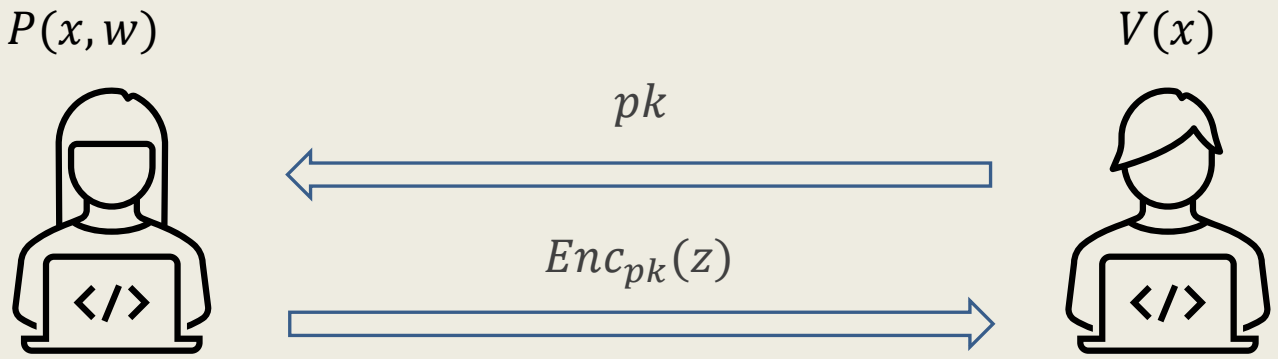
Knowledge Encryption and Its Applications to Simulatable Protocols With Low Round-Complexity

Yi Deng[1,2] and Xinxuan Zhang[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences

Motivation:

Consider the following (weakly) ZK protocol:



$z \leftarrow P_{WI}$(it knows sk ∨ other statement)

(Weakly) Zero-knowledge:

- If verifier can't distinguish/decrypt ciphertexts, simulator encrypts dummy message.
- If verifier can distinguish/decrypt ciphertexts, it *should* know the secret key. Simulator uses sk to generate the WI proofs.

Distinguishing/decryption ciphertexts ⟺ Knowledge of secret key ?

Motivation:

Consider Witness Encryption (WE) [GGSW13] and Conditional Disclosure Scheme (CDS) [AIR01]:

For any NP language L, $(x, w) \in R_L$

$(pk, sk) = (x, w)$
$c \leftarrow Enc(pk, m)$
$Dec(sk, c) = m$

$pk^* = (x, k^*)$

$pk = (x, k)$

$(pk, sk) \leftarrow Gen(1^\lambda, x, w)$
$c \leftarrow Enc(pk, m)$
$Dec(sk, c) = m$

Security: When $x \notin L$: $Enc(pk^*, m_0) \approx Enc(pk^*, m_1)$

Question: What happens when $x \in L$?

In other words, the decryption algorithm provides only a ***sound*** proof that the corresponding public key is valid.

Motivation:

Can we construct a public key encryption for which:

Only algorithms that knows the secret key (witness) can decrypt ciphertexts;

In other words, the decryption algorithm provides a ***proof of knowledge*** of the secret key?

A variant of Rabin's encryption [Deng 20].

This conception is useful in constructing (T, ϵ)-security protocol, e.g., selective opening secure commitment and concurrent zero-knowledge protocol. [Deng 20]

Knowledge Encryption (KE)

Knowledge encryption is associated with an NP language L. For any $(x, w) \in R_L$,

$pk = (x, k)$;
$sk = (w, r)$

$(pk, sk) \leftarrow Gen(1^\lambda, x, w)$
$c \leftarrow Enc(pk, m)$
$Dec(sk, c) = m$

We require this to hold even when pk is malicious generated.

Knowledge encryption satisfies the following properties:

- Witness extractability: only algorithm that knows the secret key (witness) can decrypt ciphertexts.

If $\mathcal{A}(pk)$ can decrypt/ distinguish ciphertexts

∃ efficient B, $B(\mathcal{A}, pk)$ outputs w (part of sk)

- Indistinguishability: Ciphertext indistinguishability holds for any $(x, w) \in R_L$
- Public key simulation: $\{pk \leftarrow KeySim(1^\lambda, x)\} \approx \{pk \leftarrow Gen(1^\lambda, x, w)\}$

Knowledge Encryption (KE)

Two constructions of knowledge encryption.

- Construction Ⅰ : based on CDS and RSR encryption.
- Construction Ⅱ : based on CDS only (2-round game-based secure OT) .

known based on DDH, QR, or LWE

Construction of CDS [AJ17]

Ingredients:

- A two-round OT $(OT1, OT2)$ with game-based security, and,

- A garbling circuit scheme $GC = (Garble, Eval)$ for circuit:

$$C(x,w,m) = \begin{cases} m & if\ (x,w) \in R_L \\ \perp & if\ (x,w) \notin R_L \end{cases}$$

Construction:

$Gen(x,w)$:

Run $\mathrm{OT}_1 \leftarrow \left(OT_1(w_1), \cdots, OT_1(w_l)\right)$,

Output: $pk = (x, OT_1)$; $sk = w$ and randomness

$Enc(pk,m)$:

$\left(\hat{C}, \{k_{i,b}^x\}, \{k_{i,b}^w\}, \{k_{i,b}^m\}\right) \leftarrow Garble\left(1^\lambda, C\right)$,

$OT_2 \leftarrow \left(OT_2\left(k_{1,0}^w, k_{1,1}^w\right), \cdots, OT_2\left(k_{l,0}^w, k_{l,1}^w\right)\right)$

Output: $c = \left(\hat{C}, \{k_{i,x_i}^x\}, \{k_{i,m_b}^m\}, OT_2\right)$

$Dec(sk,c)$:

Retrieve $\{k_{i,w_i}^w\}$ from OT_2

Output: $m = Eval\left(\hat{C}, \{k_{i,x_i}^x\}, \{k_{i,m_b}^m\}, \{k_{i,w_i}^w\}\right)$

Construction of KE

$$C(x,w,m)=\begin{cases} m & if\ (x,w)\in R_L \\ \perp & if\ (x,w)\notin R_L \end{cases}$$

$$C(x,w,y,m)=\begin{cases} m & if\ (x,w)\in R_L\ and\ y=0^l \\ \Sigma_{i=1}^{l} w_i y_i & if (x,w)\in R_L\ and\ \|y\|_1\geq 1 \\ \perp & if\ (x,w)\notin R_L \end{cases}$$

Construction($|m|=1$):

$Gen(x,w)$: $\mathrm{OT}_1 \leftarrow \left(OT_1(w_1),\cdots,OT_1(w_l)\right)$. Output: $pk=(x,OT_1)$; $sk=w$ and randomness

$Enc(pk,m)$: $\left(\hat{C},\{k_{i,b}^x\},\{k_{i,b}^w\},\boxed{\{k_{i,b}^y\}},\{k_{i,b}^m\}\right)\leftarrow Garble\left(1^\lambda,C\right)$,

$OT_2 \leftarrow \left(OT_2\left(k_{1,0}^w,k_{1,1}^w\right),\cdots,OT_2\left(k_{l,0}^w,k_{l,1}^w\right)\right), y\leftarrow 0^l$

Output: $c=\left(\hat{C},\{k_{i,x_i}^x\},\boxed{\{k_{i,0}^y\}},\{k_{i,m_b}^m\},OT_2\right)$

$Dec(sk,c)$: Retrieve $\{k_{i,w_i}^w\}$ from OT_2

Output: $m=Eval\left(\hat{C},\{k_{i,x_i}^x\},\{k_{i,m_b}^m\},\boxed{\{k_{i,0}^y\}},\{k_{i,w_i}^w\}\right)$

Construction of KE

$$C(x,w,m)=\begin{cases} m & if\ (x,w)\in R_L \\ \perp & if\ (x,w)\notin R_L \end{cases}$$

$$C(x,w,y,m)=\begin{cases} m & if\ (x,w)\in R_L\ and\ y=0^l \\ \Sigma_{i=1}^{l} w_i y_i & if (x,w)\in R_L\ and\ \|y\|_1\geq 1 \\ \perp & if\ (x,w)\notin R_L \end{cases}$$

Construction($|m|=1$):

$Gen(x,w)$: $\mathrm{OT}_1 \leftarrow \left(OT_1(w_1),\cdots,OT_1(w_l)\right)$. Output: $pk=(x,OT_1)$; $sk=w$ and randomness

Witness Extractability
Indistinguishability
Public key simulation (CDS)

$\left\{k_{i,b}^{w}\right\},\left\{k_{i,b}^{y}\right\},\left\{k_{i,b}^{m}\right\}\right) \leftarrow Garble\left(1^{\lambda},C\right)$,

$_2\left(k_{1,0}^{w},k_{1,1}^{w}\right),\cdots,OT_2\left(k_{l,0}^{w},k_{l,1}^{w}\right)\right), y\leftarrow 0^l$

$= \left(\hat{C},\left\{k_{i,x_i}^{x}\right\},\left\{k_{i,0}^{y}\right\},\left\{k_{i,m_b}^{m}\right\},OT_2\right)$

$Enc_j^*(pk,m)$: $\left(\hat{C},\left\{k_{i,b}^{x}\right\},\left\{k_{i,b}^{w}\right\},\left\{k_{i,b}^{y}\right\},\left\{k_{i,b}^{m}\right\}\right) \leftarrow Garble\left(1^{\lambda},C\right)$,

$OT_2 \leftarrow \left(OT_2\left(k_{1,0}^{w},k_{1,1}^{w}\right),\cdots,OT_2\left(k_{l,0}^{w},k_{l,1}^{w}\right)\right), y\leftarrow 0\cdots 010\cdots 0$

Output: $c=\left(\hat{C},\left\{k_{i,x_i}^{x}\right\},\left\{k_{i,y_i}^{y}\right\},\left\{k_{i,m_b}^{m}\right\},OT_2\right)$

j-th position

Key Observation:

$Enc_j^*(pk,m)\approx Enc\left(pk,w_j\right)$

One can extract w_j by checking:

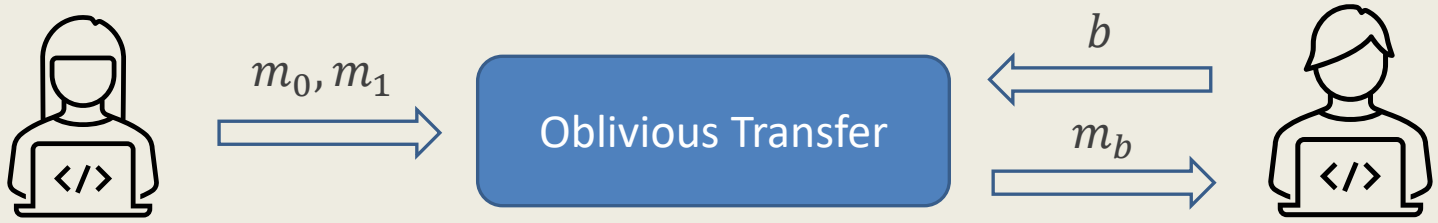
In the view of $\mathcal{A}$, $Enc_j^*(pk,m)$ is close to $Enc(pk,0)$ or $Enc(pk,1)$

Applications:

- The *first* 3-round (T, ϵ)-simulatable OT
  - Fully simulatable security for the receiver
  - (T, ϵ)-simulatable security for the sender
- A variety of protocols achieving (T, ϵ) -simulatable security (outperform prior works in either relying assumption or security)
  - 3-round delayed-input (T, ϵ)-ZK argument
  - 3-round (T, ϵ)-secure two-party computation for independent-input functionalities
  - 3-round concurrent (T, ϵ)-ZK in the BPK model
  - 2-round selective opening (T, ϵ)-secure commitment

All based on 2-round game-based secure OT (which can be based on DDH, QR, or LWE)

Applications: the first 3-round (T, ϵ)-simulatable OT



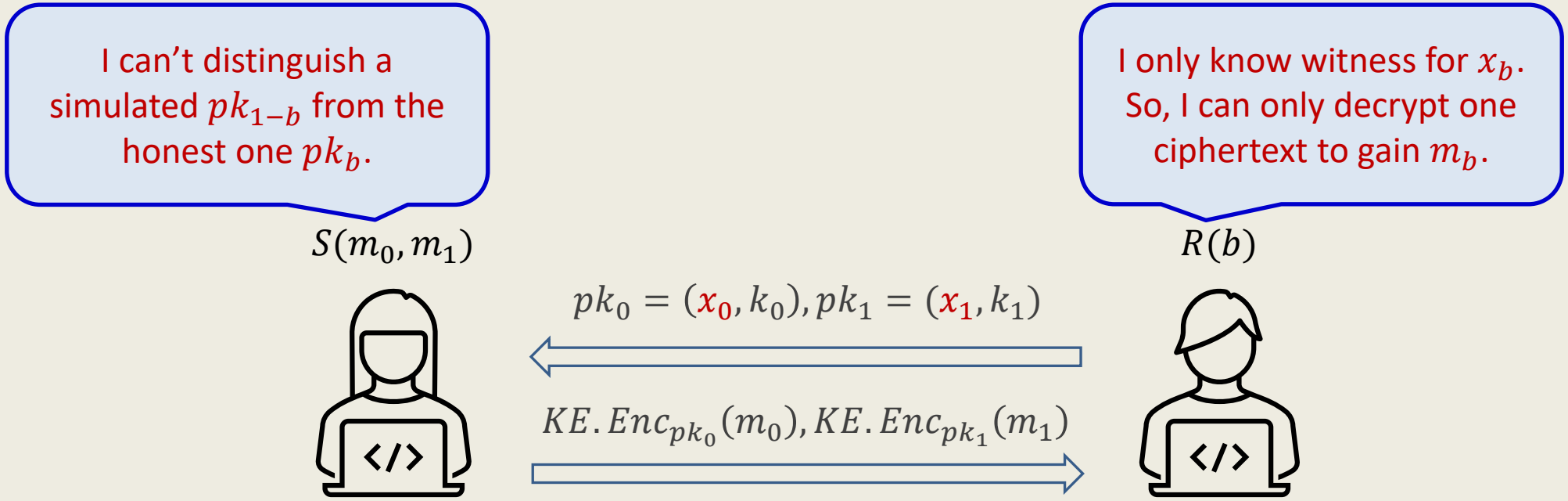
- Fully simulatable security for the receiver
- (T, ϵ)-simulatable security for the sender

Technique：

- Win-Win Construction
- Individual Reductions/Simulations [Deng 20]

Applications: the first 3-round (T, ϵ)-simulatable OT

Suppose the receiver knows a witness for *only one* of the instances x_0 and x_1



Challenges:

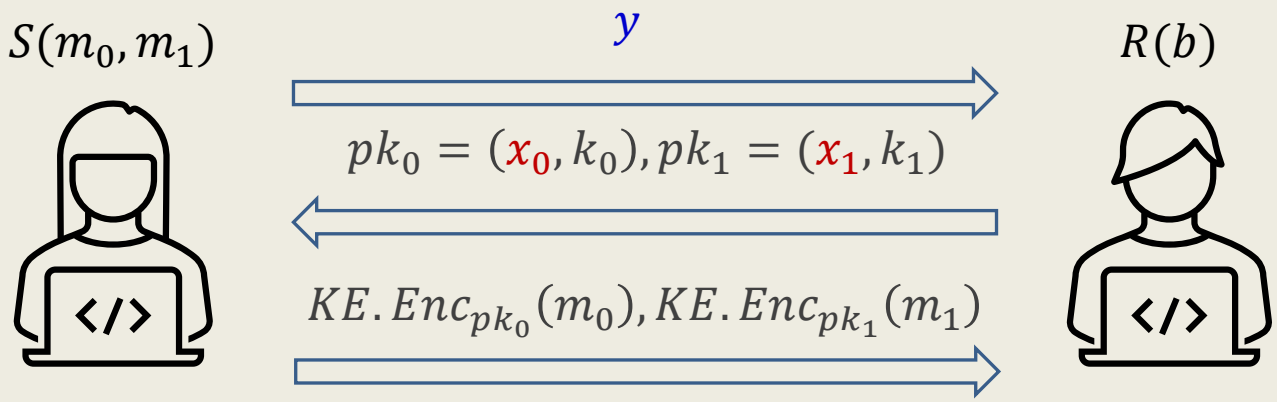
- We need to make sure that the receiver knows a witness for only one of these two instances.
- Simulator needs to extract input strings (bit) from malicious sender (receiver).

Applications: the first 3-round (T, ϵ)-simulatable OT

Make sure that R knows witness for only one of $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$.

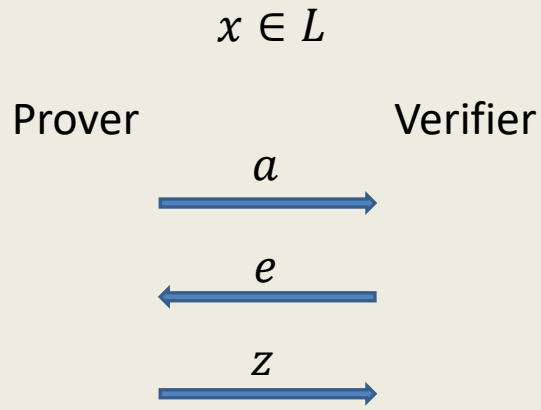
Let S generate a *hard* NP instance y!

Using Σ-Protocol to generate these two instances $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$.

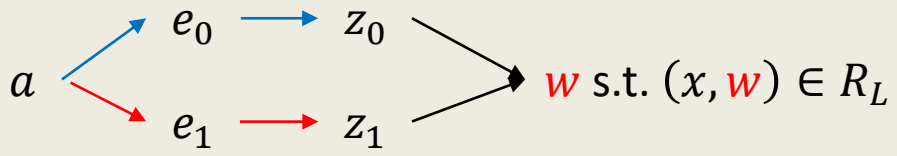


Applications: the first 3-round (T, ϵ)-simulatable OT

Σ-Protocol: A three-round public-coin protocol satisfies that:



- Special Soundness:



- Special Honest-Verifier ZK:

  Given challenge e, easy to compute an accepting transcript (a, e, z), which is indistinguishable from an honest one.

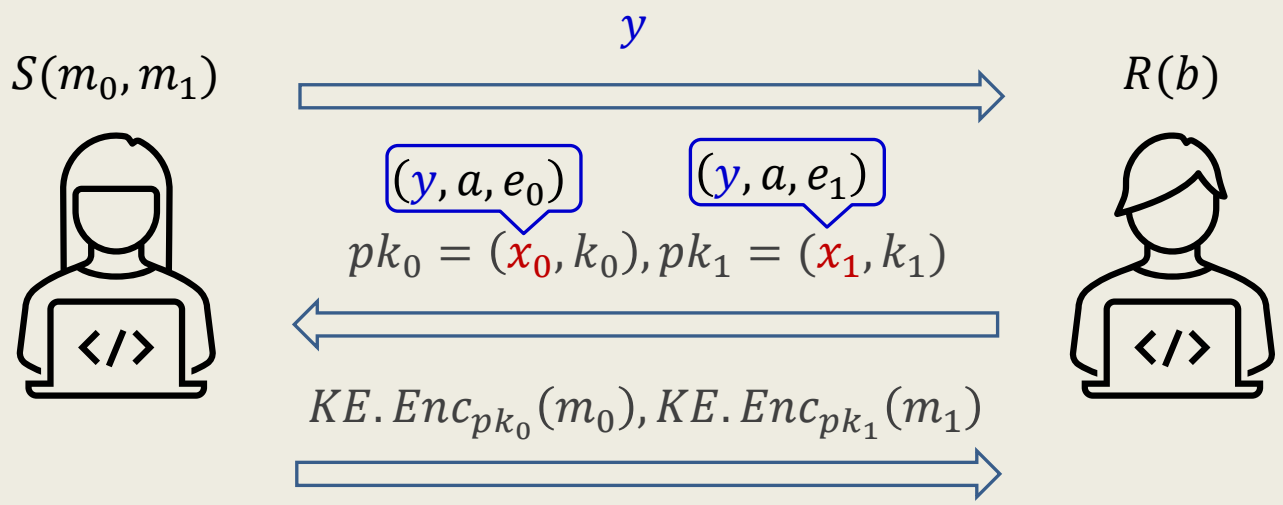
Applications: the first 3-round (T, ϵ)-simulatable OT

Let S generate a *hard* NP instance y.

Receiver chooses e_b randomly and obtain an accepting proof (a, e_b, z_b) for y

Choose e_{1-b} randomly, set: $x_0 = (y, a, e_0)$; $x_1 = (y, a, e_1)$

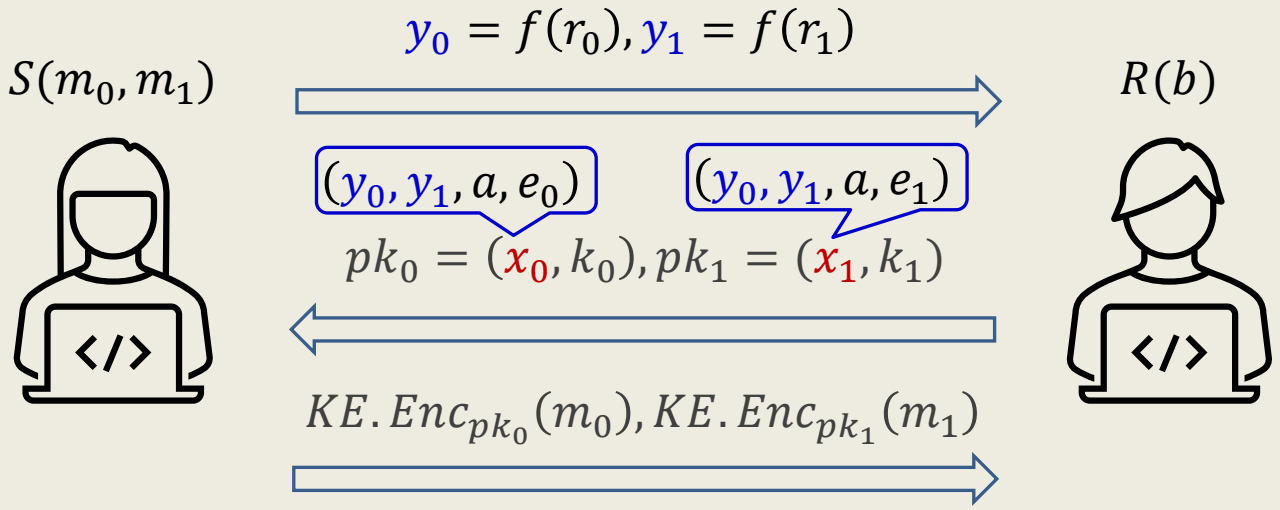
Define $x = (y, a, e) \in L$ if $\exists\ z$ s.t. (a, e, z) is an accepting proof for y



Applications: the first 3-round (T, ϵ)-simulatable OT

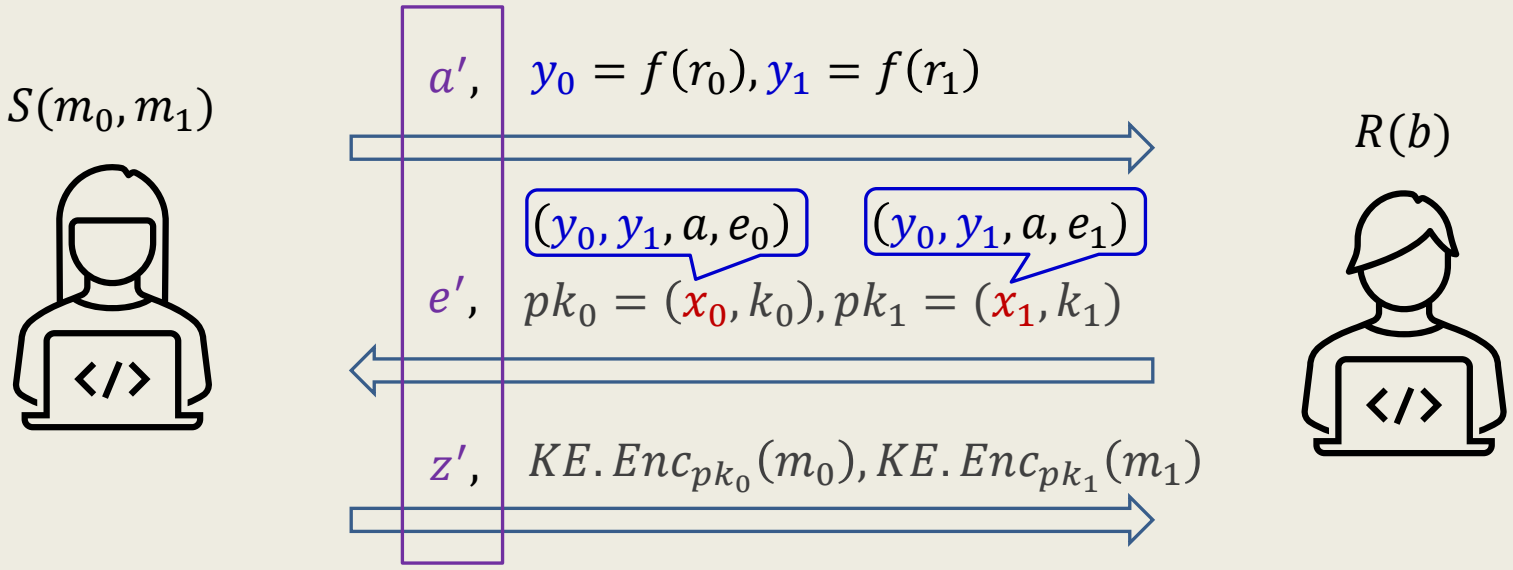
How to choose y ? $y := (y_0, y_1)$ where $y_0 = f(r_0), y_1 = f(r_1)$, f is a OWF.

y is said to be a YES instance if there exists r such that $f(r) = y_0$ ***or*** $f(r) = y_1$.

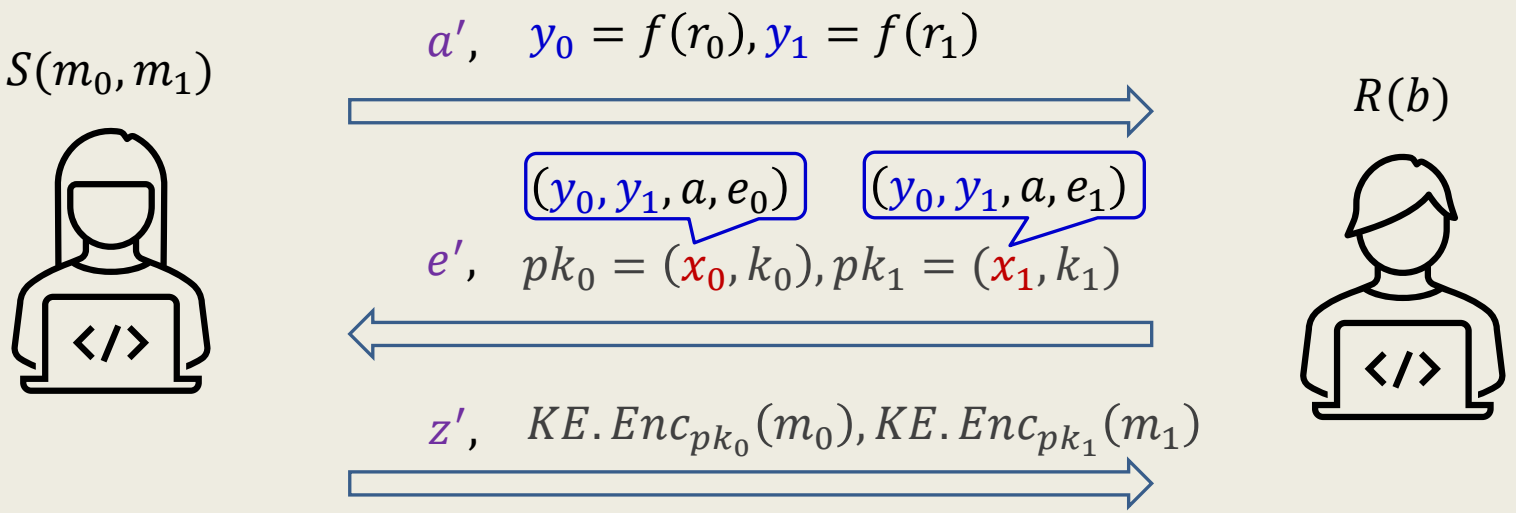


Applications: the first 3-round (T, ϵ)-simulatable OT

Finally, S proves to R that it knows one pre-image of y_0 or y_1 via a WI protocol satisfying special soundness.



Applications: the first 3-round (T, ϵ)-simulatable OT



Fully simulatable Security for the receiver R:

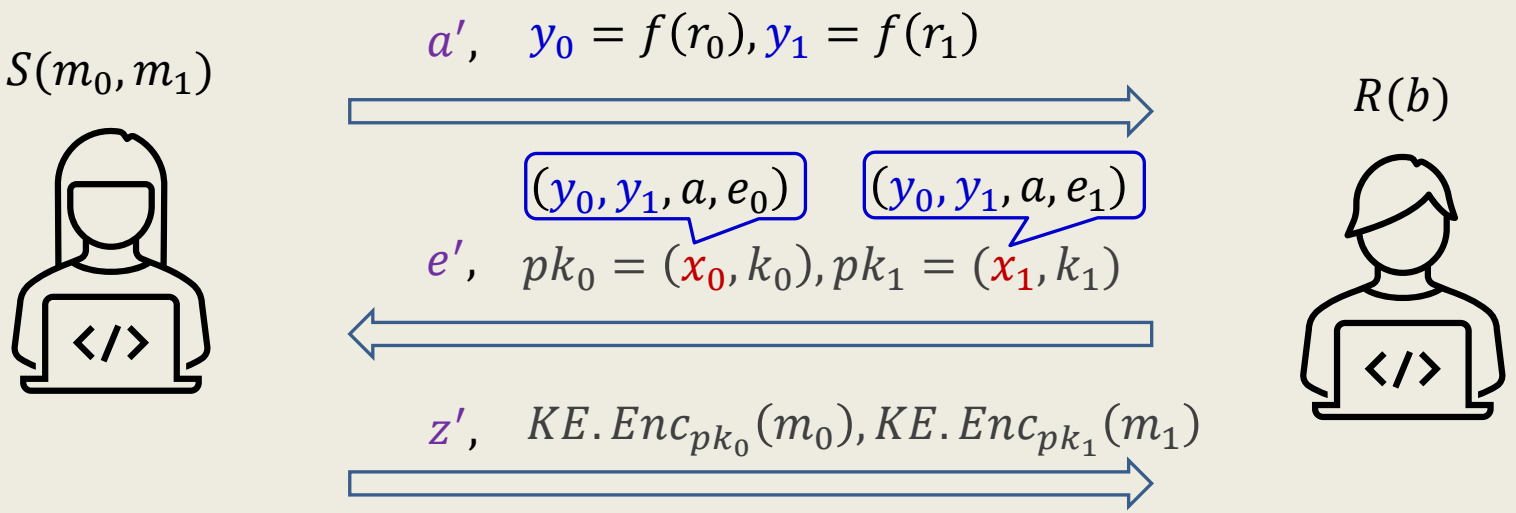
$Sim(S^*)$: 1. Extract r_0 or r_1 from (a', e', z') by rewinding.

2. Generate two YES instances x_0, x_1 and valid pk_0, pk_1.

3. Obtains m_0, m_1 by decrypting both ciphertexts and send them to the functionality.

That's a *non-traditional* extraction strategy.

Applications: the first 3-round (T, ϵ)-simulatable OT



(T, ϵ)-simulatable Security for the sender S:

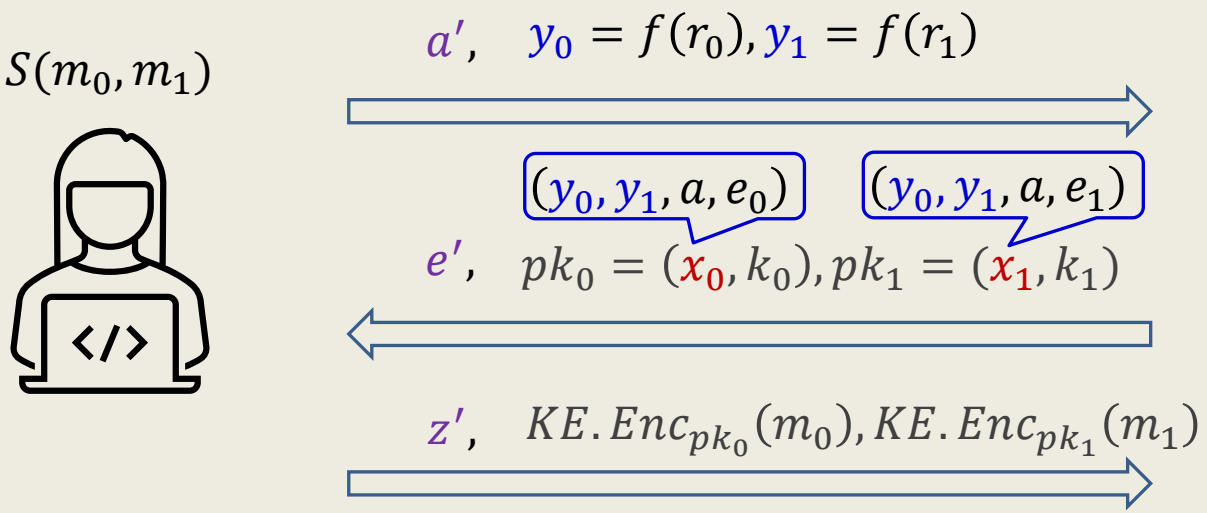
Sim need to extract b ($actually\ w_b$) from a malicious R^*

- Individual Reductions/Simulations [Deng 20]

  show there exists a good extractor E that outperforms all other algorithms in extracting secret keys

Lemma[Deng20]:
$\forall\ R^*\ \exists\ E\ \forall\ C$ of size T,
Pr [E fails but C succeeds] < ϵ.

Applications: the first 3-round (T, ϵ)-simulatable OT



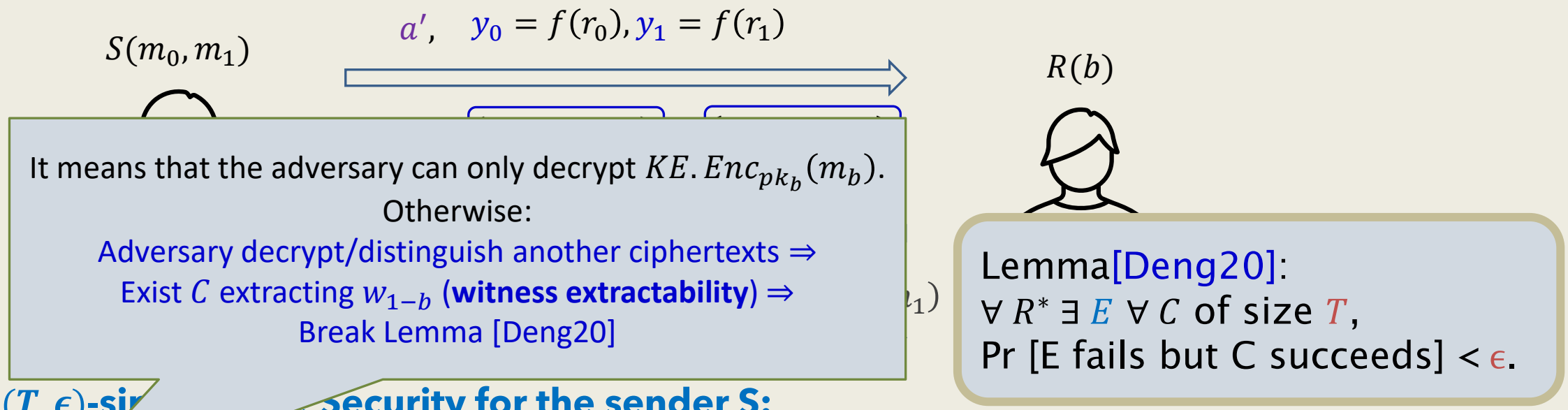
Lemma[Deng20]:
$\forall\, R^* \,\exists\, E \;\forall\, C$ of size T,
Pr [E fails but C succeeds] < ϵ.

(T, ϵ)-simulatable Security for the sender S:

$\forall\, R^* \,\exists\, E$, $Sim(E, R^*)$ *runs* E to extract w for x_0, x_1 from malicious R^*

- E succeeds to extract only one $w_{b'}$, then $b = b'$
- E fails to extract any witness, then $b = 0$,
- E succeeds to extract both witness, then aborts.

Applications: the first 3-round (T, ϵ)-simulatable OT

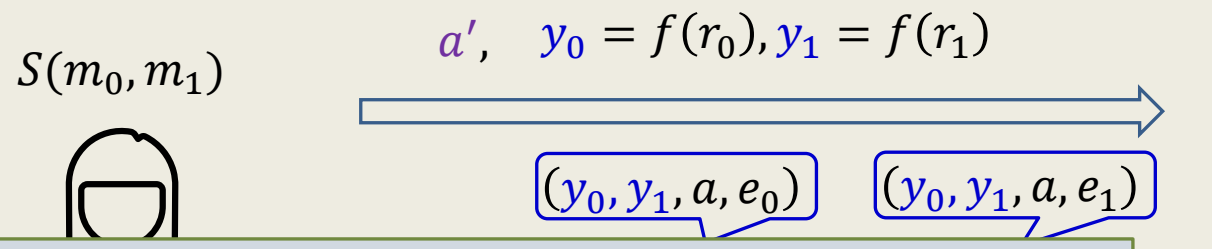


(T, ϵ)-si… Security for the sender S:

$\forall R^*$ …E, $Sim(E, R^*)$ *runs* E to extract w for x_0, x_1 from malicious R^*

- E succeeds to extract only one $w_{b'}$, then $b = b'$ WIN!
- E fails to extract any witness, then $b = 0$,
- E succeeds to extract both witness, then aborts.

Applications: the first 3-round (T, ϵ)-simulatable OT



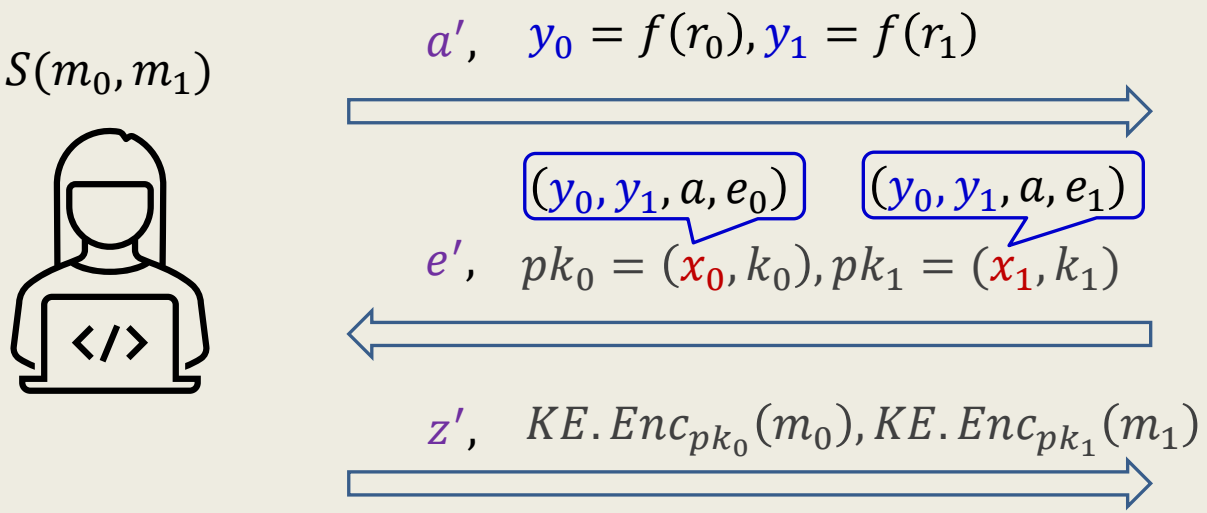
It means that the adversary cannot decrypt any ciphertexts.
Otherwise:
Adversary decrypt/distinguish one ciphertexts ⇒
Exist C extracting w (witness extractability) ⇒
Break Lemma [Deng20]

Lemma[Deng20]:
$\forall R^* \exists E \forall C$ of size T,
Pr [E fails but C succeeds] < ϵ.

$\forall R^* \exists$ [illegible] (E, R^*) *runs* E to extract w for x_0, x_1 from malicious R^*

- E [illegible]ceeds to extract only one $w_{b'}$, then $b = b'$ WIN!
- E fails to extract any witness, then $b = 0$, WIN!
- E succeeds to extract both witness, then aborts.

Applications: the first 3-round (T, ϵ)-simulatable OT



Lemma[Deng20]:
$\forall\, R^* \;\exists\, E \;\forall\, C$ of size T,
Pr [E fails but C succeeds] < ϵ.

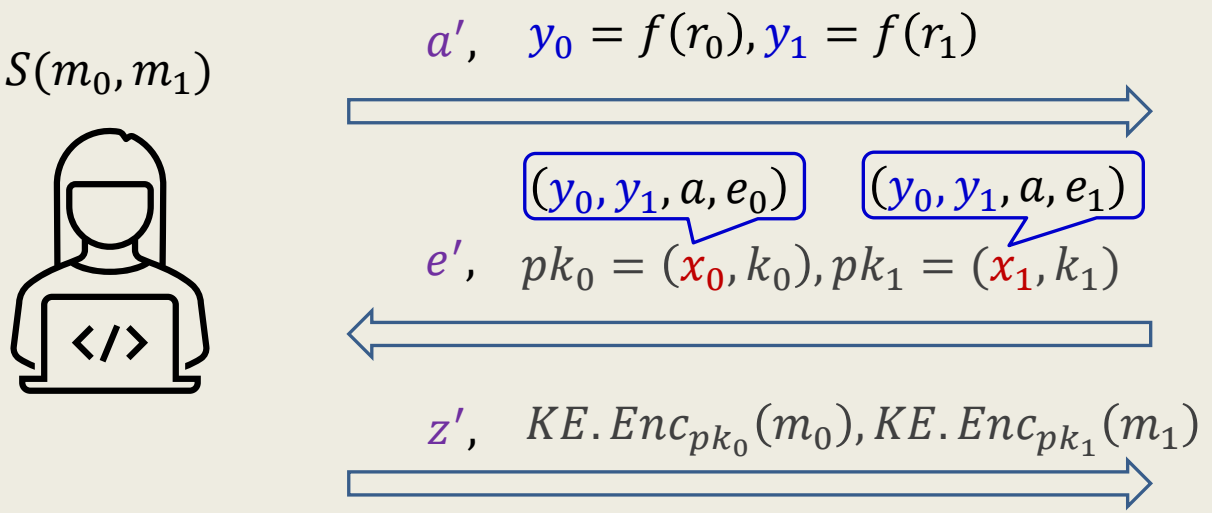
(T, ϵ)-simulatable Security for the sender S:

One could extract r_0 or r_1 from (x_0, w_0) and (x_1, w_1) , which is impossible from that (a', e', z') is witness hiding.

m malicious R^*

- E fa… extract any witness, then $b = 0$, — WIN!
- E succeeds to extract both witness, then aborts. — IMPOSSIBLE!

Applications: the first 3-round (T, ϵ)-simulatable OT



Lemma[Deng20]:
$\forall\, R^* \,\exists\, E \;\forall\, C$ of size T,
Pr [E fails but C succeeds] < ϵ.

(T, ϵ)-simulatable Security for the sender S:

$\forall\, R^* \,\exists\, E$, $Sim(E, R^*)$ *runs* E to extract w for x_0, x_1 from malicious R^*

- E succeeds to extract only one $w_{b'}$, then $b = b'$ — WIN!
- E fails to extract any witness, then $b = 0$, — WIN!
- E succeeds to extract both witness, then aborts. — IMPOSSIBLE!

WIN-WIN STRUCTURE

Thank you for your attention