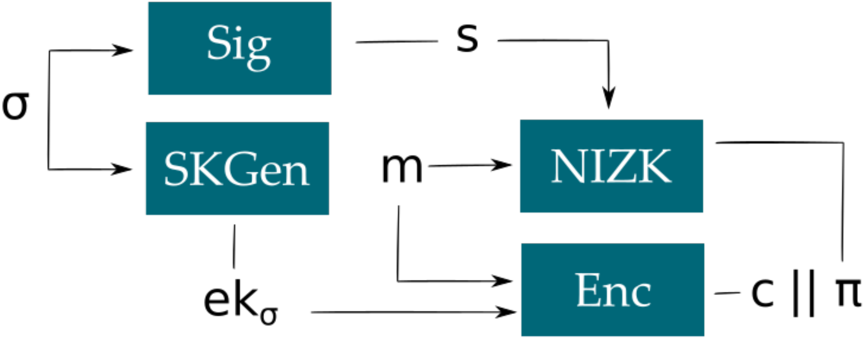
A NIZK proof system $\Pi = (\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ is a method by which the prover P can prove to the verifier V that he knows a value x:

- without interaction (**non-interactive**)

- without conveying any information apart from the fact that he knows the value x (**zero-knowledge**)

If P does not know x cannot produce a proof (**soundness**)

Showcase: CCA Direct Transformation

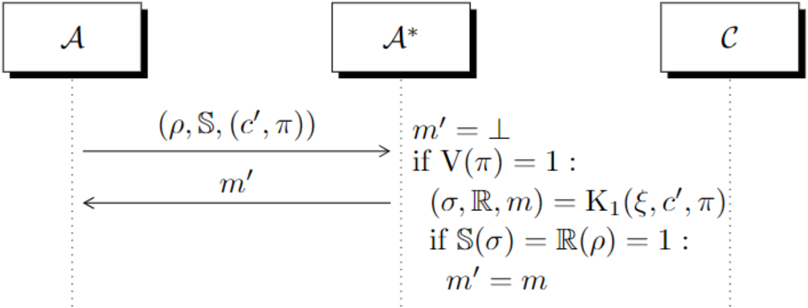


Sketch Proof

- CCA-authenticity reduced to EUF-CMA of Digital Signatures.
- CCA-privacy reduced to CPA-Privacy

CCA Privacy Reduction 1/2

The decryption oracle is simulated thanks to f-extractability of the NIZK.



$\mathcal{A}$

$\mathcal{A}^*$

$\mathcal{C}$

$(\rho, \mathbb{S}, (c', \pi))$

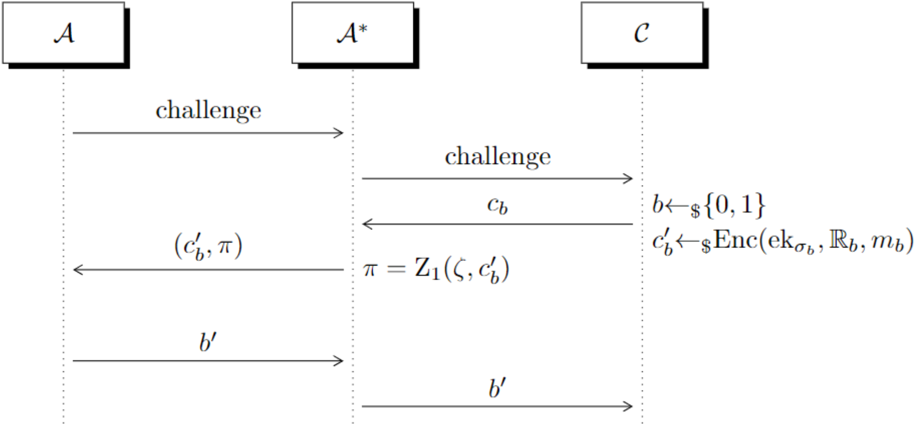
m'

```
$m' = \perp$
if $\mathrm{V}(\pi) = 1$ :
  $(\sigma, \mathbb{R}, m) = \mathrm{K}_1(\xi, c', \pi)$
  if $\mathbb{S}(\sigma) = \mathbb{R}(\rho) = 1$ :
    $m' = m$
```

* f-extractability: we can extract information from NIZK proofs if we know an extraction trapdoor ξ.

CCA Privacy Reduction 2/2

Here we can see how the last step of the reduction works. We attach a simulated proof π for c'$_b$ and let the attacker work on it


$\mathcal{A}$
$\mathcal{A}^*$
$\mathcal{C}$
challenge
challenge
c_b
$b \leftarrow_\$ \{0,1\}$
$c'_b \leftarrow_\$ \mathrm{Enc}(\mathrm{ek}_{\sigma_b}, \mathbb{R}_b, m_b)$
(c'_b, π)
$\pi = \mathrm{Z}_1(\zeta, c'_b)$
b'
b'

5. Beyond ME

Open problems

- ME from standard assumptions
- Efficient IB-ME constructions
- Mitigating Key Escrow
- Blackbox Constructions from ABE

Key Escrow

- All users’ private keys are issued by an unconditionally trusted authority

- Such an authority owns the master secret key of the system, and can decrypt all ciphertexts encrypted to any user

- Potentially this is the target for some attacker which can obtain private keys and redistribute them for malicious use

Can we reduce the trust in the authority in an ME system?

Registration-Based ME

- Substitute the trusted authority with a weaker entity called key curator who has no knowledge of any secret key

- Each party generates its own secret key and publicly register its identity and the corresponding public key to the key curator

- RBE schemes can be constructed from standard assumptions

Thanks!

Some extras

- ME vs A-ME
- ME Game-Based Definitions
- Dual Policy ABE
- Some formal definitions of cryptographic primitives
- CCA-Privacy Proof

6. Arranged ME

Many types of ME

- **General**
  - One private key per user
  - One key for each policy
- **Arranged**
  - A single key is associated with both attributes and policy
- **Identity-Based**
  - Attributes are simply identities

General Setting

- Key Generation, managed by the trusted party:
  - $\mathsf{SKGen}(\mathsf{msk}, \sigma)$
  - $\mathsf{RKGen}(\mathsf{msk}, \rho)$
  - $\mathsf{PolGen}(\mathsf{msk}, \mathsf{S})$
- Encryption. $\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{R}, \mathsf{m})$
- Decryption. $\mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{dk}_\mathsf{S}, \mathsf{c})$
- Correctness. If $\mathsf{S}(\sigma) = \mathsf{R}(\rho) = 1$:

$$\Pr[\mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{dk}_\mathsf{S}, \mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{R}, \mathsf{m})) = 1] \geqslant 1 - \mathsf{negl}(\lambda)$$

Arranged Matchmaking

- Key Generation, managed by the trusted party:
  - $\mathsf{SKGen}(\mathsf{msk}, \sigma)$
  - $\mathsf{RKGen}(\mathsf{msk}, \rho, \mathsf{S})$
  - ~~$\mathsf{PolGen}(\mathsf{msk}, \mathsf{S})$~~
- Encryption: $\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{R}, \mathsf{m})$
- Decryption: $\mathsf{Dec}(\mathsf{dk}_{\rho,\mathsf{S}}, \mathsf{c})$
- Correctness. If $\mathsf{S}(\sigma) = \mathsf{R}(\rho) = 1$:

$$\Pr[\mathsf{Dec}(\mathsf{dk}_{\rho,\mathsf{S}}, \mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{R}, \mathsf{m})) = 1] \geqslant 1 - \mathsf{negl}(\lambda)$$

7. ME: Game-Based Definitions

CCA Privacy Game

Captures secrecy of sender's inputs.

$\underline{G^{priv}_{\Pi,A}(\lambda, b)}$:

$$(mpk, kpol, msk) \leftarrow Setup(1^{\lambda})$$
$$(m_0, m_1, R_0, R_1, \sigma_0, \sigma_1, \alpha) \leftarrow A_1^{O_1,O_2,O_3,O_4}(1^{\lambda}, mpk)$$
$$ek_{\sigma_b} \leftarrow SKGen(msk, \sigma_b)$$
$$c \leftarrow Enc(ek_{\sigma_b}, R_b, m_b)$$
$$b' \leftarrow A_2^{O_1,O_2,O_3,O_4}(1^{\lambda}, c, \alpha)$$

CCA Authenticity Game

Captures security against malicious senders.

$\underline{G^{auth}_{\Pi,A}(\lambda)}:$

$$
\begin{aligned}
&(mpk, kpol, msk) \leftarrow Setup(1^\lambda)\\
&(c, \rho, S) \leftarrow A_1^{O_1,O_2,O_3,O_5}(1^\lambda, mpk)\\
&dk_\rho \leftarrow RKGen(msk, \rho_b)\\
&dk_S \leftarrow PolGen(kpol, S)\\
&m = Dec(dk_\rho, dk_S, c)
\end{aligned}
$$

$$(c \notin O_{O_5}) \wedge \forall \sigma \in Q_{O_1} : (S(\sigma) = 0) \wedge (m \neq \bot)$$

8. Dual-policy ABE

Dual-policy ABE

Similar to Arranged Matchmaking

- Both the sender and the receiver can specify a policy
- One private key per user
- A single decryption key is associated with both the receiver's policy and attributes

But

- In case of mismatch we know the reason why the match did not occur!

9. Some formal definitions

Signatures: a formal definition

A signature scheme $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ satisfies:

1. **Correctness.** $\forall \lambda \in \mathbb{N}, \forall (sk, pk) \leftarrow \mathsf{KGen}(1^\lambda)$, and $\forall m \in \mathcal{M}$:

$$P[\mathsf{Ver}(pk, m, \mathsf{Sign}(sk, m)) = 1] = 1.$$

2. **EUF-CMA.** $\forall$ PPT $\mathcal{A}$:

$$\Pr[G^{euf}_{\Pi,\mathcal{A}}(\lambda) = 1] \leqslant \mathsf{negl}(\lambda)$$

where $G^{euf}_{\Pi,\mathcal{A}}(\lambda)$ is the following game:

- $(sk, pk) \leftarrow \mathsf{KGen}(1^\lambda)$.
- $(m, s) \leftarrow \mathcal{A}^{\mathsf{Sign}(sk,\cdot)}(1^\lambda, pk)$
- If $m \notin Q_{\mathsf{Sign}}$, and $\mathsf{Ver}(pk, m, s) = 1$, output 1, else 0.

NIZK: a formal definition

A NIZK proof system $\Pi = (\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ for a relation R satisfies:

1. **Completeness.** $\forall y \in L$:

$$\Pr[\mathsf{V}(\omega, y, \pi) = 1 : \omega \leftarrow \mathsf{Gen}(1^\lambda), \pi \leftarrow \mathsf{P}(\omega, y, x)] = 1$$

2. **Soundness.** $\forall y \notin L, \forall \mathsf{P}^*$:

$$\Pr[\mathsf{V}(\omega, y, \pi) = 1 : \omega \leftarrow \mathsf{Gen}(1^\lambda), \pi \leftarrow \mathsf{P}^*(\omega, y)] \in \mathsf{negl}(\lambda)$$

3. **Zero-Knowledge.** $\exists (\mathsf{Z}_0, \mathsf{Z}_1)$ s.t. $\forall y \in L$:

$$\{\omega, \mathsf{Z}_1(\zeta, y) : (\omega, \zeta) \leftarrow \mathsf{Z}_0(1^\lambda)\}$$

$$\approx_c$$

$$\{\omega, \mathsf{P}(\omega, y, x) : \omega \leftarrow \mathsf{Gen}(1^\lambda)\}$$

True-Simulation Extractability

A NIZK Π for a relation R satisfies true-simulation f-extractability for a function f if $\exists$ PPT extractor $K = (K_0, K_1)$ s. t. $\forall A$:

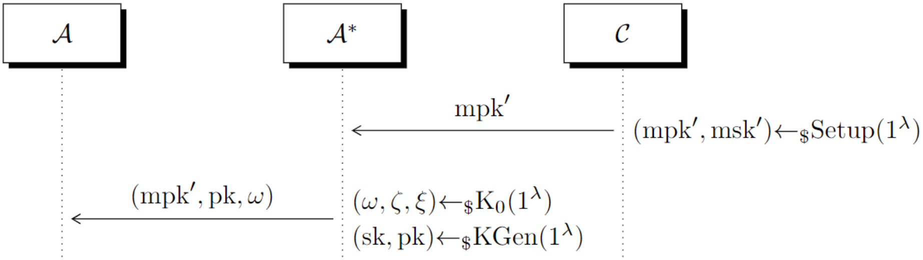
$$\Pr\left[\begin{array}{c} \mathsf{Ver}(\omega, y, \pi) = 1 \wedge \\ (y, \pi) \notin \mathcal{O}_{\mathcal{O}} \wedge \\ \forall x : f(x) = z, (y, x) \notin R \end{array} : \begin{array}{c} (\omega, \zeta, \xi) \leftarrow K_0(1^\lambda) \wedge \\ (y, \pi) \leftarrow A^{\mathcal{O}(\zeta, (\cdot, \cdot))}(\omega) \wedge \\ z \leftarrow K_1(\xi, y, \pi) \end{array}\right]$$

* the oracle $\mathcal{O}(\zeta, \cdot, \cdot)$ takes as input a pair (y, x) and returns $Z_1(\zeta, y)$ if $(y, x) \in R$ (and otherwise $\perp$).

10. CCA-Privacy Proof

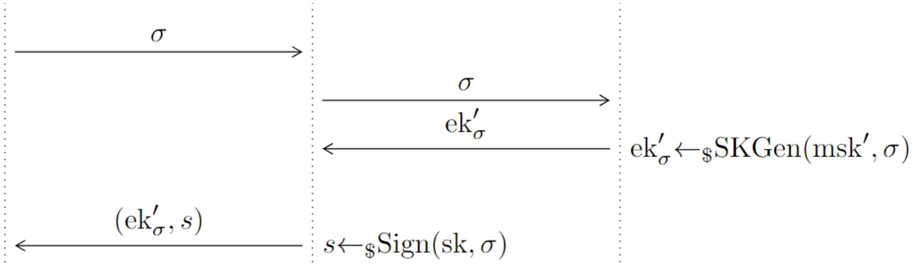
CCA Privacy Proof (1/5)

We show a proof by reduction to CPA Privacy. This is how our attacker setups the environment.


$\mathcal{A}$
$\mathcal{A}^*$
$\mathcal{C}$
mpk'
$(\mathrm{mpk}', \mathrm{msk}') \leftarrow_{\$} \mathrm{Setup}(1^\lambda)$
$(\mathrm{mpk}', \mathrm{pk}, \omega)$
$(\omega, \zeta, \xi) \leftarrow_{\$} \mathrm{K}_0(1^\lambda)$
$(\mathrm{sk}, \mathrm{pk}) \leftarrow_{\$} \mathrm{KGen}(1^\lambda)$

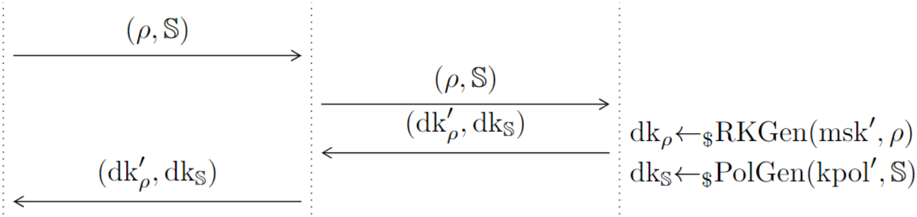
CCA Privacy Proof (2/5)

The encryption keys are enhanced with valid signatures produced by the attacker.


σ
σ
ek'_σ
$\mathrm{ek}'_\sigma \leftarrow_{\$} \mathrm{SKGen}(\mathrm{msk}', \sigma)$
(ek'_σ, s)
$s \leftarrow_{\$} \mathrm{Sign}(\mathrm{sk}, \sigma)$

CCA Privacy Proof (3/5)

Decryption keys remain the same. No need to modify them.



$(\rho, \mathbb{S})$

$(\rho, \mathbb{S})$

$(\mathrm{dk}'_\rho, \mathrm{dk}_\mathbb{S})$

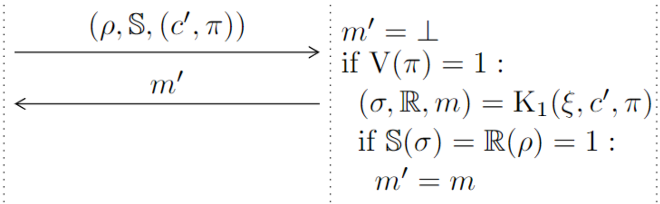
$\mathrm{dk}_\rho \leftarrow_\$ \mathrm{RKGen}(\mathrm{msk}', \rho)$

$\mathrm{dk}_\mathbb{S} \leftarrow_\$ \mathrm{PolGen}(\mathrm{kpol}', \mathbb{S})$

$(\mathrm{dk}'_\rho, \mathrm{dk}_\mathbb{S})$

CCA Privacy Proof (4/5)

The decryption queries are simulated thanks to true-simulation f-extractability of the NIZK.



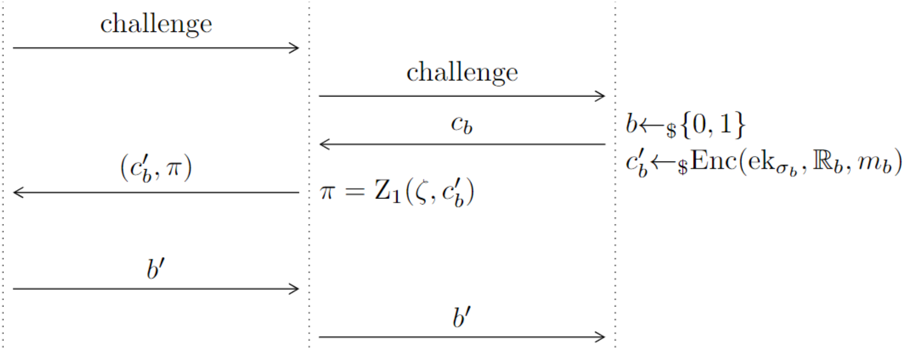
$(\rho, \mathbb{S}, (c', \pi))$

m'

```
$m' = \bot$
if $\mathrm{V}(\pi) = 1$ :
  $(\sigma, \mathbb{R}, m) = \mathrm{K}_1(\xi, c', \pi)$
  if $\mathbb{S}(\sigma) = \mathbb{R}(\rho) = 1$ :
    $m' = m$
```

CCA Privacy Proof (5/5)

Finally, the challenge is enhanced with a simulated proof. Then the attacker returns the same bit.


challenge
challenge
c_b
$b \leftarrow_{\$} \{0, 1\}$
(c'_b, π)
$c'_b \leftarrow_{\$} \mathrm{Enc}(\mathrm{ek}_{\sigma_b}, \mathbb{R}_b, m_b)$
$\pi = \mathrm{Z}_1(\zeta, c'_b)$
b'
b'