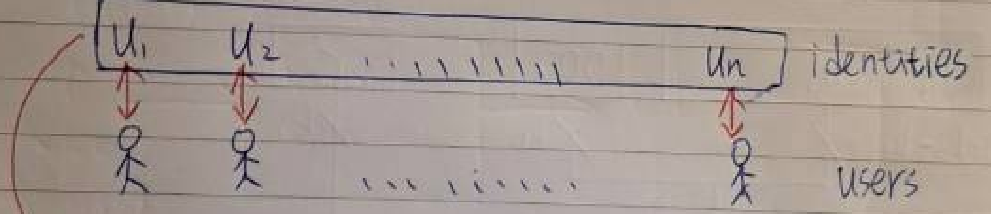
MONTH

DATE

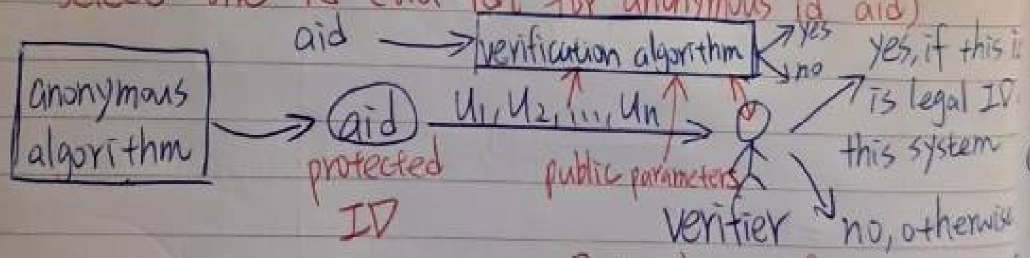
合法性

Identity Anonymity: A scheme achieves identity anonymity can prove the (legitimacy) of a protected (anonymous) identity submitted without exposing the real identity information of this protected ones, where the protected identity is produced from one of any two selected real identities and anonymity means no adversary can obtain non-negligible advantage to link the protected ones to one of the two real IDs.

The temporal identity encoded together with certain veritable information by digital signature or cryptography algorithm

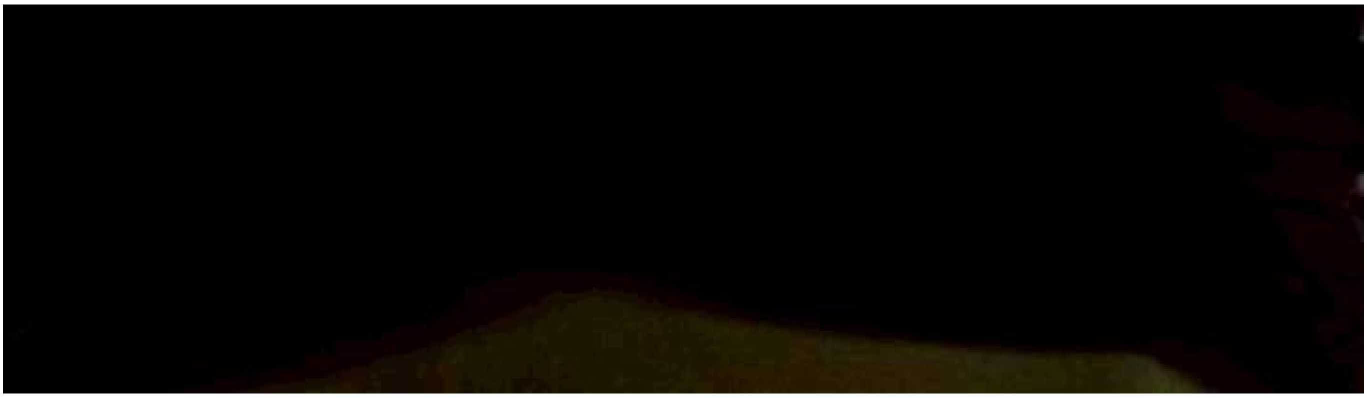


select one id (i.e. id_i for anonymous id aid)



But the verifier cannot tell you aid is linked to which U_i.

Big Question: How to estimate the security of an anonymity id system?



MONTH
DATE

How to estimate the security of an encryption (E, D)?

E: encryption algorithm
D: decryption algorithm

$m \rightarrow \boxed{E} \rightarrow C$

m: message/plaintext
C: ciphertext

$C \rightarrow$ Attacker $\rightarrow$ What is the goal of breaking (E, D)?

(recovering)
① obtaining m?
② obtaining D?
③ otherwise?

Why we have the requirement of Confidentiality?

☆ $\boxed{\text{Decision}}$

0: 50% (p)
1: 50% (1-p)

$\rightarrow \boxed{\text{Communication System}} \rightarrow$?

0 for p
1 for 1-p

Qualified

information → yes
information → no

(p) 0, (1-p) 1 $\rightarrow \boxed{E} \rightarrow$ C can encryption on either 0 or 1

(1-p)

to

Qualified

observe

Attacker → 0 for p 50%, 1 for 1-p 50%

MONTH

DATE

Select by attackers

Semantic Security Game

random bit $b \in \{0,1\}$

m_0, m_1 → E → $E(m_b) = C$ → Attacker

Attacker $\xrightarrow{\text{guess}}$ $b' \in \{0,1\}$. If $b' = b$, attacker gets success

$$Adv_A^{(E,D)} = Pr[b = b'] - \frac{1}{2}$$

E is of semantic security if $Adv_A^{(E,D)}$ is negligible. ☆

(u_0, u_1)

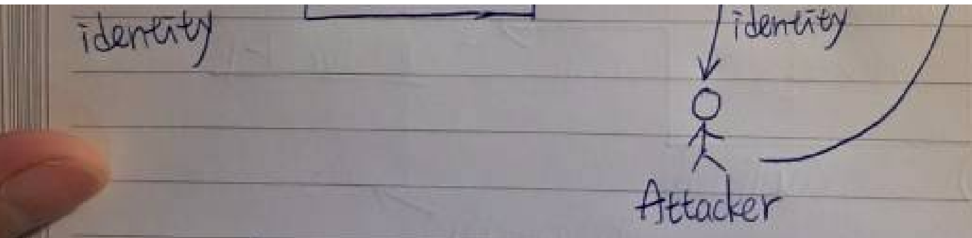
u_i exact → Anonymous System → aid

anonyous

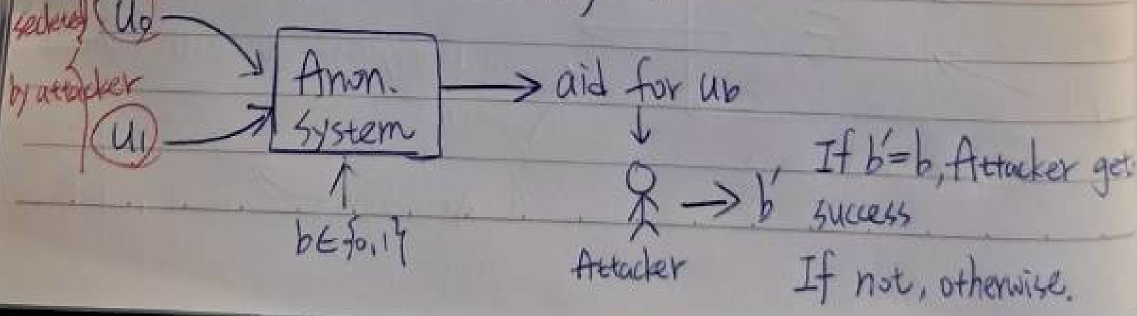
$\boxed{u_i' = u_i}$

$Adv_A^{anon.sy}$

☆ A is negligible



Anonymous Security Game



MONTH

DATE

$$Adv_A^{anon.sys} = P_r[b=b'] - \frac{1}{2}$$

☆ A is said to be anonymous secure, if $Adv_A^{anon.sys}$ is negligible.

MONTH

DATE

資訊安全理論與實務

"This is a course regarding information security."

easy to break ← Algorithm ?

msg

known

secure? Encoding ← key

↓ msg

"XYZA7E3S5G......."

encryption procedure

Base 64
用於網路間傳送

11/12, 12/03, 12/24 報告論文和實作

Diffie - Hellman 1976
(key Agreement)

RSA by 1978
(public-key encryption)

① Introduction to Cryptography
How to define the security of a function (to protect "something"?

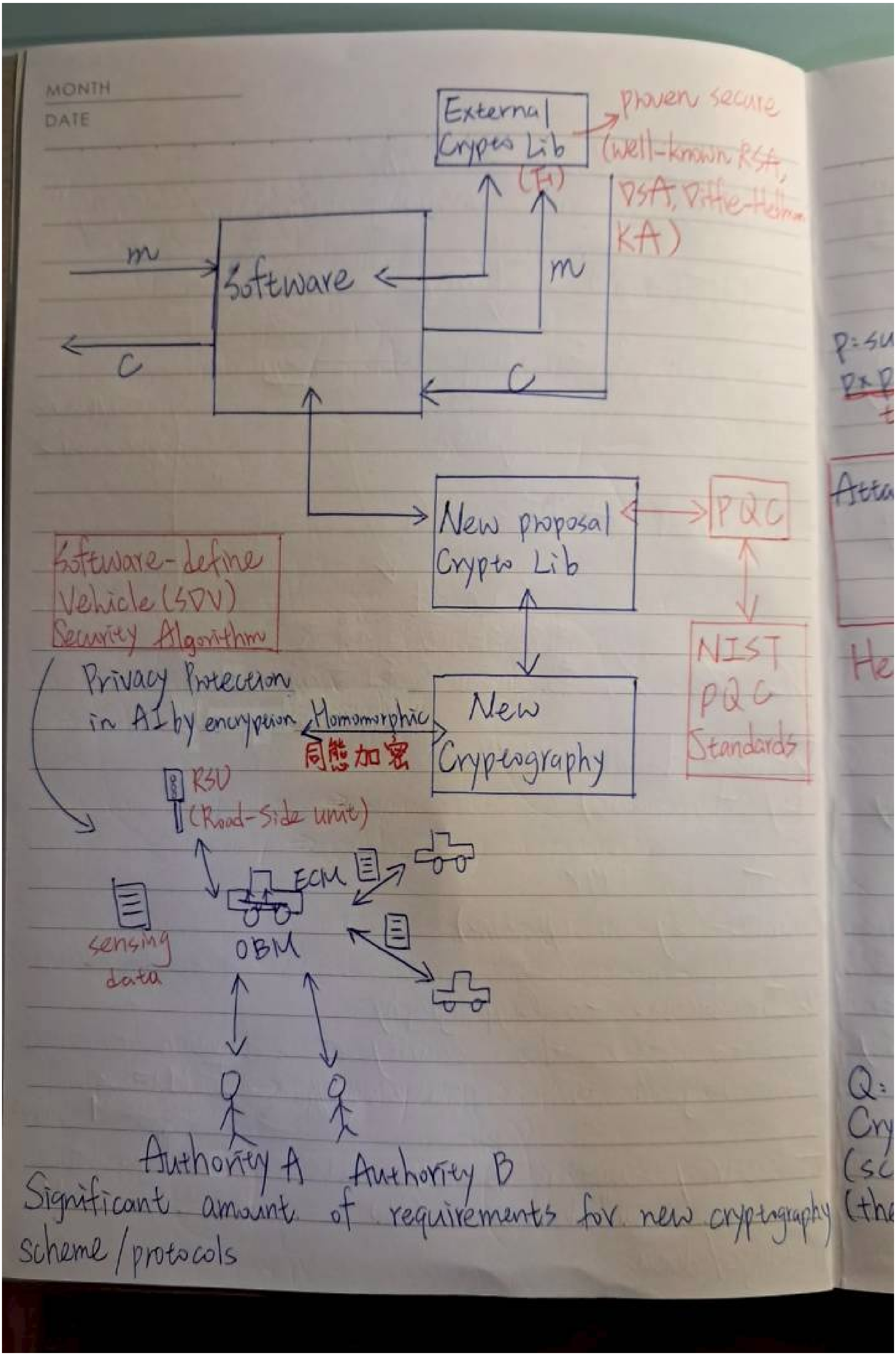
an encoding message C

a security function $E(M)=C$

a message M as input

we need to define the format of M

A security function is to protect an "information", but not saw message/data.


MONTH
DATE
External Crypto Lib
proven secure (well-known RSA, DSA, Diffie-Hellman KA)
(H)
m
Software
c
m
c
New proposal Crypto Lib
PQC
NIST PQC Standards
Software-define Vehicle (SDV) Security Algorithm
Privacy Protection in AI by encryption
Homomorphic
同態加密
New Cryptography
RSU (Road-Side unit)
ECM
sensing data
OBM
Authority A
Authority B
Significant amount of requirements for new cryptography scheme / protocols

[redacted]

MONTH
DATE

$\boxed{\text{Cryptography Algorithm}}$ → Security for Cryptography

↓

$\boxed{\text{How to prove it security ?}}$

p: success rate

$\underbrace{p \times p \times \dots \times p}_{t \text{ times}} = p^t$

X X X

Attack Algorithm 1
AA 2
⋮
AA n

Heuristic(啟發式的)

B is provable secure

provable security → If A is secure, B based on A is also secure

↓

$\boxed{\text{Not 100\% secure}}$

A is assumed to be hard / intactable 棘手的

NP hard / complete

$P \le \frac{1}{c^n}$ (ex: $\frac{1}{2^{1024}}$)

Q: What cryptography is about ?

Cryptography is the discipline 課程 that studies systems (scheme, protocol) that preserve their functionality (their goal), even under the presence of an active disrupter

Passive Attacker $\boxed{\text{Eavesdropping}}$ Man in-the-middle

MONTH

DATE

Man-in-the-middle

intercept message
modify "
delete "
} active attacker

◎Classic Problems / Goals

The principles of information security
- Confidentically
- Integrity
- Availability

- Integrity: Messages have not been altered
- Authenticity: Message comes from sender
- Secrecy: Message not know to anybody else

→ Non-repudiation 不可否認性

→ Public-key crypeography

互動地

interactively prove ⇒ symmetric key based cryptography

non-interactively ⇒ public-key cryptography

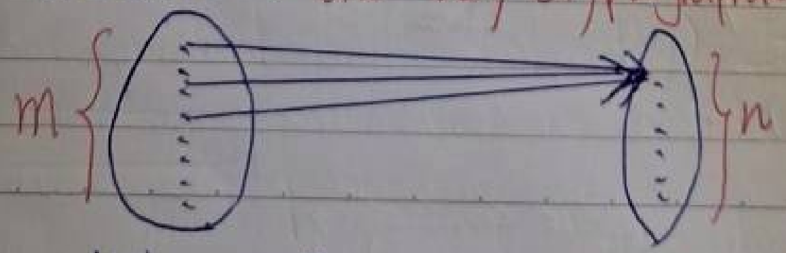
非互動地

We want to ① Store a document data-at-rest

② Send a message data-in-transit(過境)

→ security protocol to negotiate a session key

hash function → one-way cryptographic function



$|H| = m \times n$ $\quad$ $m \geq n$

MONTH

DATE

(ex. 128-bits, 1024 bits)
fixed length

Digital Signatures

M: message, pk: public key, sk: secret key

pair

$Sig(sk, M) = \sigma$ $\quad$ $Ver(pk, \sigma, M) = \{true, false\}$

sign function $\quad$ signature $\quad$ Verify function

D: data of large size, how to sign it?

Hash function $H: \{0,1\}^* \to \{0,1\}^\ell$ → fixed

$H(D) = M$, $Sig(sk, H(D)) = \sigma$, $Ver(pk, \sigma, H(D)) = \{true, false\}$

two dollars check

$H(D)$ → M

two dollars check

$H(v')$

collision

② Provable security

③ The need for provable security

Cryptanalysis driven confidentiality secrecy integrity

The Recipe security goal system model

① Define goal of scheme (or adversary) potential Attackers

② Define attack model what attackers can do

③ Give a protocol An attacker model

④ Define complexity assumptions (or assumptions on the primitive)

⑤ Provide a proof by reduction ⑥ Verify proof ⑦ Interpret proof

MONTH
DATE

Definition to "security"

◎ Generalization: One-way function

$\boxed{\text{One-way function}}$: The function $f: \underline{Dom(f)} \to \underline{Rec(f)}$ (input, output)

$x \to y = f(x)$ (easy, polynomial-time) $n^2, n^3, n^{100}, n^{1000}, \ldots, n\log n$

$y = f(x) \to x$ (input, output) (difficult for random $x \in Dom(f)$, at least super-polynomial) $2^n, 3^n, c^n, n!, \ldots$

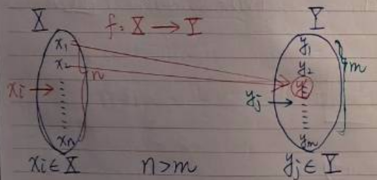
The advantage of an inverting adversary A is thus

$Adv_f^{ow}(A) = Pr[x \xleftarrow{\$} Dom(f), y = f(x) : A(y) = x]$

Resoure of A:

① Running time t (number of operations)

② Number & length of queies (if in random oracle model)



$\boxed{\begin{matrix} f(x_1) = y' \\ f(x_2) = y' \end{matrix}}$ (easy) if knowing y, can we determine it is calculate from x_1 or x_2? (impossible)

the probability of collision is negligble

Definition (Neglible Probability): If a probability $p \le \frac{1}{\text{Sup-poly}(n)}$, then p is negligble.

super polynomial function ($c^n, n!, 2^n, \ldots$)

MONTH
DATE

X Ideal Assumpeions (Perfect Security)

◎ The need of $\boxed{\text{computational assumptions}}$

Consider asymmetric cryptography, an encryption schemes $AS = (K, \mathcal{E}, \mathcal{D})$ is composed by three algorithm:

K: Key generation

$\mathcal{E}$: Encryption

$\mathcal{D}$: Decryption

$r' \to \boxed{K} \to (k_e, k_d)$

randomness; public-key, private-key; corresponding

k_e public-key; k_d private key

plaintext $m \to \boxed{\mathcal{E}} \to c$ (ciphertext) $\to \boxed{\mathcal{D}} \to m$ or $\perp$

$r \to \boxed{\mathcal{E}}$; F; F^{-1}

$F^{-1}(\underbrace{F(m)}_{c}) = m$

randomness, internal parameter

$x \to$ secret $\leftarrow y$

Alice: $X = g^x$

Bob: $Y = g^y$

$Y^x = g^{xy} = X^y$

$\boxed{\text{Diffie-Hellman Key Agreement}}$

$E(m_1) = C_1$ if $m_1 \ne m_2$, then $C_1 \ne C_2$

$E(m_2) = C_2$ but $m_1 = m_2$, then $C_1 \stackrel{?}{=} C_2$

r: $E(r, m) = c$

$E(m_1) = C_1$ $C_1 \ne C_2$, even if $m_1 = m_2$ (100% 成立)

$E(m_2) = C_2$

r': $r \ne r'$

RSA: 1024-bit, 2048bit, ..., 4096bit

k_d (private key) or $\underline{m\text{: message}}$, and r — secret key

◎ Unconditional secrecy is not possible of symmetric encryption

The ciphertext $c = \mathcal{E}_{k_e}(m; r)$ is uniquely determined by (l-bit or using)

① The public encryption key $k_e \leftarrow \{0,1\}^{l_2}$ $\to 2^{l_1} \times 2^{l_2}$

② The message m

③ The random coins $r \leftarrow \{0,1\}^{l_1}$

So, at least exhaustive search is possible!

MONTH
DATE

So, at least exhaustive search is possible!

$\Rightarrow$ unconditional secrecy is impossible

We need complexity (algorithm) assumption

◎ Integer Factoring and RSA

Multiplication vs. Factorization $O(n^2)$

$p, q \to n = p \cdot q$ is easy (quadratic) defines one-way

$n = p \cdot q \to p, q$ is hard (super-polynomial) function

RSA Function

The function $f: \mathbb{Z}_n \to \mathbb{Z}_n$ (finite field), where $n = pq$, for a fixed exponent e: x (plaintext) $\to x^e \bmod n$ (easy, cubic)

public-key $y = x^e \bmod n$ (ciphertext) (known) $\to x$ (difficult without p, q)

but easy $x = y^d \bmod n$ if trapdoor (find) $d = e^{-1} \bmod \underline{\phi(n)}$ (Euler's function) is known.

We measure the advantage of any inverting adversary A

by $\underline{Adv_{n,e}^{rsa}(A)} = Pr[\underbrace{x \xleftarrow{\$} \mathbb{Z}_n^*, y = x^e \bmod n}_{\text{condition}} : A(y) = x]$

$\hookrightarrow$ randomly select an element from $\mathbb{Z}_n^*$

Fermat's theorem

$a^{p-1} = 1 \pmod p$

$a^0 = 1 \pmod p$

$a^x \pmod p \equiv a^{x \bmod (p-1)} \bmod p$

☆ if $n = p \times q$, then $a^{\phi(n)} \equiv 1 \pmod n$

Note: Advantage $\ne$ Probability of A's goal, but here they are equal.

MONTH
DATE

◎ The Discrete Logarithm (operation of G with elements)

Let $G = (\langle g \rangle, \times)$ be any finite cyclic group. (a set of elements)

For any $y \in G$, we define $DLog_g(y) = \min\{x \ge 0 \mid y = g^x\}$ (generator) $\{g^0, g^1, g^2, \ldots, g^{n-1}\} = \{0, 1, 2, \ldots, n-1\}$

Exponenciation Function

The function $DExp_g: \mathbb{Z}_q \to G$, where $q = |G|$:

$x \to y = g^x$ (easy, $\boxed{\text{cubic}}$ $O(n^3)$) $g^0, g^1, \ldots, g^q = \{1, 2, \ldots, q\}$

$y = g^x \to x$ (difficult, super-polynomial) $g^x \bmod p$, $p = \frac{p-1}{2}$

$Adv_g^{dl}(A) = Pr[x \xleftarrow{\$} \mathbb{Z}_q, y = g^x : A(y) = x]$

Note: n is the size of input $\Rightarrow x$ is n-bit

③ Reduction (化约)

$$\begin{array}{ccc} P & \xrightarrow{\text{Transform}} & P' \\ \downarrow \times \text{difficult} & & \downarrow \text{easy} \\ Sol(P) & \xleftarrow[\text{easy}]{\text{convert}} & Sol'(P') \end{array}$$

$\Rightarrow$ reduction

◎ Algorithmic assumptions are necessary

Recall that for RSA: Reduce P to P'

$n = pq$, public modulus $\mathcal{E}_{n,e}(m) = m^e \bmod n$

e: public exponent and $\mathcal{D}_{n,d}(c) = c^d \bmod n$ — Hard problem; Proposed Security Algorithm (Cryptography)

$d = e^{-1} \bmod \phi(n)$: private exponent

Underlying hard problem:

Computing m from $c = \mathcal{E}_{n,e}(m)$ for $\boxed{m \xleftarrow{\$} \mathbb{Z}_n^*}$

finite field $\mathbb{Z}_n^*$ ($\mathbb{Z}_n \setminus \{1\} = \mathbb{Z}_n^*$)

Easy fact: If the RSA problem is easy, secrecy does not hold: anybody (not only the owner of the trapdoor) can remove m from c.

MONTH
DATE

Problem reduction

assumption ⟷ Security goal

But are algorithm assumptions sufficient?
We want the guarantee that an assumption is enough for security.
For example, in the case of encryption.

$Pr[A]$ $\leq$ $Pr[B]$

If an adversary can break the secrecy (More difficult) (P')	→	Then we can break the assumption (P)

A → B
holds then holds

(B) A — Attacker Ability

Hard problem P — Security Algorithm P'

$\underbrace{Pr[P']}_{\text{也趨近於0}} \leq \underbrace{Pr[P]}_{\text{趨近於0}}$

This is a reductionist proof!

◎ Proof by reduction
Let P be a problem.
Let A be an adversary that breaks the schemes.
Then A can be used to solve P ← We want to solve

P → the scheme P' (Homework)
sol ↓ — A can solve (power and smart guy)
sol(P) ⟵ sol(P')

MONTH
DATE

Instance I of P → [New Algorithm for P [Adversay A]] → Solution of I

Sol x for x^2 — A machine created to solve problem P
$x = 1, 2, 3, \pi, 3.5, \ldots$
↑ I

If so, we say solving P reduces to breaking the scheme.
Conclusion: If P untractable then scheme is unbreakable

◎ Provable Security?
A misleading name?
Not really proving a scheme secure but showing a reduction from security of scheme to the security of the underlying assumption (or primitive) ⇒ Reduction Security

assumption, hard problem > primitives (P)
Security Algorithm, Scheme, Cryptographic scheme } scheme to be proven (P')

B

$$\begin{array}{ccc} P & \longrightarrow & P' \\ & & \downarrow A(P') \\ sol(P) & \longleftarrow & sol(P') \end{array}$$

Provable security scheme
Before calling a scheme provably secure, we need
① To make precise the algorithmic assumptions (some given)
② To define the security notions to be guaranteed (next)
→ Security goal, Attack model
③ A reduction!

MONTH
DATE

◎ Complexity-theory vs. Exact security vs. Practical
The interpretation of the reduction matters!
(翻譯)

Given: A within time t, success probability ε ⟹ Build: Algorithm against P that runs in time $t' = T(t)$ with success probability $\varepsilon' = R(\varepsilon)$

The reduction requires showing T (for simplicity, suppose R depends only linearly in ε)
- Complexity theory: T polynomial
- Exact security: T explict
- Practical security: T small (linear)

} $\varepsilon \approx \varepsilon'$

Each gives us a way to interpret reduction results. (翻譯)

Given: A within time t and success probability ε ⟹ Build: Algorithm against P that runs in time $t' = T(t, \varepsilon)$ (non-polynomial)

- Assumption: P is hard = "no polynomial time algorithm"
- Reduction: T is polynomial in t and ε
- Security result: There is no polynomial time adversary… which really means that there is no attack if the parameters are large enough

tuning machine ⇒ probabilistic polynomial-time tuning machine (PPT)

Not always meaningful, as when analyzing block ciphers.

MONTH
DATE

◎ Measuring the Quality of the Reduction

$P \xrightarrow{\text{tight-reduction}} P'$
1024-bit — 1024-bit ×10, ×100 → inefficient

How much is lost in the reduction? How much of the "power" of adversary A breaking the scheme remains in the algorithm breaking the problem P.

Tightness: A reduction is tight if $t' \approx t$ and $\varepsilon' \approx \varepsilon$.
Otherwise, if $t' \gg t$ or $\varepsilon' \ll \varepsilon$, the reduction is not tight.
The tightness gap is $(t'\varepsilon)/(t\varepsilon') = (t'/\varepsilon')(t/\varepsilon)$
We want tight reduction, or at least reductions will small tightness gap.

◎ 補充
(a) Complexity-theorem Security: Results
General Results: Under polynomial reductions, against polynomial-time adversaries
(1) Trapdoor one-way permutations are enough for secure encryption
(2) One-way functions are enough for secure signatures
If only care about feasibility, these results close the chapter (no more problems left)… but
- the schemes for which these results were originally obtained are rather inefficient,
- looking into the complexity of the reduction may gives us some insight.

④ Security Notions

Security Notions: Example

Problem: Authentication and no-repudiation (i.e. signature) 不可否認性

How do we come up with a security notion?

$A \xrightarrow{msg} B$ — contain a knowledge that can prove msg is sent by A

$msg = \{ID, msg\_content\}$ ✗ — copy or forge; ID: identity of A

$\rightarrow msg = \{\underline{proof\_ID\_msg}, msg\_content\}$ — every time is difficult (unforgeable 不可偽造)

We need to think and define

① Security goal of the scheme (= Opposite to Adversary's goal)
→ Property that needs to be guaranteed

② Attack model → unique
→ Attack venues, what the adversary can and cannot do
→ Leaked information, what the adversary can know from honest users (often modeled by oracles)

☆ ⓪ Signature Schemes (Authentication)

Goal: Existential Forgery — condition of success by Adversary

(The adversary wins if it forges a valid message-signature pair without private key) Informal description

Adversary does a good job (or the scheme is insecure) if
- given the verification key kv → public key
- outputs a pair m', σ' of message and its signature

such that the following probability is large $Pr[Vf(kv, m', \sigma') = 1]$

⓪ Possible Attack Models

① No-Message Attack (NKA): adversary only knows the verification key

② Known-Message Attack (KMA): adversary also can access list of message / signature pairs.

③ Chosen-Message Attack (CMA): adversary can chose the messages for which he can see the message/signature pairs. (Strongest attack)

⓪ Security Notion for Signature Scheme: EUF-CMA

Given signature scheme $\Sigma = (K, Sign, Vf)$

Security game:

$(kv, ks) \xleftarrow{\$} K(\cdot)$

$\downarrow kv$ adversary; $\downarrow ks$ Signing Oracle: $\sigma \leftarrow Sign(ks, m)$

adversary $\xrightarrow{m}$ Signing Oracle (chosen by adversary)

adversary $\xleftarrow{\sigma}$ Signing Oracle (produced by Signing Oracle) CMA

$\downarrow (m', \sigma')$

$Adv_{\Sigma}^{euf\text{-}cma}(A) = Pr[Vf(k, m', \sigma') = 1, \text{ for new } m']$

→ The advantage by Adversary against Σ under euf-cma security game. (security goal: euf; condition of attacker: cma)

(Existenal unforgeability under chosen-message attacks)

Security Definition (EUF-CMA security)

If for any attackers, $Adv_{\Sigma}^{EUF\text{-}CMA}$ is negligible, Σ is said to be EUF-CMA secure.

⓪ Security Models

Sometimes it is helpful to consider models where some tools (primitives) used by cryptographic schemes such as,

① Hash functions
② Block ciphers
③ Finite groups

are considered to be a ideal, that is, the adversary can only use (attack) them in a certain way.

⇒ Idealized Security Models:

① Hash function → Random oracle
② Block ciphers → Ideal cipher
③ Finite groups → Generic group

Standard model: no idealized primitives (sort of) ☆

- Random Oracle

Arguably the most used idealized model to prove security of practical schemes.

(Hash function $H: \{0,1\}^* \xrightarrow{pseudorandom} Rec(H)$) is analized as it were a perfectly random function.

- Each new query receives a random answer in $\boxed{Rec(H)}$
- The same query asked twice receives the same answer twice

But for actual scheme, H is replaced by cryptographic hash function (SHA-1, RIPEMD-160, ...)

$x_1 \rightarrow y_1$ $x_1 \mid y_1$ $H(x_1) = y_1$ SHA-1

$x_2 \rightarrow y_2$ $x_2 \mid y_2$ $H(x_2) = y_2$ SHA-2

$x_i \xrightarrow{?}$ $x_2 = x_i$ $H(x_i) = y_2$

Examples of use:

① Signature Schemes: Full-Domain Hash, Schnorr
② Encryption schemes: OAEP-based constructions

Somehow controversial: not really proof, only heuristic

⓪ Full-Domain Hash Signatures

Scheme FDH is (K, S, V) as follows

$Ver(pk, \sigma, m) = f(\sigma, m)$ (σ: signature, m: message)

$Sign(sk, m) = f^{-1}(m)$ fixed length

- K: Key Generation returns (f, f^{-1}) where
  - ① Public key $f: X \rightarrow X$, a trapdoor one-way permutation onto X — efficient algorithm of f; 1G Byte file, 128 bits message
  - ② Private key f^{-1} cryptograph on one-way function (everyone can do with pk)
- S: Signature of m, returns $\sigma \leftarrow \boxed{f^{-1}(H(m))}$ f can decrypt σ as H(m)
- V: Verification of (m, σ), returns true if $f(\sigma) = H(m)$

Existential Unforgenability – Chosen Message Attack; encryption of m by sk

Theorem (FDH is $\boxed{\text{EUF-CMA}}$ in the RO model) (by owner of sk) — Random oracle

Let FDH be the FDH signature scheme using one-way permutation f (ex: f = RSA) — the approach to prove the scheme

For each adversary A there exist an adversary B such that $Adv_{FDH}^{euf\text{-}cma}(A) \leq (q_h + q_s + 1) \cdot Adv_{f}^{OW}(B)$ where (negligible)

① A runs in time t, makes q_h queries to hash function (RO), and q_s signature queries. — polynomial variables

② T_f is the time to compute f (in the forward direction) — verification function for FDH signature scheme

③ B runs in time $t' = t + (q_h + q_s) \cdot T_f$ — B against FDH; t: time by A to break FDH; simulation time to interact with A by B

Proof: We use a game-based proofs technique:

① Define sequence of games $G_0, G_1, \ldots, G_5$ of games or experiments
② All games in the same probability space (No additional advantage gained by A)
③ Rules on how the view of the game is computed differs (Ideal → Real)
④ Successive games are very similar, typically with slightly different

distribution probabilities

⑤ G_0 is the actual security game (EUF-CMA)

⑥ G_5 is the game for the underlying assumption (OW)

⑦ We relate the probabilities of the events that define the advantages in G_0 and G_5, via all the intermediate games.

Game G_0: the real euf-cma game with signing oracle and a random oracle, but we also provide a verification oracle Vf.

Verification oracle $Vf(m, \sigma)$: Return true if $H(m) = f(\sigma)$. The game ends when adversary sends (m, σ) here.

Let S_0 be the event: "A outputs a pair (m, σ) for which Vf returns true"

Clearly, $Adv^{euf\text{-}cma}_{FDH}(A) = Pr[S_0]$

Game G_1: as G_0 but oracles are simulated as below.

Hashing oracle $H(q)$: Create an initially empty list called H-List.

$q' \to (q', *, r') \to r'$, $q \to (q, *, r) \to r$ H-List

→ If $(q, *, r) \in$ H-List, return r.

→ Otherwise reply using Rule $H^{(1)}$: $r \xleftarrow{\$} X$, and add record $(q, *, r)$ to H-List. (signature)

(Ideal)

Signing oracle $S(m)$: $r \leftarrow H(m)$. Reply using Rule $S^{(1)}$: $\sigma \leftarrow f^{-1}(r)$.

(Ideal)

Verification oracle $Vf(m, \sigma)$: $r \leftarrow H(m)$ Return true if $r = f(\sigma)$

Game ends when oracle called.

Let S_1 be the event: "Vf returns true in G_1". Clearly $Pr[S_1] = Pr[S_0]$. The number of querying Hash oracle / The number of querying signing oracle / occasion of forging a signature

Game G_2: as G_1 but where ① $\boxed{c \xleftarrow{\$} \{1, \dots, q_H + q_S + 1\}}$

② Let c' = index of first query where message m' (the one for which A outputs a forgery) was sent to the hashing oracle by A.

③ If $c \neq c'$, then abort. (target: $c = c'$)

Success verification is within the game ⇒ the adversary must query his output message m.

$$Pr[S_2] = Pr[S_1 \wedge GoodGuess] = \underbrace{Pr[S_1 \mid GoodGuess]}_{\geq Pr[S_1]} \times \boxed{Pr[GoodGuess]}$$

$$\geq Pr[S_1] \times \frac{1}{q_H + q_S + 1}$$

Game G_3: as G_2 but now use the following rule in the hashing oracle. ① Let y (hard problem) be the challenge from which we want to extract a preimage x by f. ($H(m) = y$, $f^{-1}(y) = x$, consider $f^{-1}(H(m)) = \sigma$, $f(\sigma) = H(m)$)

② Rule $H^{(3)}$: If this is the c-th query, set $r \leftarrow y$. Otherwise, choose random. Add record $(q, \perp, r)$ to H-List. $r_i \leftarrow H(m_i)$

H-List: $q_i \to (q_i, \perp, r_i) \to r_i$

Since position y is chosen uniformly at random: $\boxed{Pr[S_3] = Pr[S_2]}$

($f^{-1}(y) = H(m)^d \bmod N = y^d \bmod N$, $f(x) = (y^d)^e \bmod N = y$)

Game G_4: as G_3 but modify simulation of hashing oracle (which may be used in signing queries)

Rule $H^{(4)}$: If this is the c-th query, set $r \leftarrow y$ and $s \leftarrow \perp$

Otherwise, choose random $s \xleftarrow{\$} X$, compute $r \leftarrow f(s)$

Add record (q, s, r) to H-List ($s^e = r = H(m)$, $(H(m)^d)^e = H(m)$)

Since position y is random, f is permutation, and s is random:

$Pr[S_4] = Pr[S_3]$ ($f^{-1}(y) = s$, $s^e = H(m)$; $H(m)^d$, $(H(m)^d)^e$) Note: $(H(m)^d)^e = H(m)$

Normal Procedure of FDH Signature Scheme

$m \to$ Signing Oracle $\to s = f^{-1}(r)$; $r = H(m) \to f^{-1}(r)$; $f(s) = f(f^{-1}(r)) = r$

$m \to$ ①② (m, s, r) input

Game G_5: except for the c-th query, all $\boxed{\text{preimage are without } f^{-1}}$ known. Then, we can simulate signing oracle

Rule $S^{(5)}$: Lookup (m, s, r) in H-List, and set $\boxed{\sigma \leftarrow s}$ ($H(m)^d = \sigma$)

Since c-th query cannot be asked to hash oracle, then $Pr[S_5] = Pr[S_4]$. (① $s \xleftarrow{\$} X$ ② $f(s) = r$, $(H(m)^d)^e = H(m)$ ③ (m, s, r))

Moreover, simulation can be done computing $\boxed{(q_S + q_H)}$ evaluations of f. ($f(x) = y$)

Signature foregy for y gives preimage for y: (unknown)

$Pr[S_5] = Adv^{ow}_f(B)$, where B = G_5 runs in time $t + (q_S + q_H)T_f$ ($f^{-1}(y) = x$; algorithm (simulator); time for A breaks f; simulation time)

Combining the relations from previous games.

$$\boxed{Adv^{ow}_f(B)} = Pr[S_5] = Pr[S_4] = Pr[S_3] = Pr[S_2]$$

$$(\to 0) \quad \geq \frac{1}{q_H + q_S + 1} \times Pr[S_1]$$

$$\geq \frac{1}{q_H + q_S + 1} \times Pr[S_0]$$

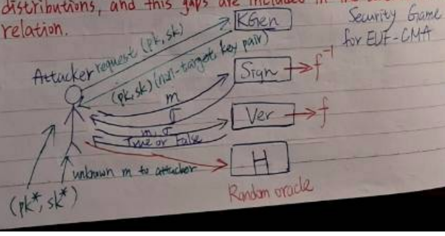
$$= \frac{1}{q_H + q_S + 1} \times \boxed{Adv^{euf\text{-}cma}_{FDH}(A)}$$

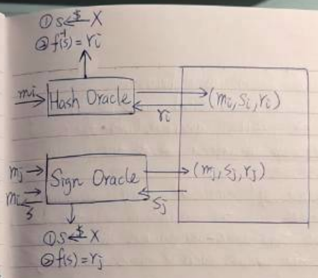
$Adv^{ow}_f(B) \geq \frac{1}{q_H + q_S + 1}$

$Adv^{ow}_f(B) \times (q_H + q_S + 1) \geq Adv^{euf\text{-}cma}_{FDH}$

negligible $= \frac{1}{\text{superpoly}(n)}$ $(\to 0)$

Game-playing proofs: In general, games can have different distributions, and this gaps are included in the concrete security relation.





⑨ Full-Domain Hash: Interpreting the result

Suppose feasible security bounds for any adversary are:

① at most 2^{75} operations (t),

② at most 2^{55} hash queries (q_H), and

③ at most 2^{30} signing queries (q_S)

$Adv^{euf\text{-}cma}_{FDH}(A) \leq (q_H + q_S + 1) \cdot Adv^{ow}_f(B)$

B runs in time $t' = t + (q_H + q_S) \cdot T_f$

Interpreting the result: If one can break the scheme with time t, then one can invert f within time $t' \leq (q_H + q_S + 1)(t + (q_H + q_S) \cdot T_f) \leq 2^{130} + 2^{110} \cdot T_f$

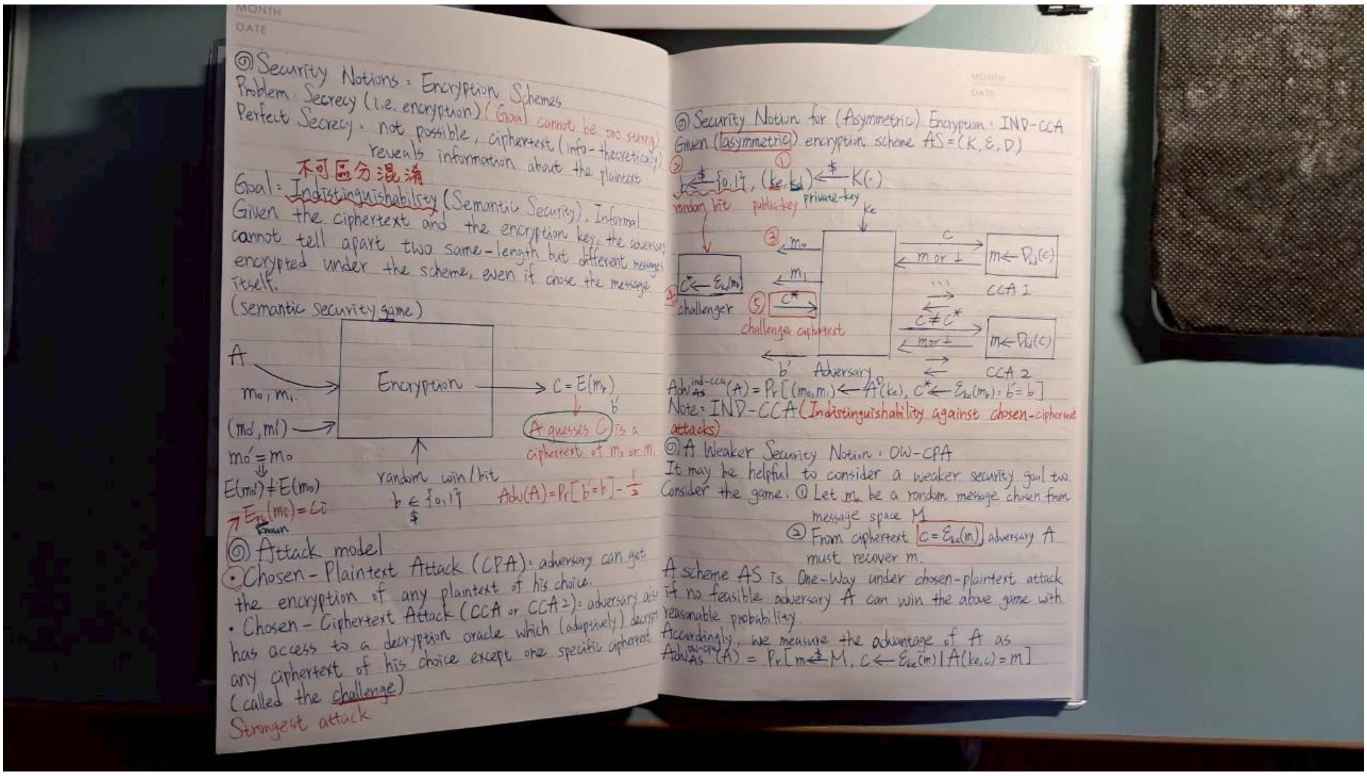
Thus, inverting f can be done in time $t' \leq 2^{130} + 2^{110} \cdot \boxed{T_f}$

Recall that $T_f = O(k^3)$ operations, if $k = |n|$ and e small. (key is of k-bit and the time complexity of calculation f is $O(k^3)$)

We compare it with known bounds on inverting RSA (namely, factoring using the best known inverting algorithm, the Number Field Sieve (NFS)) for f = RSA ($k = 1024 = 2^{10}$, $k^3 = 2^{30}$)

- 1024 bits → $t' \leq 2^{160}$ ... but NFS takes 2^{80} } insecure
- 2048 bits → $t' \leq 2^{143}$ ... but NFS takes 2^{111} } insecure
- 4096 bits → $t' \leq 2^{146}$ ... but NFS takes 2^{149} → secure!

⇒ RSA-FDH is secure for keys at least 4096!



MONTH
DATE

◎ Security Notions : Encryption Schemes

Problem: Secrecy (i.e. encryption) (Goal cannot be too strong)

Perfect Secrecy: not possible, ciphertext (info-theoretically) reveals information about the plaintext

不可區分混淆

Goal: Indistinguishability (Semantic Security). Informal

Given the ciphertext and the encryption key, the adversary cannot tell apart two same-length but different messages encrypted under the scheme, even if chose the message itself.

(semantic security game)

A

m_0, m_1

(m_0', m_1')

Encryption

$C = E(m_b)$

b'

A guesses C is a ciphertext of m_0 or m_1

random win/bit

$b \overset{\$}{\in} \{0,1\}$

$Adv(A) = P_r[b' = b] - \frac{1}{2}$

$m_0' = m_0$

$\Downarrow$

$E(m_0') \neq E(m_0)$

$E_{ke}(m_i) = C_i$

known

◎ Attack model

- Chosen-Plaintext Attack (CPA): adversary can get the encryption of any plaintext of his choice.
- Chosen-Ciphertext Attack (CCA or CCA2): adversary also has access to a decryption oracle which (adaptively) decrypt any ciphertext of his choice except one specific ciphertext (called the challenge)

Strongest attack

MONTH
DATE

◎ Security Notion for (Asymmetric) Encryption : IND-CCA

Given ($\boxed{\text{asymmetric}}$) encryption scheme $AS = (K, \mathcal{E}, D)$

② $b \overset{\$}{\leftarrow} \{0,1\}$, ① $(k_e, k_d) \overset{\$}{\leftarrow} K(\cdot)$

random bit public-key private-key

k_e

③ m_0

m_1

④ $C^* \leftarrow \mathcal{E}_{k_e}(m_b)$

challenger

⑤ $\boxed{C^*}$

challenge ciphertext

b' Adversary

c

m or $\perp$

$m \leftarrow D_{kd}(c)$

CCA 1

$c \neq c^*$

m or $\perp$

$m \leftarrow D_{kd}(c)$

CCA 2

$Adv_{AS}^{ind\text{-}cca}(A) = P_r[(m_0, m_1) \leftarrow A^D(k_e), C^* \leftarrow \mathcal{E}_{k_e}(m_b) : b' = b]$

Note: IND-CCA (Indistinguishability against chosen-ciphertext attacks)

◎ A Weaker Security Notion : OW-CPA

It may be helpful to consider a weaker security goal too.

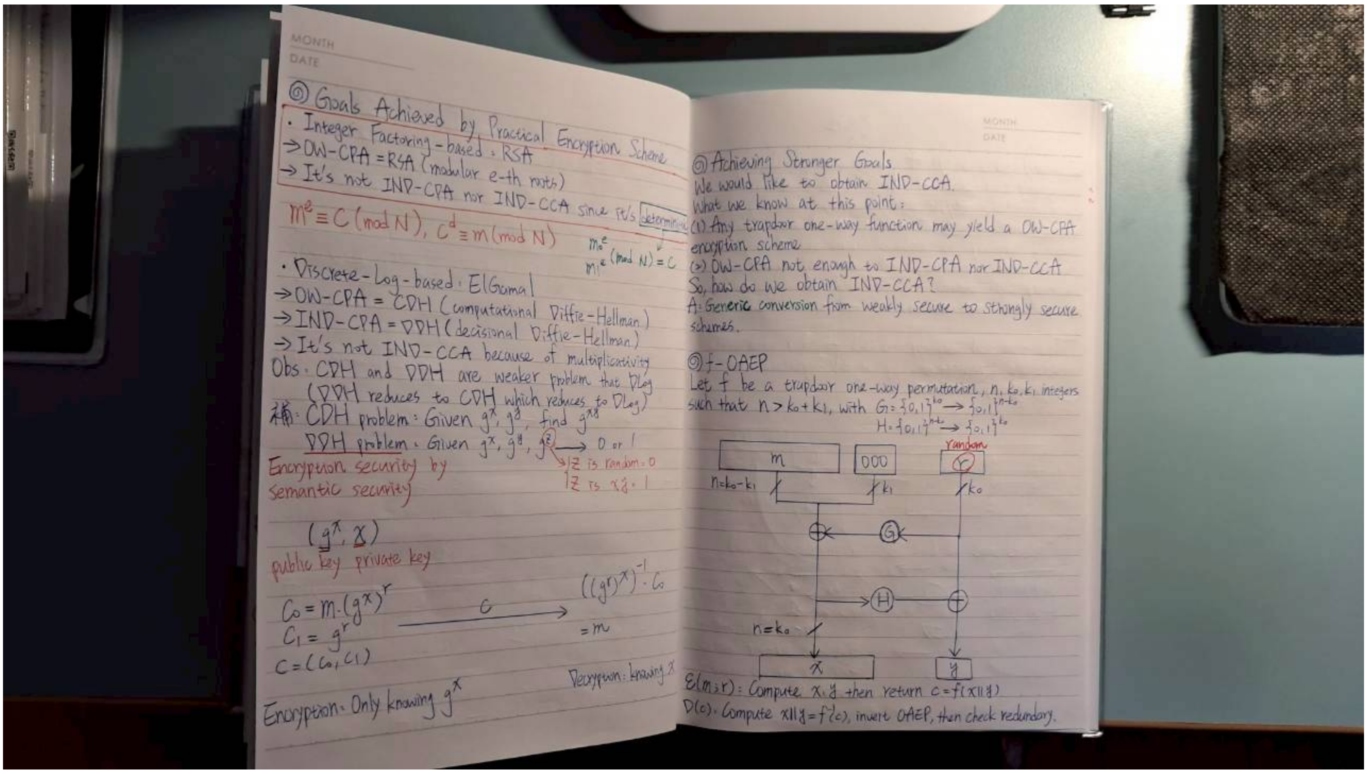
Consider the game: ① Let m be a random message chosen from message space M

② From ciphertext $\boxed{c = \mathcal{E}_{k_e}(m)}$ adversary A must recover m.

A scheme AS is One-Way under chosen-plaintext attack if no feasible adversary A can win the above game with reasonable probability.

Accordingly, we measure the advantage of A as

$Adv_{AS}^{ow\text{-}cpa}(A) = P_r[m \overset{\$}{\leftarrow} M, c \leftarrow \mathcal{E}_{k_e}(m) \mid A(k_e, c) = m]$



MONTH
DATE

◎ Goals Achieved by Practical Encryption Scheme

- Integer Factoring-based : RSA

→ OW-CPA = RSA (modular e-th roots)

→ It's not IND-CPA nor IND-CCA since it's $\boxed{\text{deterministic}}$

$m^e \equiv C \pmod N$, $C^d \equiv m \pmod N$

m_0^e

$m_1^e \pmod N = C$

- Discrete-Log-based : ElGamal

→ OW-CPA = CDH (computational Diffie-Hellman)

→ IND-CPA = DDH (decisional Diffie-Hellman)

→ It's not IND-CCA because of multiplicativity

Obs: CDH and DDH are weaker problem that DLog (DDH reduces to CDH which reduces to DLog)

補: CDH problem: Given g^x, g^y, find g^{xy}

DDH problem: Given $g^x, g^y, g^z \longrightarrow 0$ or 1

z is random : 0

z is xy : 1

Encryption security by semantic security

(g^x, x)

public key private key

$C_0 = m \cdot (g^x)^r$

$C_1 = g^r$

$C = (C_0, C_1)$

$\xrightarrow{C}$

$((g^r)^x)^{-1} \cdot C_0$

$= m$

Encryption: Only knowing g^x

Decryption: knowing x

MONTH
DATE

◎ Achieving Stronger Goals

We would like to obtain IND-CCA.

What we know at this point:

(1) Any trapdoor one-way function may yield a OW-CPA encryption scheme

(2) OW-CPA not enough to IND-CPA nor IND-CCA

So, how do we obtain IND-CCA?

A: Generic conversion from weakly secure to strongly secure schemes.

◎ f-OAEP

Let f be a trapdoor one-way permutation, n, k_0, k_1 integers such that $n > k_0 + k_1$, with $G: \{0,1\}^{k_0} \rightarrow \{0,1\}^{n-k_0}$

$H: \{0,1\}^{n-k_0} \rightarrow \{0,1\}^{k_0}$

random

m

000

r

$n-k_0-k_1$

k_1

k_0

G

H

$n-k_0$

x

y

$\mathcal{E}(m; r)$: Compute x, y then return $c = f(x \| y)$

$D(c)$: Compute $x \| y = f^{-1}(c)$, invert OAEP, then check redundancy.