A Recent Topic of Cryptography: How to Prove the Security of Cryptography

Tatsuaki Okamoto
NTT

Two Approaches of Proving Cryptographic Schemes and Protocols

1. Computational Approach

- Models an adversary as a probabilistic polynomial-time Turing machine (PPTM), and proves that the security against any PPTM. (Probability and resource bounded TM are introduced.)

  Blum-Micali'82, Yao'82, Goldwasser-Micali'82, ･･･
- Widely accepted in the cryptology community as standard security definitions.
- The security proofs are complicated in general and flawed proofs are often presented (not so easy to check the correctness of a proof).

2. Formal Method Approach

- Expresses a cryptographic scheme or protocol by symbols, and prove the security by logical reasoning and rewriting rules.

  Dolev-Yao'82, BAN Logic, ･･･
- A lot of research has been made in the formal method community, but has not been well accepted by the cryptology community due to the lack of the soundness to the computational approach security.
- The security proofs are clear and can be (partially) automated.

Computational Approach

How to Define the Security in the Computational Approach?

- Attack-Based Formulation： Formulates the security by a game between an attacker and a challenger. Recently a new methodology using a series of games, the game hopping technique (sequence of games), is being advanced.

  IND-CCA of public-key encryption, EUF-CMA of digital signatures, …

- Simulation-Based Formulation： Formulates the security by the gap between an actual scheme/protocol (real world) and a simulated scheme/protocol using an ideal functionality (ideal world). Provides a unified formulation for all cryptographic schemes/protocols.

  Universal composability (UC) framework by Canetti, …

Security Definition of Public-Key Encryption (PKE) : An Example of Attack-Based Formulation

Public-Key Encryption (PKE) (G, E, D)

- G: Key Generation Algorithm

$$1^n \longrightarrow \boxed{G} \longrightarrow (\ e, d\ )$$

n: Security Parameter　　Public-Key　Secret-Key

- E: Encryption Algorithm

$$M \longrightarrow \underset{\overset{\Uparrow}{e}}{\boxed{E}} \longrightarrow C$$

- D: Decryption Algorithm

$$C \longrightarrow \underset{\overset{\Uparrow}{d}}{\boxed{D}} \longrightarrow M$$

Security of PKE

- Goal
  - One-Wayness（OW） $c=E_{pk}(m) \xrightarrow{\text{hard}} m$
  - Indistinguishability（IND）

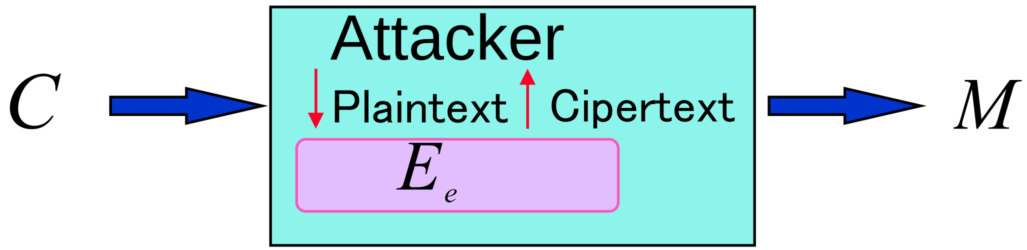
    no partial information of m is revealed from $c=E_{pk}(m)$
  - Non-malleability（NM）

    $$c=E_{pk}(m) \xrightarrow{\text{hard}} c'=E_{pk}(m')$$

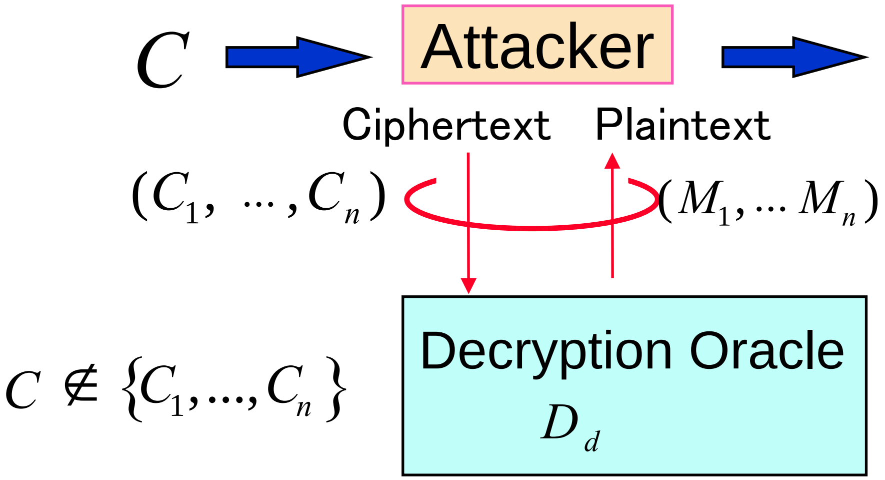
    $R(m, m')$ holds for a non-trivial relation R
- Attack
  - ○Passive attacks ･･･ Chosen-Plaintext Attacks（CPA）
  - ○Active attacks ･･･ Chosen-Ciphertext Attacks（CCA）

Attacks

- CPA：Chosen Plaintext Attack



- CCA：Chosen Ciphertext Attack

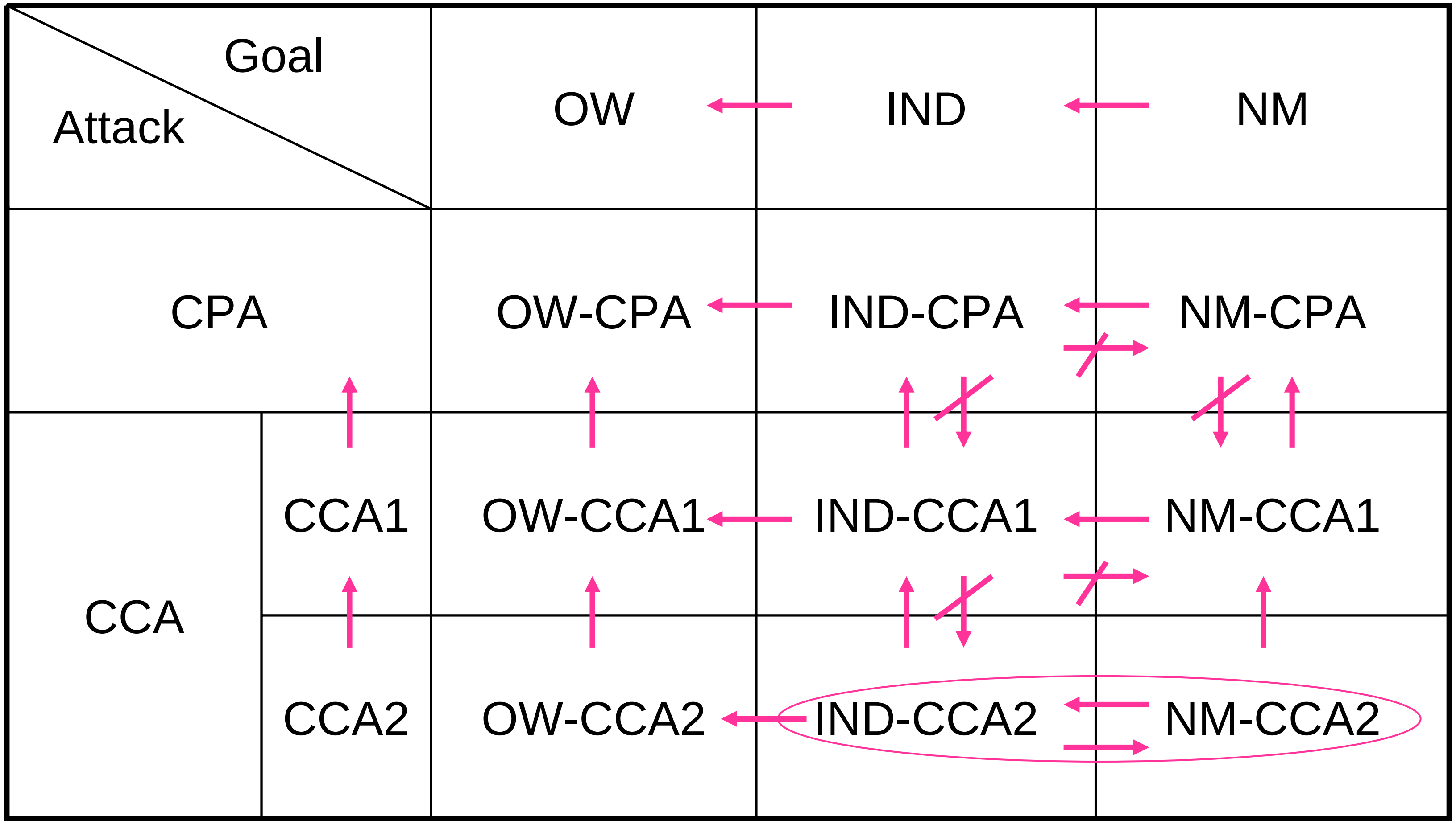


CCA1‥Only before given C, CCA is allowed..

CCA2‥Even after given C, CCA is allowed.

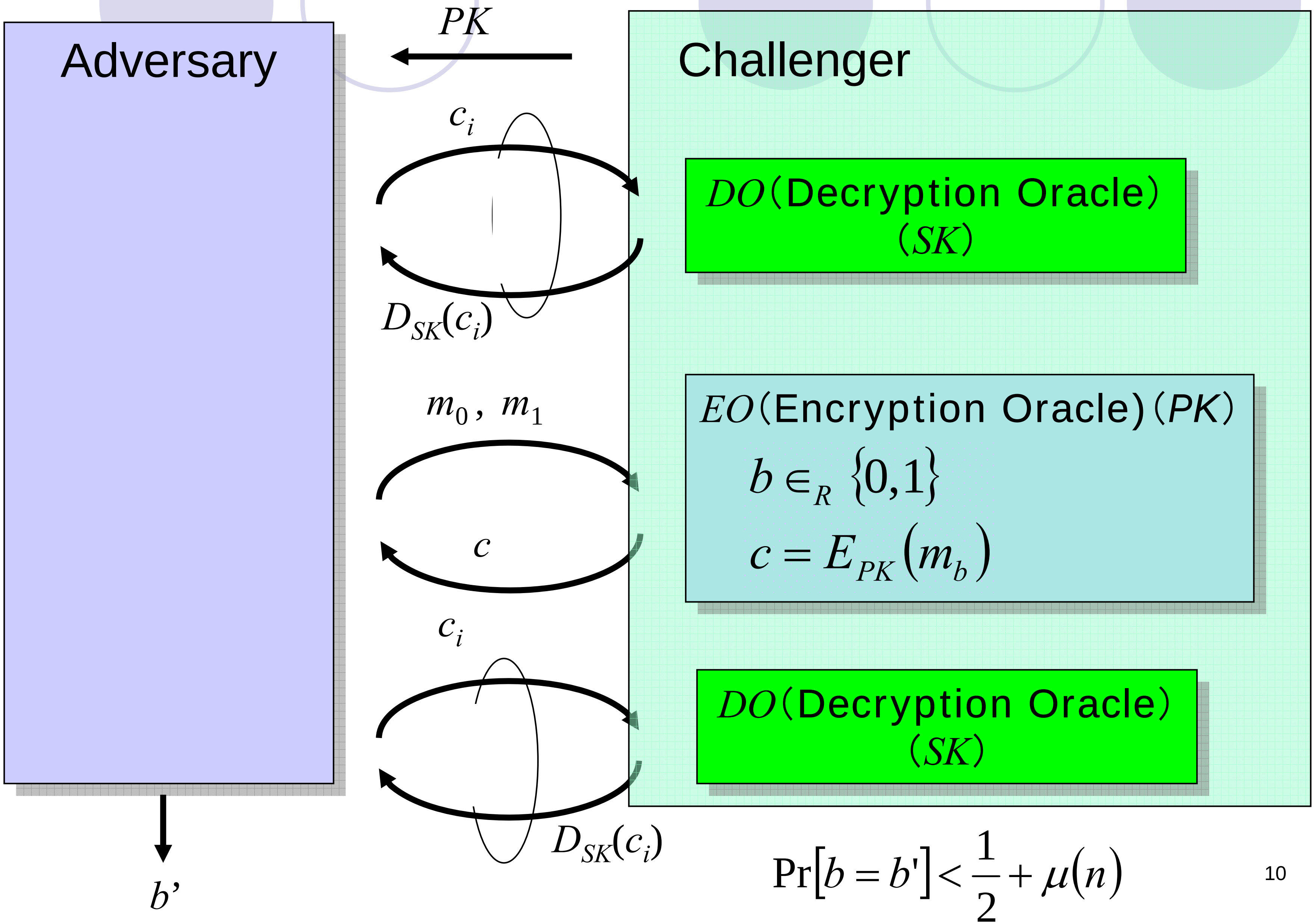
(CCA2 is a stronger attack than CCA1.)

Relationship among Security Definitions



Goal Attack		OW	IND	NM
CPA		OW-CPA	IND-CPA	NM-CPA
CCA	CCA1	OW-CCA1	IND-CCA1	NM-CCA1
	CCA2	OW-CCA2	IND-CCA2	NM-CCA2

IND-CCA2 Definition



Adversary

PK

Challenger

c_i

DO (Decryption Oracle) (SK)

$D_{SK}(c_i)$

m_0, m_1

EO (Encryption Oracle) (PK)

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

$c = E_{PK}(m_b)$

c

c_i

DO (Decryption Oracle) (SK)

$D_{SK}(c_i)$

b'

$$\Pr[b = b'] < \frac{1}{2} + \mu(n)$$

Security Proof of PKE: An Example of the Game Transformation Technique

Cramer-Shoup PKE Scheme

SK $\leftarrow (x_1, x_2, y_1, y_2, z_1, z_2) \in \mathbf{Z}_q{}^6$

$$e \leftarrow g^{x_1}\hat{g}^{x_2},\ f \leftarrow g^{y_1}\hat{g}^{y_2},\ h \leftarrow g^{z_1}\hat{g}^{z_2}$$

PK $\leftarrow (\Gamma[G, \hat{G},\ g,\ q], hk, \hat{g}, e, f, h)\quad (g \in G, \hat{g} \in \hat{G})$

Plaintext

$m \longrightarrow$

Encryption (PK)

$E1\!:\ u \leftarrow_{\cup} \mathbf{Z}_q$

$E2\!:\ a \leftarrow g^{u}$

$E3\!:\ \hat{a} \leftarrow \hat{g}^{u}$

$E4\!:\ c \leftarrow h^{u} \cdot m$

$E5\!:\ v \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}\ (a^*, \hat{a}^*, c^*)$

$E6\!:\ d \leftarrow e^{u} f^{uv}$

Ciphertext

$(a, \hat{a}, c, d)$

$\longrightarrow$

Decryption (SK)

D1: Check the form.

$\mathrm{D2}\!:\ a,\ \hat{a},\ c \overset{?}{\in} G$

$\mathrm{D3}\!:\ v \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}(a, \hat{a}, c)$

$\mathrm{D4}\!:\ d \overset{?}{=} a^{x_1 + y_1 v}\ \hat{a}^{x_2 + y_2 v}$

$\mathrm{D5}\!:\ m \leftarrow c \cdot (a^{z_1}\hat{a}^{z_2})^{-1}$

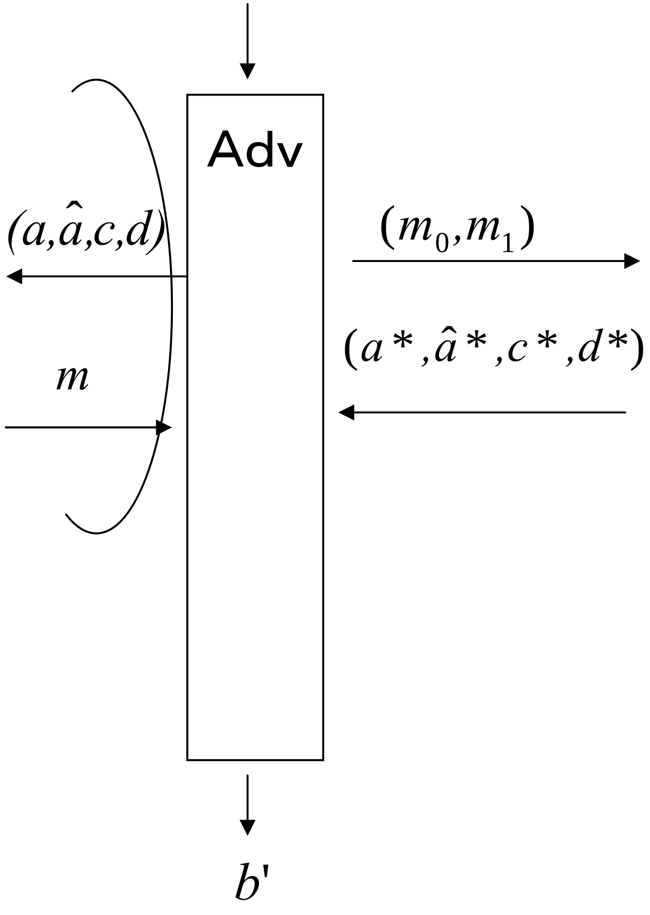
$\xrightarrow{m}$

The Security of Cramer-Shoup

The Cramer-Shoup PKE scheme is IND-CCA2 secure, assuming that the decisional Diffie-Hellman (DDH) problem is hard and that HF is a target collision resistant (TCR) function family.

Game0: IND-CCA2 Game

$(\Gamma[G, \hat{G}, g, q], hk, \hat{g}, e, f, h)$



Decryption Oracle

D1: Check the form.

D2: $a, \hat{a}, c \overset{?}{\in} G$

D3: $v \leftarrow \mathsf{HF}^{\lambda,\Gamma}_{hk}(a,\hat{a},c)$

D4: $d \overset{?}{=} a^{x_1+y_1 v}\hat{a}^{x_2+y_2 v}$

D5: $m \leftarrow c \cdot (a^{z_1}\hat{a}^{z_2})^{-1}$

SK $\leftarrow (x_1, x_2, y_1, y_2, z_1, z_2)$

$e \leftarrow g^{x_1}\hat{g}^{x_2}$, $f \leftarrow g^{y_1}\hat{g}^{y_2}$,

$h \leftarrow g^{z_1}\hat{g}^{z_2}$

Encryption Oracle

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

E1: $u \leftarrow_{\cup} \mathbf{Z}_q$

E2: $a^* \leftarrow g^u$

E3: $\hat{a}^* \leftarrow \hat{g}^u$

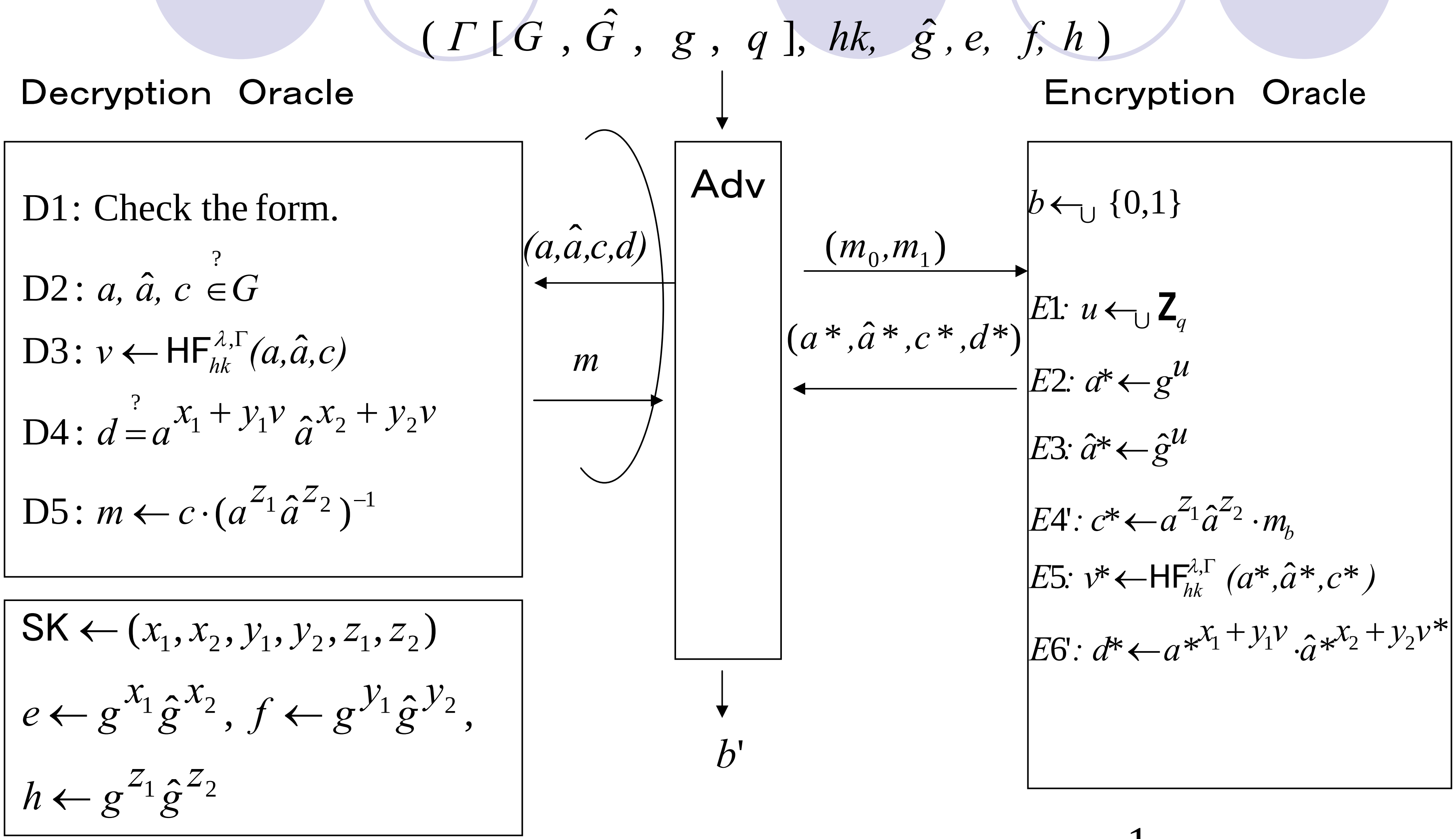
E4: $c^* \leftarrow h^u \cdot m_b$

E5: $v^* \leftarrow \mathsf{HF}^{\lambda,\Gamma}_{hk}(a^*,\hat{a}^*,c^*)$

E6: $d^* \leftarrow e^u f^{uv^*}$

$$\text{Advantage} = |\Pr[b = b'] - \frac{1}{2}|$$

Game1



$(\Gamma\,[\,G\,,\hat{G}\,,\ g\,,\ q\,],\ hk,\ \hat{g}\,, e,\ f,\ h\,)$

Decryption Oracle

D1: Check the form.

D2: $a,\ \hat{a},\ c \overset{?}{\in} G$

D3: $v \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}(a,\hat{a},c)$

D4: $d \overset{?}{=} a^{x_1+y_1 v}\,\hat{a}^{x_2+y_2 v}$

D5: $m \leftarrow c\cdot(a^{z_1}\hat{a}^{z_2})^{-1}$

SK $\leftarrow (x_1, x_2, y_1, y_2, z_1, z_2)$

$e \leftarrow g^{x_1}\hat{g}^{x_2},\ f \leftarrow g^{y_1}\hat{g}^{y_2},$

$h \leftarrow g^{z_1}\hat{g}^{z_2}$

$(a,\hat{a},c,d)$

m

Adv

b'

(m_0,m_1)

$(a^*,\hat{a}^*,c^*,d^*)$

Encryption Oracle

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

$E1$: $u \leftarrow_U \mathbf{Z}_q$

$E2$: $a^* \leftarrow g^u$

$E3$: $\hat{a}^* \leftarrow \hat{g}^u$

$E4'$: $c^* \leftarrow a^{z_1}\hat{a}^{z_2}\cdot m_b$

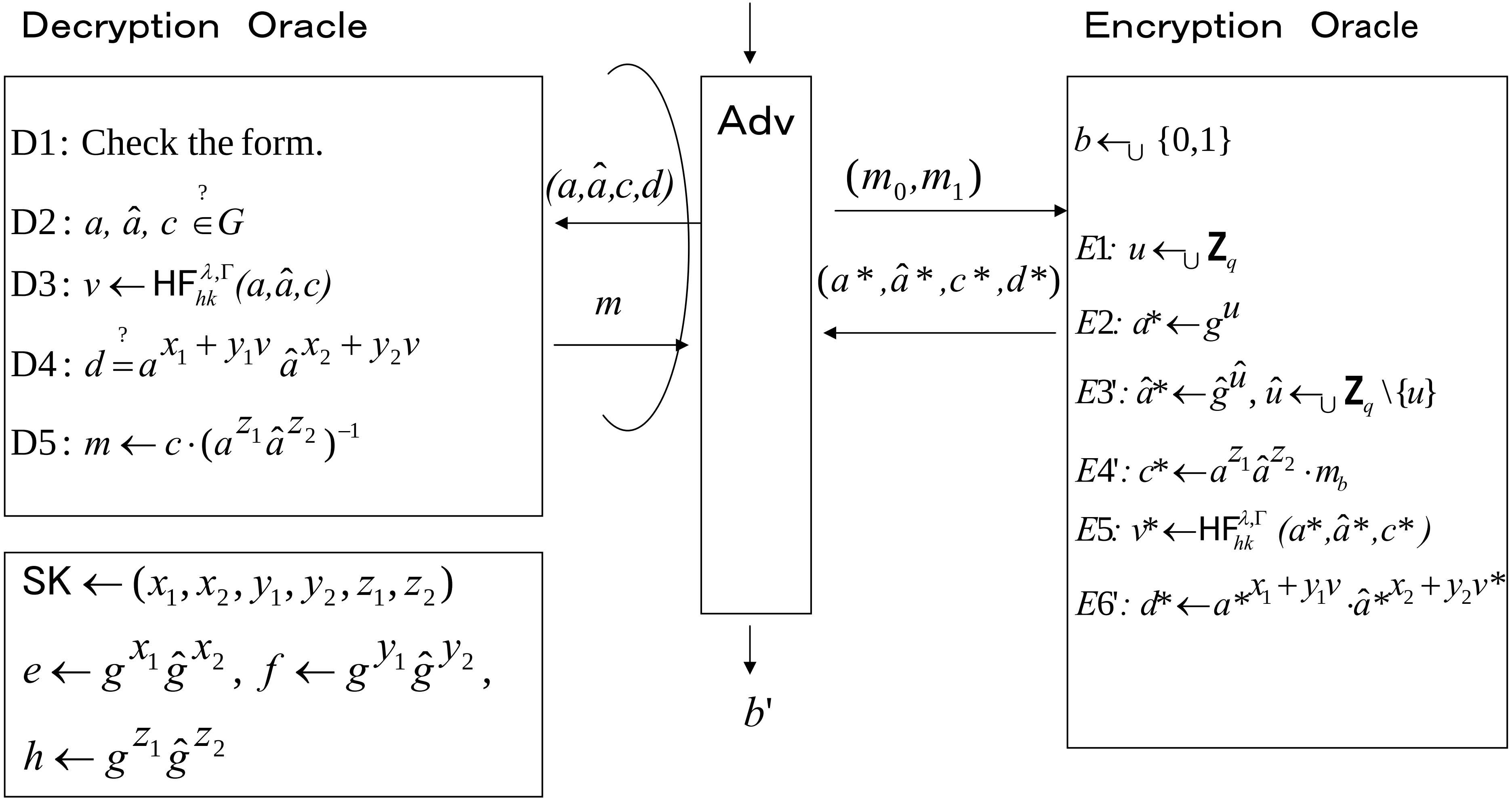
$E5$: $v^* \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}\ (a^*,\hat{a}^*,c^*)$

$E6'$: $d^* \leftarrow a^{*x_1+y_1 v}\cdot\hat{a}^{*x_2+y_2 v^*}$

$$\text{Advantage} = \left|\ \Pr[\,b = b'\,] - \frac{1}{2}\ \right|$$

Game2

$$(\Gamma\,[G, \hat{G}, g, q], hk, \hat{g}, e, f, h)$$



Decryption Oracle

D1: Check the form.

D2: $a,\ \hat{a},\ c \overset{?}{\in} G$

D3: $v \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}(a,\hat{a},c)$

D4: $d \overset{?}{=} a^{x_1 + y_1 v}\ \hat{a}^{x_2 + y_2 v}$

D5: $m \leftarrow c \cdot (a^{z_1}\hat{a}^{z_2})^{-1}$

$\mathsf{SK} \leftarrow (x_1, x_2, y_1, y_2, z_1, z_2)$

$e \leftarrow g^{x_1}\hat{g}^{x_2},\ f \leftarrow g^{y_1}\hat{g}^{y_2},$

$h \leftarrow g^{z_1}\hat{g}^{z_2}$

$(a,\hat{a},c,d)$

m

Adv

b'

(m_0, m_1)

$(a^*, \hat{a}^*, c^*, d^*)$

Encryption Oracle

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

$E1:\ u \leftarrow_{\cup} \mathbf{Z}_q$

$E2:\ a^* \leftarrow g^{u}$

$E3':\ \hat{a}^* \leftarrow \hat{g}^{\hat{u}},\ \hat{u} \leftarrow_{\cup} \mathbf{Z}_q \setminus \{u\}$

$E4':\ c^* \leftarrow a^{z_1}\hat{a}^{z_2} \cdot m_b$

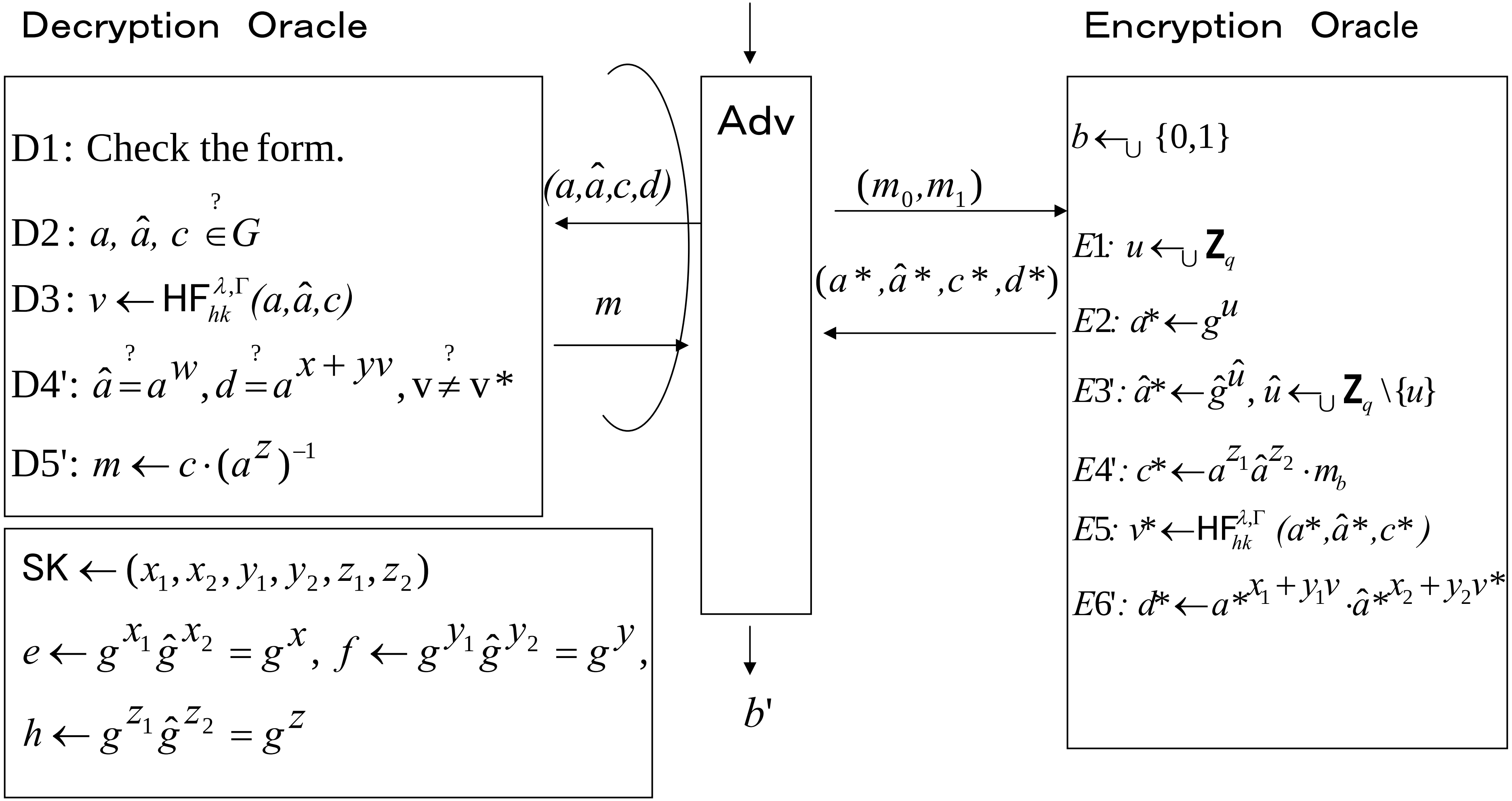
$E5:\ v^* \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}\ (a^*, \hat{a}^*, c^*)$

$E6':\ d^* \leftarrow a^{*x_1 + y_1 v} \cdot \hat{a}^{*x_2 + y_2 v^*}$

$$\text{Advantage} = |\Pr[b = b'] - \frac{1}{2}|$$

Game3

$$(\Gamma[G, \hat{G}, g, q], hk, \hat{g}, e, f, h)$$



Decryption Oracle

D1: Check the form.

D2: $a, \hat{a}, c \overset{?}{\in} G$

D3: $v \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}(a,\hat{a},c)$

D4': $\hat{a} \overset{?}{=} a^{w}, d \overset{?}{=} a^{x+yv}, \mathrm{v} \overset{?}{\neq} \mathrm{v}*$

D5': $m \leftarrow c \cdot (a^{z})^{-1}$

$\mathsf{SK} \leftarrow (x_1, x_2, y_1, y_2, z_1, z_2)$

$e \leftarrow g^{x_1}\hat{g}^{x_2} = g^{x},\ f \leftarrow g^{y_1}\hat{g}^{y_2} = g^{y},$

$h \leftarrow g^{z_1}\hat{g}^{z_2} = g^{z}$

$(a,\hat{a},c,d)$

m

Adv

b'

(m_0, m_1)

$(a^*, \hat{a}^*, c^*, d^*)$

Encryption Oracle

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

$E1$: $u \leftarrow_{\cup} \mathbf{Z}_q$

$E2$: $a^* \leftarrow g^{u}$

$E3'$: $\hat{a}^* \leftarrow \hat{g}^{\hat{u}}, \hat{u} \leftarrow_{\cup} \mathbf{Z}_q \setminus \{u\}$

$E4'$: $c^* \leftarrow a^{z_1}\hat{a}^{z_2} \cdot m_b$

$E5$: $v^* \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}(a^*, \hat{a}^*, c^*)$

$E6'$: $d^* \leftarrow a^{*x_1+y_1 v} \cdot \hat{a}^{*x_2+y_2 v^*}$

$$\text{Advantage} = |\Pr[b = b'] - \frac{1}{2}|$$

Game4

$$(\Gamma\,[G\,,\hat{G}\,,\ g\,,\ q\,],\ hk,\ \ \hat{g}\,,e,\ \ f,\ h\,)$$

$\downarrow$

Decryption Oracle

D1: Check the form.

D2: $a,\ \hat{a},\ c \overset{?}{\in} G$

D3: $v \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}(a,\hat{a},c)$

D4': $\hat{a} \overset{?}{=} a^{w}, d \overset{?}{=} a^{x+yv}, \mathrm{v} \overset{?}{\neq} \mathrm{v}^*$

D5': $m \leftarrow c\cdot(a^{z})^{-1}$

$(a,\hat{a},c,d)$

m

Adv

$\downarrow$

b'

(m_0,m_1)

$(a^*,\hat{a}^*,c^*,d^*)$

Encryption Oracle

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

$E1{:}\ u \leftarrow_{\cup} \mathbf{Z}_q$

$E2{:}\ a^* \leftarrow g^{u}$

$E3'{:}\ \hat{a}^* \leftarrow \hat{g}^{\hat{u}},\ \hat{u} \leftarrow_{\cup} \mathbf{Z}_q \setminus \{u\}$

$E4'{:}\ c^* \leftarrow g^{r}, r \leftarrow_{\cup} \mathbf{Z}_q$

$E5{:}\ v^* \leftarrow \mathsf{HF}_{hk}^{\lambda,\Gamma}\ (a^*,\hat{a}^*,c^*)$

$E6'{:}\ d^* \leftarrow a^{*x_1+y_1v} \cdot \hat{a}^{*x_2+y_2v^*}$

$\mathsf{SK} \leftarrow (x_1,x_2,y_1,y_2,z_1,z_2)$

$e \leftarrow g^{x_1}\hat{g}^{x_2} = g^{x},\ f \leftarrow g^{y_1}\hat{g}^{y_2} = g^{y},$

$h \leftarrow g^{z_1}\hat{g}^{z_2} = g^{z}$

$$\text{Advantage} \;=\; |\,\Pr[\,b = b'\,] - \frac{1}{2}\,| \;=\; 0$$

Universal Composability (UC) : An Example of Simulation-Based Formulation

Universal Composability (UC)

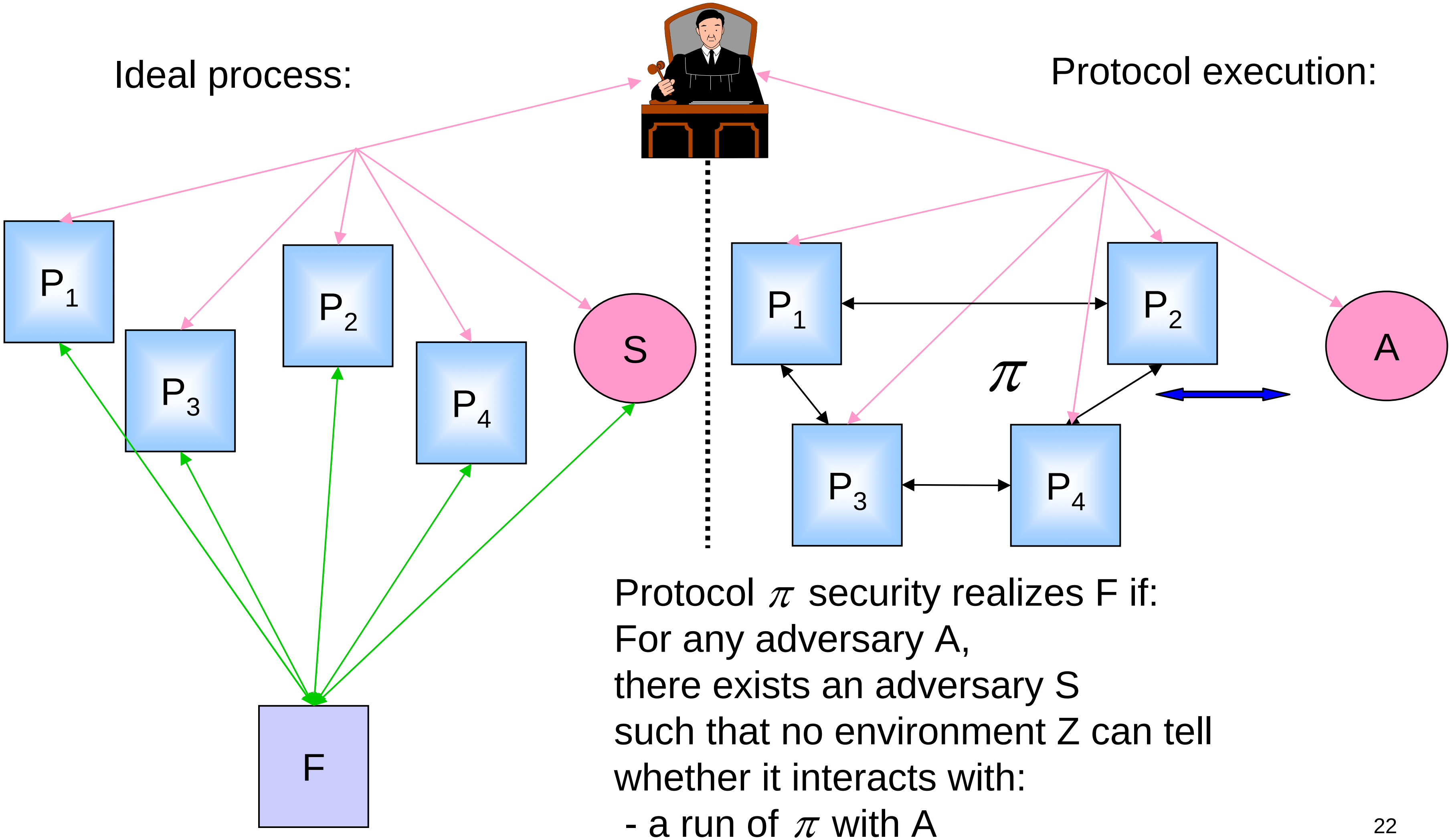
- In 2001, Ran Canetti proposed the concept of UC, which has been extensively advanced by many researchers and is being continued.
- UC guarantees the strongest security; i.e., the security is preserved under any composition and environment.
- The (UC) security of any cryptographic functionalities can be formalized in a unified manner, for primitives and protocols.

How Security Is Defined in UC (1):

1. Write an "ideal functionality" F that captures the requirements of the task at hand.

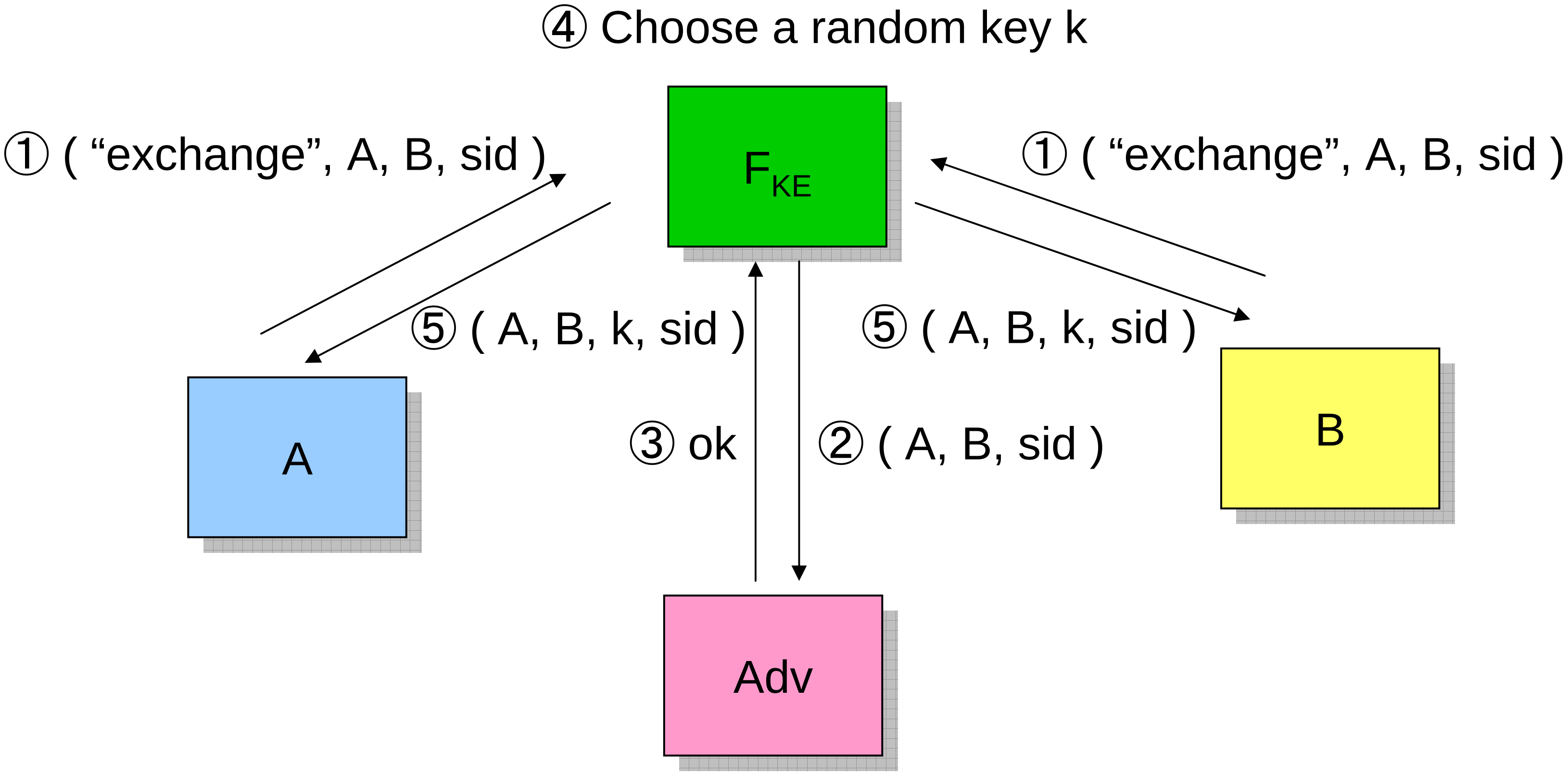
F is a "code for an ideal trusted service on the net". (F captures both correctness and secrecy requirements.)

How Security Is Defined in UC (2):

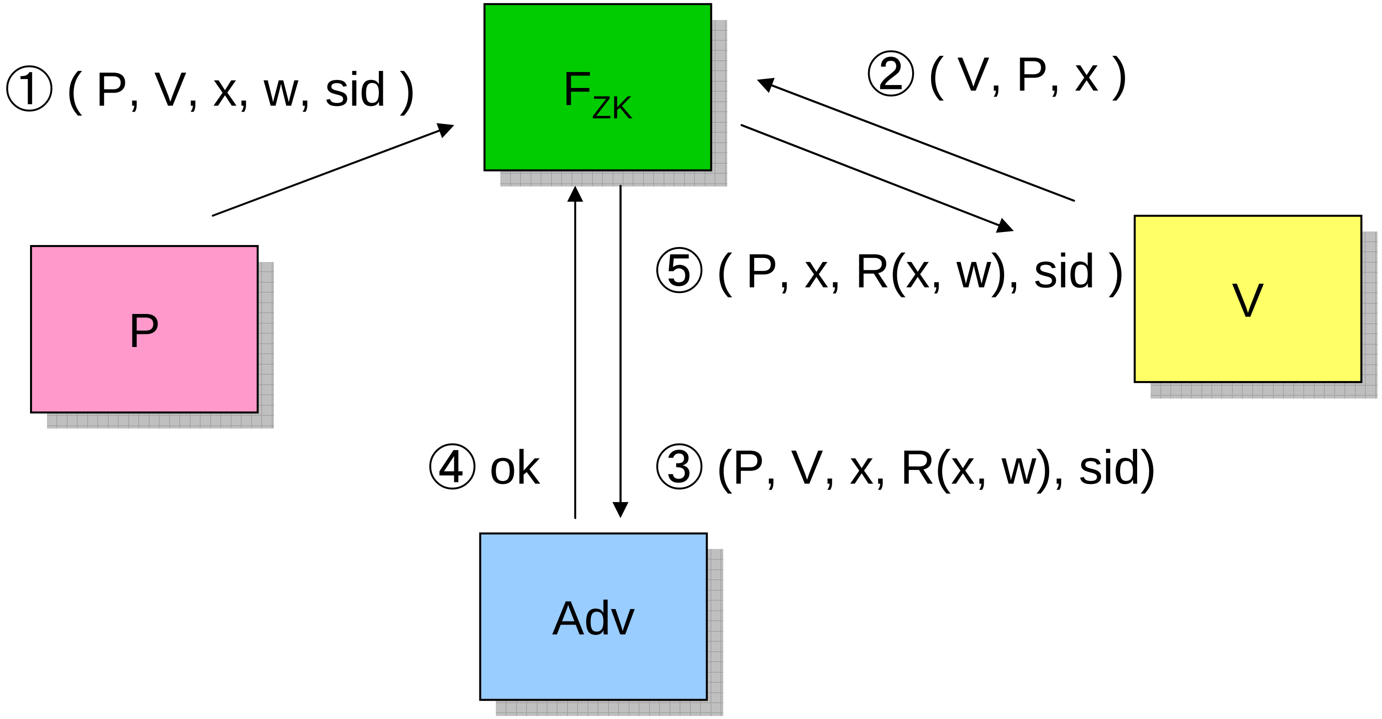


Protocol π security realizes F if:
For any adversary A,
there exists an adversary S
such that no environment Z can tell
whether it interacts with:
- a run of π with A
- an ideal run with F and S

Example: The Key-Exchange Functionality F_{KE}



Example: The ZK Functionality (for Relation R)



① (P, V, x, w, sid)

F_{ZK}

② (V, P, x)

P

⑤ (P, x, R(x, w), sid)

V

④ ok

③ (P, V, x, R(x, w), sid)

Adv

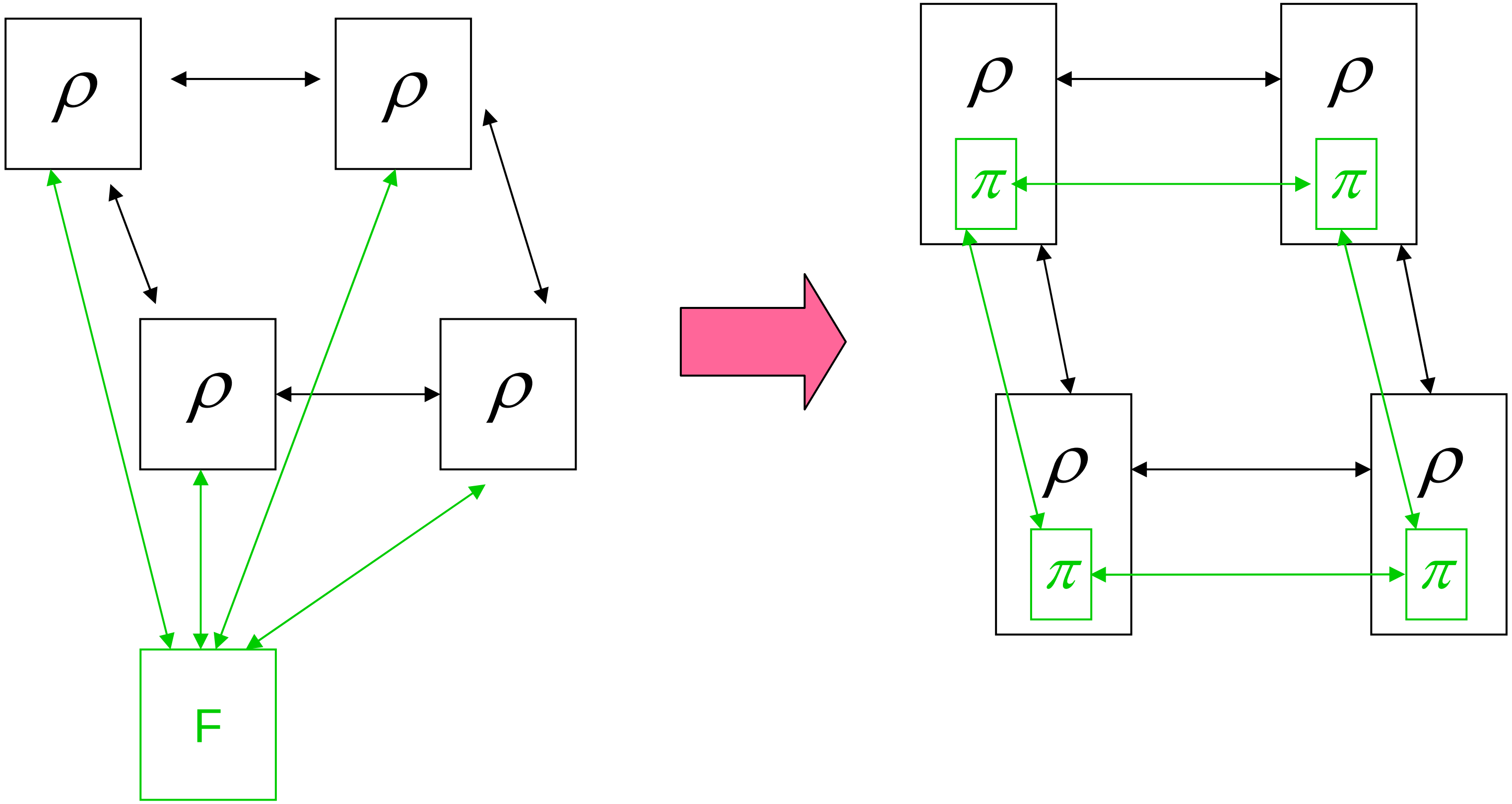
Note:
V is assumed that it accepts only if R(x, w) =1 (soudness)
P is assumed that V learns nothing but R(x,w) (Zero-Knowledge)

The Composition Operation
(originates with [Micali-Rogaway91])

- Start with:
  - Protocol ρ^F that uses ideal calls to F
  - Protocol π that securely realizes F construct the composed protocol ρ^π.
  - Each call to F is replaced with an invocation of π.
  - Each value returned from π is treated as coming from F.

Note: In ρ^F parties call many copies of F.
In ρ^π many copies of π run concurrently.

The Composition Operation



The Universal Composition Theorem:

[Canetti 01]

- Protocol ρ^{π} "emulates" protocol ρ^{F}.
  - That is , for any adversary A there exists a simulator S such that no Z can tell whether it is interacting with (ρ^{π}, A) or with (ρ^{F}, S).
- Corollary: If ρ^{F} security realizes function G then so does ρ^{π}.

(weaker composition theorem were proven in e.g. [Micali-Rogaway91, Canetti00, Dodis-Micali00, Pfitzmann-Schunter-Waidner00].)

Implications of the UC Theorem

1. Can design and analyze protocols in a modular way:
   - Partition a given task T to simpler sub-tasks $T_1 \ldots T_k$
   - Construct protocols for realizing $T_1 \ldots T_k$.
   - Construct a protocol for T assuming ideal access to $T_1 \ldots T_k$.
   - Use the composition theorem to obtain a protocol for T from scratch.

(Analogues to subroutine composition for correctness of programs, but with an added security guarantee.)

Implications of the UC Theorem

2. Assume protocol π security realizes ideal functionality F. Can deduce security of π in any multi-execution environment:

As far as the environment is concerned, interacting with (multiple copies of) π is equivalent to interacting with (multiple copies of) F.

(For instance, a protocol that realizes the ZK functionality is guaranteed to withstand all the attacks we discussed, and more.)

An Example of Formal Method Approach:
Dolev-Yao Model

Dolev-Yao Model

- Expressions:

K	key (symbol sequence）
i	bit（bit sequence）
(M,N)	pair (M, N: expressions)
$\{K\}_K$	encryption (K: key)

- $M \vdash N$ implies ``N is calculated from M''.
- Patterns: implies `` available information''.

  p(M,T) information available from key set T and M.

  $$p(\{M\}_K,T) = \begin{cases} M & \text{if } K \in T \\ \triangle & \text{otherwise} \end{cases}$$

  (△ implies ``unavailable''）

  $pattern(M) = p(M, \{K \mid M \vdash K\})$

  $pattern((\{\{K1\}_{K2}\}_{K3},K3)) = (\{\triangle\}_{K3},K3)$

- Equivalence:

  $M \equiv N$ iff $pattern(M) = pattern(N)$

  $(\{\{K1\}_{K2}\}_{K3},K3) \equiv (\{\{0\}_{K2}\}_{K3},K3)$

A New Trend:
Merging Computational Approach and Formal Method Approach

How to Apply the Formal Methods to the Computational Approach

- Aim： Simplify or (partially) automate the computational approach security proof
- Necessary Condition： Soundness of the formal method proof to the computational approach security
- Possible Areas to Apply：

(Area 1) Functionality description and security of hybrid model (e.g., the security of a protocol composed of ideal functionalities)

(Area 2) The UC security of a primitive scheme/protocol realizing an ideal functionality

(Area 3) Formalize and automate the game hopping technique

Area1: Apply the Formal Methods to Proving the Security of Hybrid Models (1)

1.Abadi-Rogaway 2000

- Shows that the security proof on a formal method (like the Dolev-Yao model) guarantees the security of symmetric-encryption-based protocols on the computational approach (i.e., shows the soundness of the formal method to the computational approach）.

2.Canetti-Herzog 2006

- Shows that the security proof on a formal method (like the Dolev-Yao model) guarantees the UC security of a hybrid model. Also shows the UC security of a key exchange protocol using a theorem-proving tool.

Area1: Apply the Formal Methods to Proving the Security of Hybrid Models (2)

1.Abadi-Rogaway 2000

- Introduces a special symbol, $\{M\}_{k,}$ that means (ideal) encryption functionality in the Dolev-Yao like formal method.

  ⇒ If this encryption functionality is considered to be a UC's ideal functionality, this result is very akin to that by Canetti-Herzog。

2.Canetti-Herzog 2006

- Introduces a special symbol, $\{M\}_{PK}$, that means (ideal) public-key encryption functionality in the Dolev-Yao like formal method.
- Introduces ideal functionality F_{CPKE} of public-key encryption in the computational model or UC

  ⇒ An ideal functionality in UC can be corresponded to a special symbol in the Dolev-Yao like formal method.

(Area 2) Apply the Formal Methods to Proving the UC Security of a Primitive Scheme/Protocol

Canetti-Cheung-Kaynar-Liskov-Lynch-Pereira-Segala 2006

- Uses a probabilistic I/O automaton (PIOA) in place of symbols/reasoning-rules with Dolev-Yao, and proves the UC security of a primitive (OT) by the formal method approach based on PIOA.

⇒A step to (partially) automating the UC security of a primitive.

(Area 3) Formalize and Automate the Game Hopping Technique

Blanchet-Pointcheval 2006

- This result applies a formal method to the computational security without ideal functionalities/symbols (instead, assuming a computational assumption), while the other results use ideal functionalities/symbols.
- Applies a formal method to the game-hopping technique in the computational approach. Extracts rules to make a sequence of the games, and generates an automated proof on a process calculus.

⇒ They used a process calculus tool and automated the security proof of some concrete digital signature schemes like FDH-RSA.

Summary

- There are two approaches to prove the security of cryptographic schemes/protocols. One is computational approach, which is widely accepted in the cryptology community as standard security definitions. The other is the formal method approach, which has been studied in the formal method community,
- Recent advances of the computational approach are clarifying the relationship between the two approaches and promoting to merge them. One is the UC framework and the other is the game-hopping technique.
- By merging the two approaches, it is expected to (partially) automate the security proof in the computational approach (or widely accepted in the cryptology community as standard security definitions).