CRYE 6138 Security Models

Jorge L. Villar

UBa Cyber Crypto Center, Fall 2025

Some Proving Techniques

Outline

Game Hopping

Security notion elements:

- **Game**: A random experiment run by a Challenger and an Adversary. A sequence of definitions of random variables.
- **Win**: A distinguished event, that defines the success of the adversary. Then, the output (0 or 1) of Game.
- The goal is finding an upper bound of $\mathrm{Pr}_{\mathsf{Game}}(\mathsf{Win})$.
- In some cases there would be other interesting events.

The analysis of Game is done by introducing a **sequence of slight modifications**. The variables are mostly the same but their probability distributions are changed.

Example: A Hybrid Argument (I)

Encryption of vectors:

$\mathsf{Enc}^k(pk, (m_1, \ldots, m_k)) = (\mathsf{Enc}(pk, m_1), \ldots, \mathsf{Enc}(pk, m_k))$

IND-CPA security of Enc^k: For any two vectors $\mathbf{m}_0 = (m_{0,1}, \ldots, m_{0,k})$ and $\mathbf{m}_1 = (m_{1,1}, \ldots, m_{1,k})$, their encryptions are unindistinguishable.

Define games $\mathsf{G}_0, \ldots, \mathsf{G}_k$ as "hybrids" of IND-CPA(Enc^k):

- In IND-CPA(Enc^k), $\mathbf{c}_* = \mathsf{Enc}^k(pk, (m_{b,1}, \ldots, m_{b,k}))$
- In G_i, $\mathbf{c}_* = \mathsf{Enc}^k(pk, (\underbrace{m_{1,1}, \ldots, m_{1,i}}_{\text{take } i \text{ from } \mathbf{m}_1}, \underbrace{m_{0,i+1}, \ldots, m_{0,k}}_{\text{take } k-i \text{ from } \mathbf{m}_0}))$.

In all games define the event Out1 that $b' = 1$.

Example: A Hybrid Argument (II)

$G_0 \equiv \text{IND-CPA}(\text{Enc}^k) \mid b = 0$ and $G_k \equiv \text{IND-CPA}(\text{Enc}^k) \mid b = 1$

Then, the advantage in the IND-CPA game is

$$\text{Adv}_{\text{IND-CPA}(\text{Enc}^k)} = \left|\Pr_{\text{IND-CPA}(\text{Enc}^k)}(b' = b) - \tfrac{1}{2}\right| =$$
$$= \left|\tfrac{1}{2}\Pr_{\text{IND-CPA}(\text{Enc}^k)}(b' = 1 \mid b = 1) + \tfrac{1}{2}\Pr_{\text{IND-CPA}(\text{Enc}^k)}(b' = 0 \mid b = 0) - \tfrac{1}{2}\right| =$$
$$\tfrac{1}{2}\left|\Pr_{\text{IND-CPA}(\text{Enc}^k)}(b' = 1 \mid b = 1) - \Pr_{\text{IND-CPA}(\text{Enc}^k)}(b' = 1 \mid b = 0)\right| =$$
$$\tfrac{1}{2}\left|\Pr_{G_k}(\text{Out1}) - \Pr_{G_0}(\text{Out1})\right| \leq \tfrac{1}{2}\sum_{i=1}^{k}\left|\Pr_{G_i}(\text{Out1}) - \Pr_{G_{i-1}}(\text{Out1})\right|$$

For any i, any non-negligible difference $\left|\Pr_{G_i}(\text{Out1}) - \Pr_{G_{i-1}}(\text{Out1})\right|$ can break IND-CPA(Enc).

Lemma

Any attack against IND-CPA(Enc^k) with advantage ε implies another attack against IND-CPA(Enc) with advantage at least ε/k.

Example: A Hybrid Argument (III)

The only difference between G_{i-1} and G_i is in the i-th coordinate of the encrypted message:

- In G_{i-1}, $(\underbrace{m_{1,1}, \ldots, m_{1,i-1}}_{\text{take } i-1 \text{ from } \mathbf{m}_1}, \underbrace{m_{0,i}, m_{0,i+1}, \ldots, m_{0,k}}_{\text{take } k-i+1 \text{ from } \mathbf{m}_0})$.
- In G_i, $(\underbrace{m_{1,1}, \ldots, m_{1,i-1}, m_{1,i}}_{\text{take } i \text{ from } \mathbf{m}_1}, \underbrace{m_{0,i+1}, \ldots, m_{0,k}}_{\text{take } k-i \text{ from } \mathbf{m}_0})$.

A reduction can define $\mathbf{c}_*$ as
$(\text{Enc}(pk, m_{1,1}), \ldots, \text{Enc}(pk, m_{1,i-1}), c_*, \text{Enc}(pk, m_{0,i+1}), \ldots, \text{Enc}(pk, m_{0,k}))$
where $c_* = \text{Enc}(pk, m_{i,b})$ is the target ciphertext in the IND-CPA(Enc) game for adversarial message pair $(m_{0,i}, m_{1,i})$.

Then, $\text{Adv}_{\text{IND-CPA}(\text{Enc})} \geq \frac{1}{2}\left|\Pr_{G_i}(\text{Out1}) - \Pr_{G_{i-1}}(\text{Out1})\right|$

Example: A Hybrid Argument (IV)

$\mathcal{C}_{\text{IND-CPA}}$:
$(pk, sk) \leftarrow \text{KeyGen}$;
$b \leftarrow \{0,1\}$;
$(m_0, m_1, st) \leftarrow \mathcal{A}_1(pk)$;

$c_* = \text{Enc}(pk, m_b)$;
$b' \leftarrow \mathcal{A}_2(st, c_*)$;

if $b' = b$ **output** 1;
else output 0;

$\mathcal{R}_1[\mathcal{A}^k](pk)$:
$(\mathbf{m}_0, \mathbf{m}_1, st^k) \leftarrow \mathcal{A}_1^k(pk)$;
$st \leftarrow (pk, \mathbf{m}_0, \mathbf{m}_1, st^k)$;
output $(m_{0,i}, m_{1,i}, st)$;

$\mathcal{R}_2[\mathcal{A}^k](st, c_*)$:
$(pk, \mathbf{m}_0, \mathbf{m}_1, st^k) \leftarrow st$;
for $1 \leq j < i$; **do**
  $c_j \leftarrow \text{Enc}(pk, m_{1,j})$;
for $i < j \leq k$; **do**
  $c_j \leftarrow \text{Enc}(pk, m_{0,j})$;
$\mathbf{c}_* \leftarrow (c_{1\ldots i-1}, c_*, c_{i+1\ldots k})$;
$b' \leftarrow \mathcal{A}_2^k(st^k, \mathbf{c}_*)$;
output b';

$\mathcal{C}_{G_{i-1}/G_i}$:
$(pk, sk) \leftarrow \text{KeyGen}$;
$b \leftarrow \{0,1\}$;

$(\mathbf{m}_0, \mathbf{m}_1, st) \leftarrow \mathcal{A}_1^k(pk)$;

$\mathbf{m}_* \leftarrow (m_{1,1\ldots i-1}, m_{b,i}, m_{0,i+1\ldots k})$;
$\mathbf{c}_* \leftarrow \text{Enc}^k(pk, \mathbf{m}_*)$;
$b' \leftarrow \mathcal{A}_2^k(st, \mathbf{c}_*)$;

if $b' = b$ **output** 1;
else output 0;

The Game Hopping Technique (I)

Sequence of games that bridges between two given games

- From the two "partial" games in an IND-CPA like definition, fixing $b = 0$ or $b = 1$.
- From the game that defines the security notion to a trivial game (with a negligible advantage/success probability)
- Between two consecutive games in other game sequence (layered game hopping)

There are different types of hop between consecutive games:

- **Game rewriting:** The games behaves identically, the computations are performed in a different equivalent way.
- **Conditional change:** There is a well-defined event F such that the games behave identically unless F occurs.
- **Computational hop:** The difference between the games would imply solving a computational problem by means of a reduction (e.g., previous example).

The Game Hopping Technique (II)

Each type of hop uses different methods:

- **Game rewriting:** All the probabilities are exactly the same in both games. It is typically used as a preparation for subsequent hops.
- **Conditional change:** The difference in the probability of an event defined in both games is upper-bounded by the probability of $\Pr(\mathrm{F})$. Typically, $\Pr(\mathrm{F})$ can be shown to be negligible, or a reduction can use a non-negligible $\Pr(\mathrm{F})$ to solve some computational problem.
- **Computational hop:** The games behave differently, but a reduction uses any noticeable difference to solve some computational problem.

After all hops are analyzed, the triangle inequality bounds the differences between the sequence endpoint games.

Vector Encryption Revisited (I)

The previous hybrid argument shows a **generic** result, for all IND-CPA encryption schemes. It is an **almost tight** reduction, because the factor $1/k$ in the advantage.

For ElGamal encryption scheme, a specific **tight** reduction using only one game hop replaces the factor $1/k$ by $1/2$, based on the random self-reducibility of DDH.

Given a DDH tuple (R, Y, Z), Y is used as the public key and the vector ciphertext uses a diferent randomized version of (R, Z) for every vector component.

$\mathbf{c}_* = ((g^{a_1} R^{b_1}, Y^{a_1} Z^{b_1} m_{b,1}), \ldots, (g^{a_k} R^{b_k}, Y^{a_k} Z^{b_k} m_{b,k}))$ for random $a_1, \ldots, a_k \in \mathbb{Z}_q$ and $b_1, \ldots, b_k \in \mathbb{Z}_q^\times$.

Vector Encryption Revisited(II)

For a DDH 0-tuple $(R = g^r, Y = g^x, Z = g^{rx})$ the ciphertext is computed **exactly as in the IND-CPA(ElGamal^k) game**.

$\mathbf{c}_* = ((g^{a_1+rb_1}, g^{x(a_1+rb_1)} m_{b,1}), \ldots, (g^{a_k+rb_k}, g^{x(a_k+rb_k)} m_{b,k}))$

For a DDH 1-tuple $(R = g^r, Y = g^x, Z = g^z)$ the messages $m_{b,1}, \ldots, m_{b,k}$ are perfectly hidden by the randomness in z and in $b_1, \ldots, b_k$, and **the advantage of the adversary is 0**.

$\mathbf{c}_* = ((g^{a_1+rb_1}, g^{x(a_1+rb_1)} \widetilde{m}_1), \ldots, (g^{a_k+rb_k}, g^{x(a_k+rb_k)} \widetilde{m}_k))$
where $\widetilde{m}_i = g^{(z-xr)b_i} m_{b,i}$

This implies that (half of) the advantage of any adversary against IND-CPA(ElGamal^k) is transferred to the adversary against the DDH assumption by the described reduction.

Vector Encryption Revisited (III)

$\mathcal{C}_{\mathrm{DDH}}$:
$r, x, z \leftarrow \mathbb{Z}_q$;
$\beta \leftarrow \{0,1\}$;
$R \leftarrow g^r$; $Y \leftarrow g^x$;
if $\beta = 0$ **then** $Z = g^{rx}$;
else $Z = g^z$;
$\beta' \leftarrow \mathcal{A}_{\mathrm{DDH}}(R, Y, Z)$;
if $\beta' = \beta$ **output** 1;
else output 0;

$\mathcal{R}[\mathcal{A}](R, Y, Z)$:
$b \leftarrow \{0,1\}$;
$(\mathbf{m}_0, \mathbf{m}_1, st) \leftarrow \mathcal{A}_1(Y)$;
for $1 \le i \le k$ **do**
  $a_i \leftarrow \mathbb{Z}_q$; $b_i \leftarrow \mathbb{Z}_q^\times$;
  $c_i \leftarrow (g^{a_i} R^{b_i}, Y^{a_i} Z^{b_i} m_{b,i})$;
$\mathbf{c}_* \leftarrow (c_1, \ldots, c_k)$;
$b' \leftarrow \mathcal{A}_2(st, \mathbf{c}_*)$;
if $b' = b$ **output** 0;
else output 1;

$\mathcal{C}_{\mathrm{IND\text{-}CPA(ElGamal}^k)}$:
$x \leftarrow \mathbb{Z}_q$;
$Y \leftarrow g^x$;
$b \leftarrow \{0,1\}$;
$(\mathbf{m}_0, \mathbf{m}_1, st) \leftarrow \mathcal{A}_1(Y)$;
$\mathbf{c}_* \leftarrow \mathrm{Enc}^k(Y, \mathbf{m}_b)$;
$b' \leftarrow \mathcal{A}_2(st, \mathbf{c}_*)$;
if $b' = b$ **output** 1;
else output 0;

$\beta = 0$: perfect simulation, $\beta = 1$: zero advantage

Outline

Random Oracle Model

Random Oracle Model (ROM): In the security analysis a real hash function is replaced by **a truly random function available to all parties**.

In the real world no truly random function available to all parties exists, so **ROM is an ideal model of computation**.

The random function is simulated with a global dynamically generated random table $\{x_i, y_i = H(x_i)\}$.

The security proofs in the ROM are just heuristic:

There exist cryptosystems secure in the ROM, but insecure for any possible instantiation with real hash function.

Random Oracle Simulation

In a reduction in the ROM, a random oracle $H : \{0,1\} \to \mathcal{Y}$ is simulated by a table of random values:

- Any "new" call $H(x)$ is answered with a random value $y \leftarrow \mathcal{Y}$, which is stored in the table as (x, y).
- Any "repeated" call $H(x)$ is answered with the value of y stored in the table.
- Although a truly random function requires exponential space, the table will only have polynomial size.

Random oracles provide **automatic privacy amplification**: If an adversary learns something about $H(x)$ is because it knows x.

Reductions With Random Oracles

A typical proof of security of a cryptosystem Π in the ROM means proving the reduction $P \overset{\mathcal{R}}{\Rightarrow} \mathrm{SEC}(\Pi^{H(\cdot)})$ for some problem family P, where $\Pi^{H(\cdot)}$ uses some hash function H modelled as a random oracle.

The reduction $\mathcal{R}$ receives an instance of P and tries to solve it using an attacker against $\mathrm{SEC}(\Pi^{H(\cdot)})$. Thus, $\mathcal{R}$ has to simulate the random oracle for the attacker.

In an implication between security notions in the ROM, $\mathrm{SEC1}(\Pi) \overset{\mathcal{R}}{\Rightarrow} \mathrm{SEC2}(\Pi)$, the reduction simulates the oracles specified in SEC2 but it can use the oracles provided by SEC1.

Reduction Example: Basic IND-CPA Construction (I)

A basic IND-CPA secure encryption in the ROM from any injective trapdoor one-way function family $f_k : \mathcal{X}_k \to \mathcal{Y}_k$ and a hash function H:

KeyGen(ℓ) :
$(k, t) \leftarrow$ Sample(ℓ);
output (k, t);

Enc(k, m) :
$x \leftarrow \mathcal{X}_k$;
output (Eval(k, x), $H(x) \oplus m$); // $c = (f_k(x), H(x) \oplus m)$

Dec(t, c) :
$(c_1, c_2) = c$;
output $H(\text{Inv}(t, c_1)) \oplus c_2$; // $m = H(f_k^{-1}(c_1)) \oplus c_2$

In the ROM, getting any partial information about m implies recovering the whole preimage x.

Reduction Example: Basic IND-CPA Construction (II)

Proof strategy in the ROM:

The adversary $\mathcal{A}$ against IND-CPA cannot directly evaluate $H(\cdot)$, but it must call the (simulated) random oracle.

There are two cases:

- $\mathcal{A}$ at some point queries $H(x_*)$ to the oracle.
- $\mathcal{A}$ does not query $H(x_*)$ to the oracle.

In the first case, $\mathcal{A}$ is able to invert the one-way function on $c_{*,1}$.

In the second case, $\mathcal{A}$ cannot notice the difference between computing $c_* = (f_k(x_*), H(x_*) \oplus m_b)$ or $c_* = (f_k(x_*), r \oplus m_b)$ for a random r. Then, b is perfectly hidden from $\mathcal{A}$.

The whole advantage of $\mathcal{A}$ must come from the first case.

Reduction Example: Basic IND-CPA Construction (III)

Sequence of games:

G_0: The actual IND-CPA game, but the challenger also simulates the random oracle H. The target ciphertext is computed as $c_* = (f_k(x_*), H(x_*) \oplus m_b)$ for $x_* \leftarrow \mathcal{X}_k$.

$$\Pr_{G_0}(b' = b) = \tfrac{1}{2} + \text{Adv}_{\text{IND-CPA}}(\mathcal{A})$$

G_1: Same game but now $c_* = (f_k(x_*), r \oplus m_b)$ for a random r.

$$\Pr_{G_1}(b' = b) = \tfrac{1}{2}$$

Let F be the event that at some point $\mathcal{A}$ queries $H(x_*)$.

Both games are identical unless F occurs. Then,

$$\text{Adv}_{\text{IND-CPA}}(\mathcal{A}) = \left|\Pr_{G_1}(b' = b) - \Pr_{G_0}(b' = b)\right| \leq \Pr \mathsf{F}$$

Reduction Example: Basic IND-CPA Construction (IV)

F can be used in a reduction to break the one-wayness of f_k.

$\mathcal{C}_{\text{TOW}(f)}$:
$(k, t) \leftarrow$ Sample;
$x_* \leftarrow \mathcal{X}_k$; $y_* \leftarrow f_k(x_*)$;
$x' \leftarrow \mathcal{A}_{\text{TOW}}(k, y_*)$;

if $x' = x_*$ **output** 1;
else output 0;

$\mathcal{R}[\mathcal{A}](k, y_*)$:
$b \leftarrow \{0,1\}$; $r \leftarrow \{0,1\}^\ell$;
$(m_0, m_1, st) \leftarrow \mathcal{A}_1^{H(\cdot)}(k)$;
$c_* \leftarrow (y_*, r \oplus m_b)$;
$b' \leftarrow \mathcal{A}_2^{H(\cdot)}(st, c_*)$;
$x' \leftarrow \mathcal{X}_k$; **output** x';

$H(x)$:
if $f_k(x) = y_*$ **halt & output** x;
if (x, h) in table **output** h;
else
  $h \leftarrow \{0,1\}^\ell$; store (x, h);
  output h;

$\mathcal{C}_{G_1}$:
$b \leftarrow \{0,1\}$;
$(k, t) \leftarrow$ Sample;
$(m_0, m_1, st) \leftarrow \mathcal{A}_1^{H(\cdot)}(k)$;
$r \leftarrow \{0,1\}^\ell$; $x_* \leftarrow \mathcal{X}_k$;
$c_* \leftarrow (f_k(x_*), r \oplus m_b)$;
$b' \leftarrow \mathcal{A}_2^{H(\cdot)}(st, c_*)$;
if $b' = b$ **output** 1;
else output 0;

$H(x)$:
if (x, h) in table **output** h;
else
  $h \leftarrow \{0,1\}^\ell$; store (x, h);
  output h;

$\text{Adv}_{\text{IND-CPA}}(\mathcal{A}) \leq \text{Succ}_{\text{TOW}}(\mathcal{R}[\mathcal{A}])$

The Generic Group Model

Ideal computation model for DLOG-related security analysis.

The group elements $x \in G$ are hidden with a **random encoding** $e_x = H(x)$, and the group operations are performed via **oracles**, like $e_{xy} \leftarrow \mathrm{Op}(e_x, e_y)$.

The direct manipulation of the encoding of a group element results in an error with overwhelming probability.

The group encodings give no additional information to the adversary, other than the equality of two group elements.

The simulation of the oracles is performed with a table $\{(x, e_x)\}$.

The only feasible attacks cannot use any special property of the group G, besides being a cyclic group of prime order q.

Rationale of Ideal Models

They do not provide real security proofs but only heuristic arguments.

The failure of a cryptosystem proven secure in the Random Oracle Model would not be due to a bad design but to the a weakeness of the selected hash function.

The security problem is fixed by a better choice of the hash function.

The failure of a cryptosystem proven secure in the Generic Group Model would not be due to a bad design but to the a weakeness of the selected group.

The security problem is fixed by a better choice of the group.

CRYE 6138 Security Models

Jorge L. Villar

UBa Cyber Crypto Center, Fall 2025

Some Proving Techniques

—END—