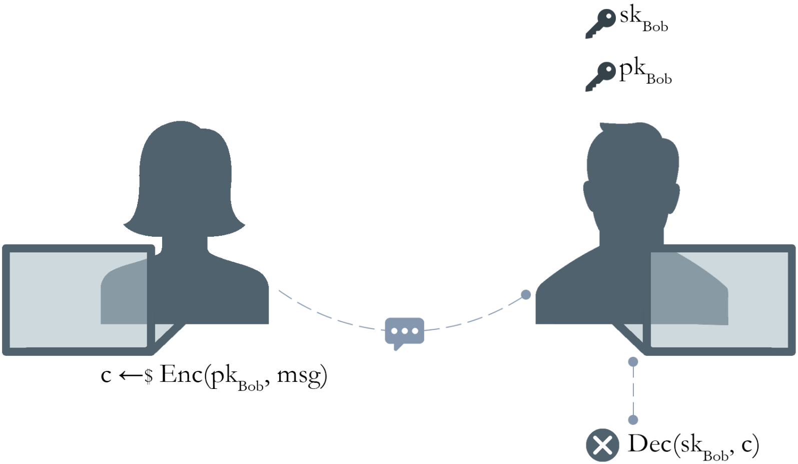
Matchmaking Encryption against Chosen-Ciphertext Attacks

Luigi Russo

September 30, 2020

PKE: an example



sk_{Bob}

pk_{Bob}

$c \leftarrow_{\$} \mathrm{Enc}(pk_{Bob}, msg)$

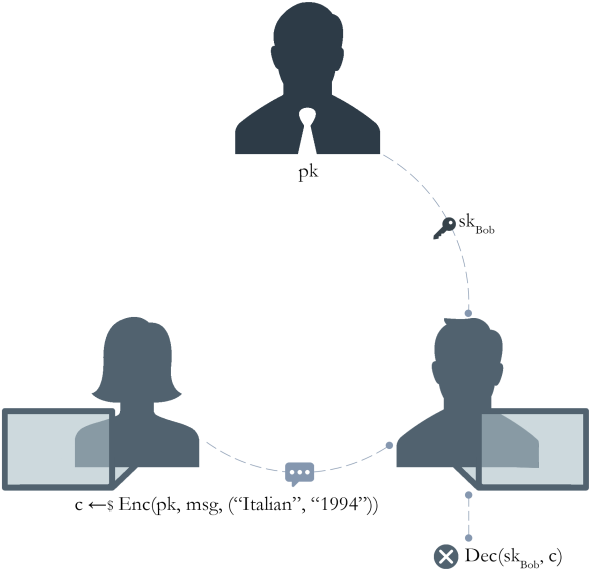
$\mathrm{Dec}(sk_{Bob}, c)$

Attribute-Based Encryption

A type of PKE 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
- the keys are distributed by a trusted party

ABE: an example



pk

sk_{Bob}

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

☒ $\mathrm{Dec}(sk_{Bob}, c)$

Matchmaking Encryption

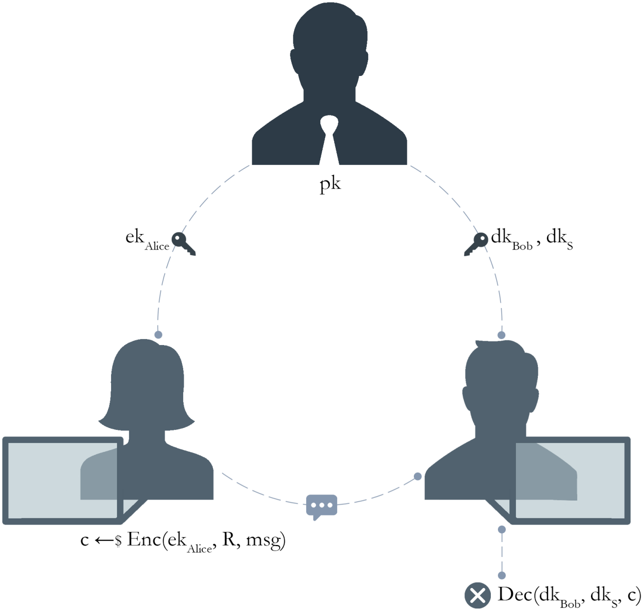
Main requirements

- 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 $\mathsf{C} : \{0, 1\}^* \to \{0, 1\}$
- If both the policies are matched, i.e. $\mathsf{S}(\sigma) = \mathsf{R}(\rho) = 1$, then the decryption is successful. Otherwise <u>nothing</u> is leaked.
- A trusted party stores the secrets

Applications

- Social matchmaking
- Encrypting bids and votes
- ...

ME: an example



pk

ek_{Alice}

dk_{Bob} , dk_S

$c \leftarrow_{\$} \text{Enc}(ek_{Alice}, R, msg)$

☒ $\text{Dec}(dk_{Bob}, dk_S, c)$

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

Matchmaking Encryption did not consider CCA.

Thesis Goals

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

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

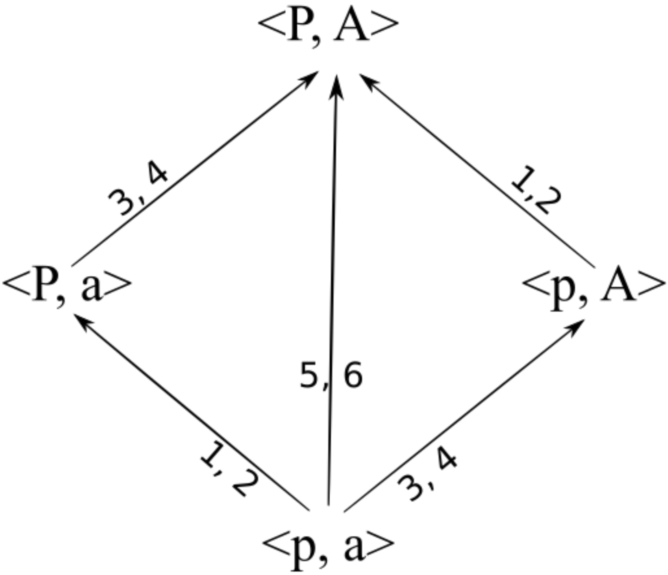
CPA setting

- 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



Digital Signatures

A digital signature $\Pi = (\mathsf{KGen}, \mathit{Sign}, \mathit{Ver})$ 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**)

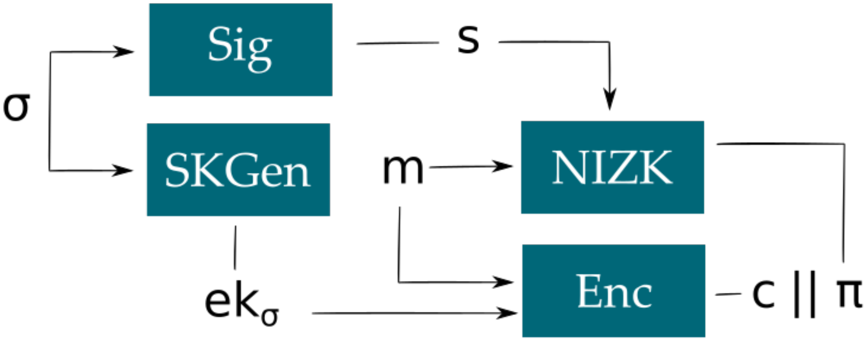
Non-Interactive Zero-Knowledge

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

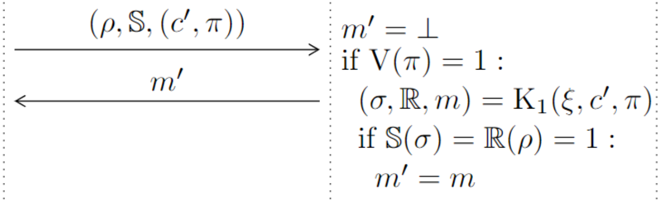


Sketch Proof

- 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.



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

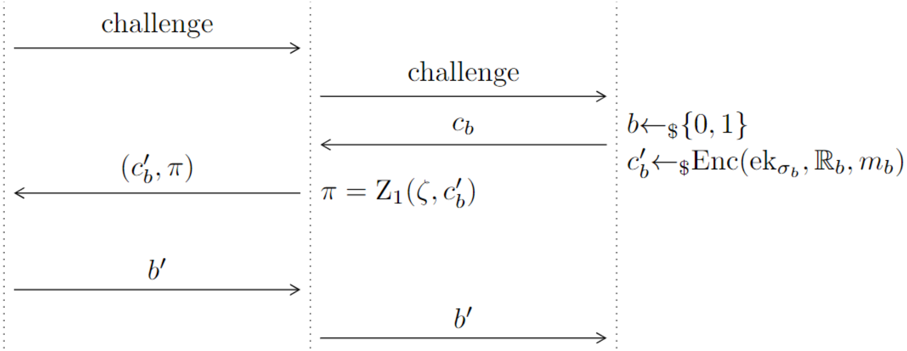
m'

```
$m' = \perp$
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



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'

Open problems

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

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

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

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 $\mathsf{S}(\sigma) = \mathsf{R}(\rho) = 1$:

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

CCA Privacy Game

Captures secrecy of sender's inputs.

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

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

CCA Authenticity Game

Captures security against malicious senders.

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

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

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

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

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

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!

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 $m \notin \mathcal{Q}_{\mathsf{Sign}}$, and $\mathsf{Ver}(pk, m, s) = 1$, output 1, else 0.

NIZK: a formal definition

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

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

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

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

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

3. **Zero-Knowledge.** $\exists (Z_0, Z_1)$ s.t. $\forall y \in L :$

$$\{\omega, Z_1(\zeta, y) : (\omega, \zeta) \leftarrow Z_0(1^\lambda)\}$$

$$\approx_c$$

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

True-Simulation Extractability

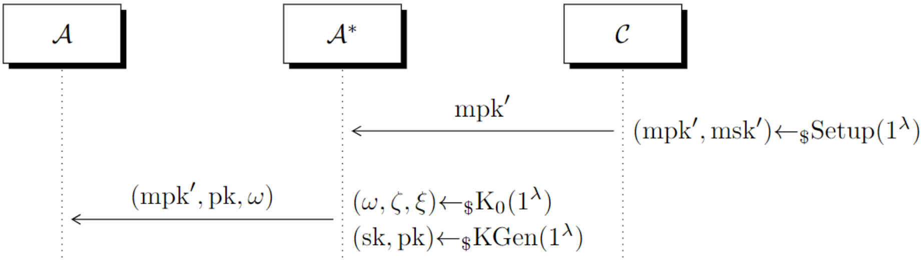
A NIZK Π for a relation R satisfies true-simulation f-extractability for a function f if $\exists$ PPT extractor $K = (K_0, K_1)$ s. t. $\forall \mathcal{A}$:

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

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

CCA Privacy Proof (1/5)

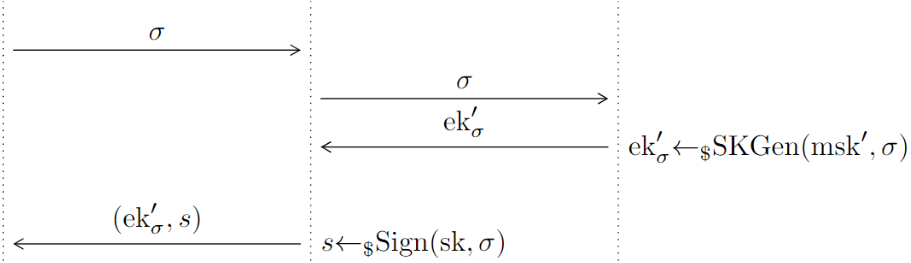
We show a proof by reduction to CPA Privacy. This is how our attacker setups the environment.



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

CCA Privacy Proof (2/5)

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



σ

σ

ek'_σ

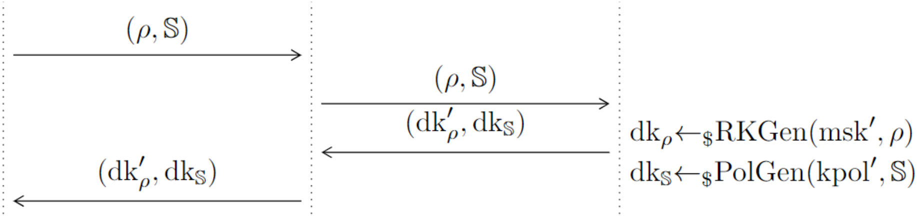
$\mathrm{ek}'_\sigma \leftarrow_\$ \mathrm{SKGen}(\mathrm{msk}', \sigma)$

(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.



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

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

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

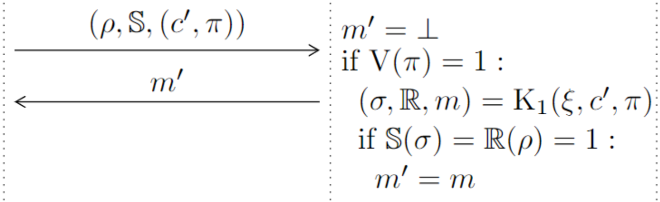
$\mathrm{dk}_\rho \leftarrow_\$ \mathrm{RKGen}(\mathrm{msk}', \rho)$

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

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

CCA Privacy Proof (4/5)

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



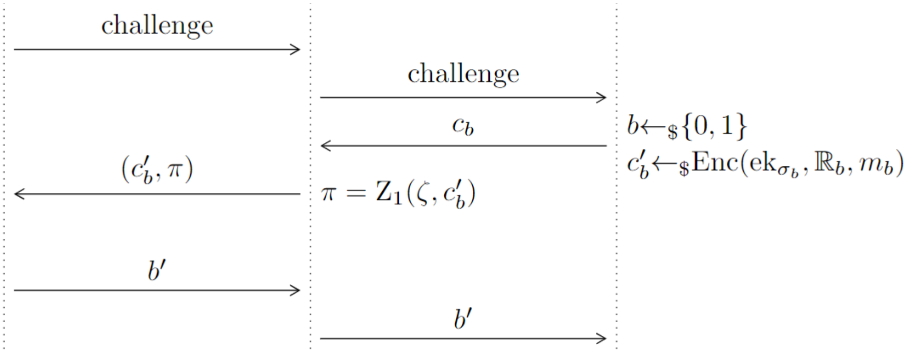
$\xrightarrow{(\rho, \mathbb{S}, (c', \pi))}$

$\xleftarrow{m'}$

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

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

$b \leftarrow_{\$} \{0,1\}$

$c'_b \leftarrow_{\$} \mathrm{Enc}(\mathrm{ek}_{\sigma_b}, \mathbb{R}_b, m_b)$

(c'_b, π)

$\pi = \mathrm{Z}_1(\zeta, c'_b)$

b'

b'