The Missing Lecture of Your Cryptography

junyu33

2024/11/21

Review of Probability Theory

DEF: We define S^n as a vector (concatenation) of elements in set S, for example:

- (010) is an element in $\{0,1\}^3$

So there are $|S|^n$ elements in S^n, where $|A|$ ($\#A$) repesents the number of elements in set A (cardinality).

DEF: **Probability distribution** P over universe U (finite) is a function P, which satisfies:

- $\forall x \in U, P(x) \geq 0$
- $\sum_{x\in U} P(x) = 1$

for example:

1. uniform distribution: $\forall x \in U, P(x) = \frac{1}{|U|}$
2. point distribution at x_0: $P(x_0) = 1, \forall x \neq x_0, P(x) = 0$

QUESTION: please describe the probability distribution when you roll up a dice.

DEF: For a set $A \subseteq U$, $Pr[A] = \sum_{x \in A} P(x) \in [0, 1]$, the set A is called an **event**:

for example:

1. $Pr[U] = 1$
2. If $U = \{0,1\}^3$ and $x \in U$, $Pr[lsb_2(x) = 11] = \frac{1}{4}$
3. For events A_1 and A_2, $Pr[A_1 \cup A_2] \leq Pr[A_1] + Pr[A_2]$

DEF: A **ramdom varible** X is a function $X\colon U \to V$, for example:

$X : \{0,1\}^n \to \{0,1\}$, $X(y) = lsb(y)$, we have:

$Pr[X = 0] = Pr[X = 1] = \frac{1}{2}$

DEF: Specifically, a **uniform ramdom varible** is a ramdom varible which satisfies:

$r \overset{R}{\leftarrow} U, Pr[r = a] = \frac{1}{|U|}$

QUESTION: please write two uniform ramdom varibles where $U = \{1, 2, 3, 4, 5\}$

Perfect Secrecy

DEF: A cipher (E, D) over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ has **perfect secrecy** if:

$$\forall m_0, m_1 \in \mathcal{M}, len(m_0) = len(m_1) \wedge \forall c \in \mathcal{C}$$

$$Pr[E(k, m_0) = c] = Pr[E(k, m_1) = c]$$

where k is uniform in $\mathcal{K}$, i.e. ($k \overset{R}{\leftarrow} \mathcal{K}$)

The bad news is to satisfy this condition, we must have $|\mathcal{K}| \geq |\mathcal{M}|$, which makes it hard to use in practice.

As we have learned eariler, OTP (One-Time-Pad) is an example of cipher scheme which satisfies perfect secrecy.

QUESTION: Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1, 2, \cdots, 255\}$ and consider the following cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

$$E(k, m) = m + k \pmod{256}; D(k, c) = c - k \pmod{256}$$

Does this cipher have perfect secrecy?

Pseudo Random Generator (PRG)

So what will happen if we expand the ciphertext from OTP? Then it comes to the PRG:

DEF: In *Foundations of Cryptography: Basic Tools, page 113*, we define pseudo random generator G as this:

> A pseudorandom generator is a deterministic polynomial-time algorithm G satisfying the following 2 conditions:
>
> 1. There exists a function $l : \mathbb{N} \to \mathbb{N}\ \mathrm{s.t.}\ \forall n \in \mathbb{N}, l(n) > n\ \mathrm{and}\ \forall s \in \{0,1\}^*, |G(s)| = l(|s|)$
> 2. Pseudorandomness: The ensemble $\{G(U_n)\}_{n \in \mathbb{N}}$ is pseudorandom (unpreditable).
>
> The function l is called the expansion factor of G, and the input s to the generator is called its seed.
>
> (Mention that $|x|$ means the length for x, not the cardinality.)

One practical example of PRG is stream cipher, like RC4 and Salsa20, while the former one is found to have bias in initial output.

Anyone intrested in this, please see details in paper *Statistical Analysis of the Alleged RC4 Keystream Generator, FSE 2001*.

Predictability (Pseudorandomness)

As I said eariler, it is not practical to construct an encryption scheme which satisfies perfect secrecy. However, that doesn’t mean an encryption scheme which doesn’t have perfect secrecy is not secure, due to the limited power of adversary in practice.

To address security in cryptography, we need to clarify a set of security defnitions such as predictability and distinguishability (in the next slide):

DEF: We say that $G : K \to \{0,1\}^n$ is **predictable** if:

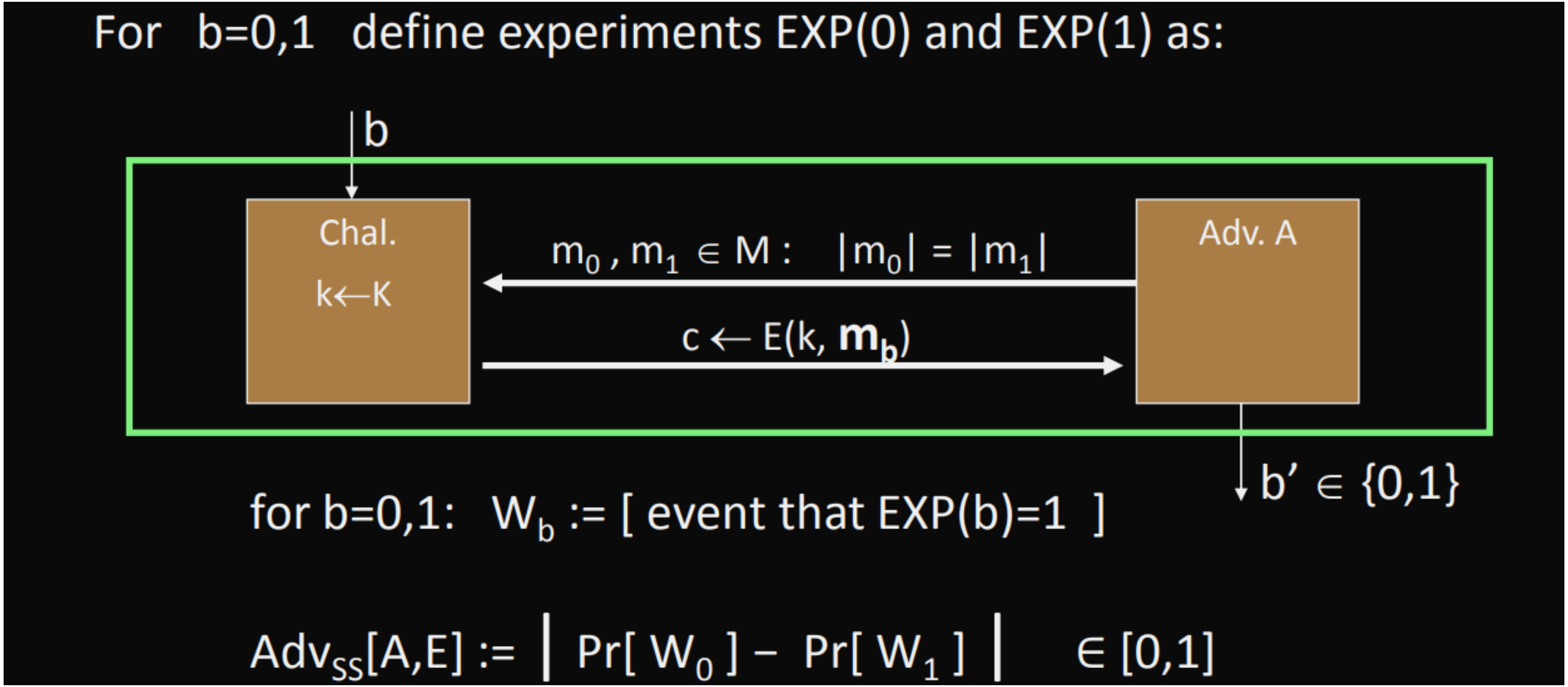
$$\exists \text{eff alg. } A \text{ and } \exists_{0 \leq i \leq n-1} \text{ s.t.}$$

$$Pr_{k \overset{R}{\leftarrow} \mathcal{K}}[A(G(k))|_{1,\cdots,i} = G(k)|_{i+1}] > \frac{1}{2} + \varepsilon$$

for non-negligible ε (e.g. $\varepsilon = \frac{1}{2^{30}}$)

DEF: PRG is unpreditable if it is not predictable.

Semantic Security (Indistinguishability)



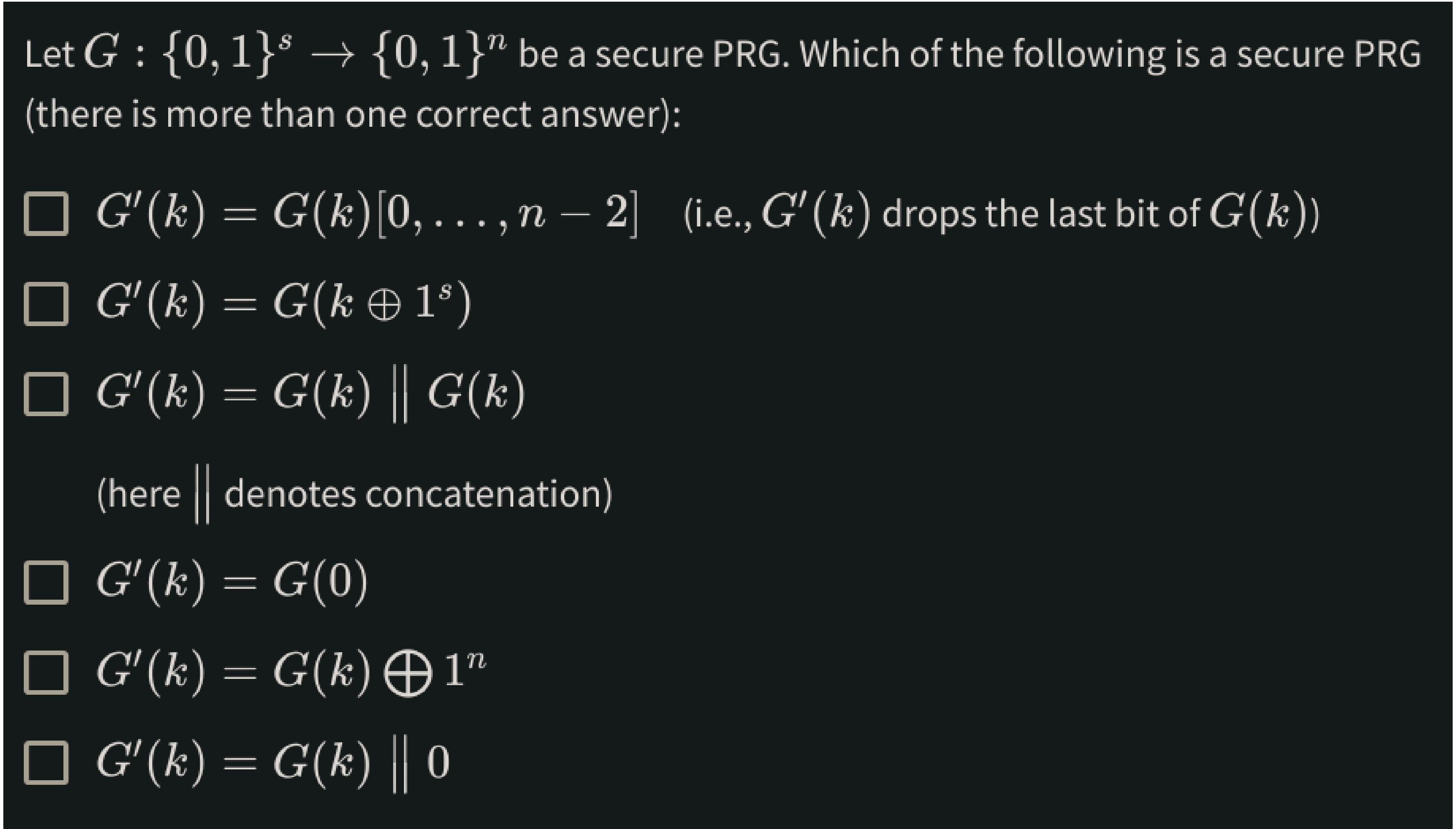
For b=0,1 define experiments EXP(0) and EXP(1) as:

b
Chal.
k⟵K
Adv. A
$m_0, m_1 \in M: \quad |m_0| = |m_1|$
$c \leftarrow E(k, \mathbf{m_b})$
$b' \in \{0,1\}$

for b=0,1: W_b := [event that EXP(b)=1]

$\text{Adv}_{SS}[A,E] := \left| \Pr[W_0] - \Pr[W_1] \right| \quad \in [0,1]$

DEF: E is semantically secure if for all efficient adversary A, $Adv_{ss}[A, E]$ is negligible.



Let $G : \{0,1\}^s \to \{0,1\}^n$ be a secure PRG. Which of the following is a secure PRG (there is more than one correct answer):

□ $G'(k) = G(k)[0, \ldots, n-2]$ (i.e., $G'(k)$ drops the last bit of $G(k)$)

□ $G'(k) = G(k \oplus 1^s)$

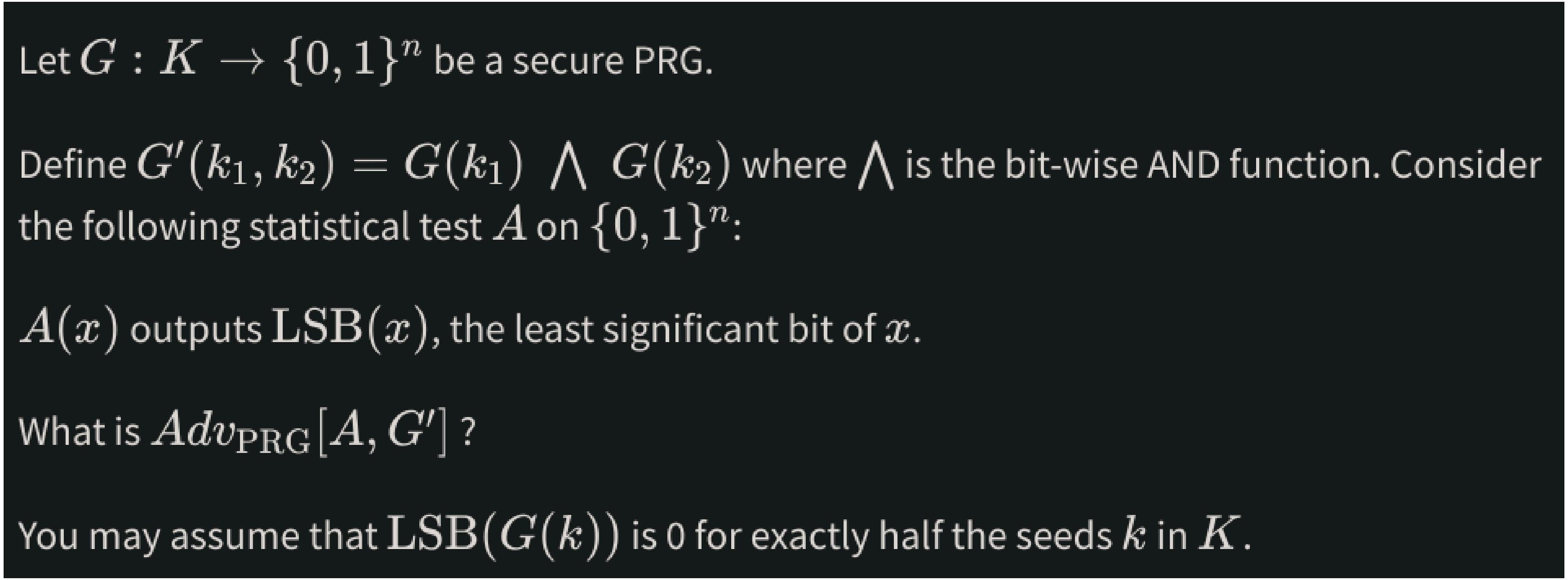
□ $G'(k) = G(k) \parallel G(k)$

(here $\parallel$ denotes concatenation)

□ $G'(k) = G(0)$

□ $G'(k) = G(k) \bigoplus 1^n$

□ $G'(k) = G(k) \parallel 0$



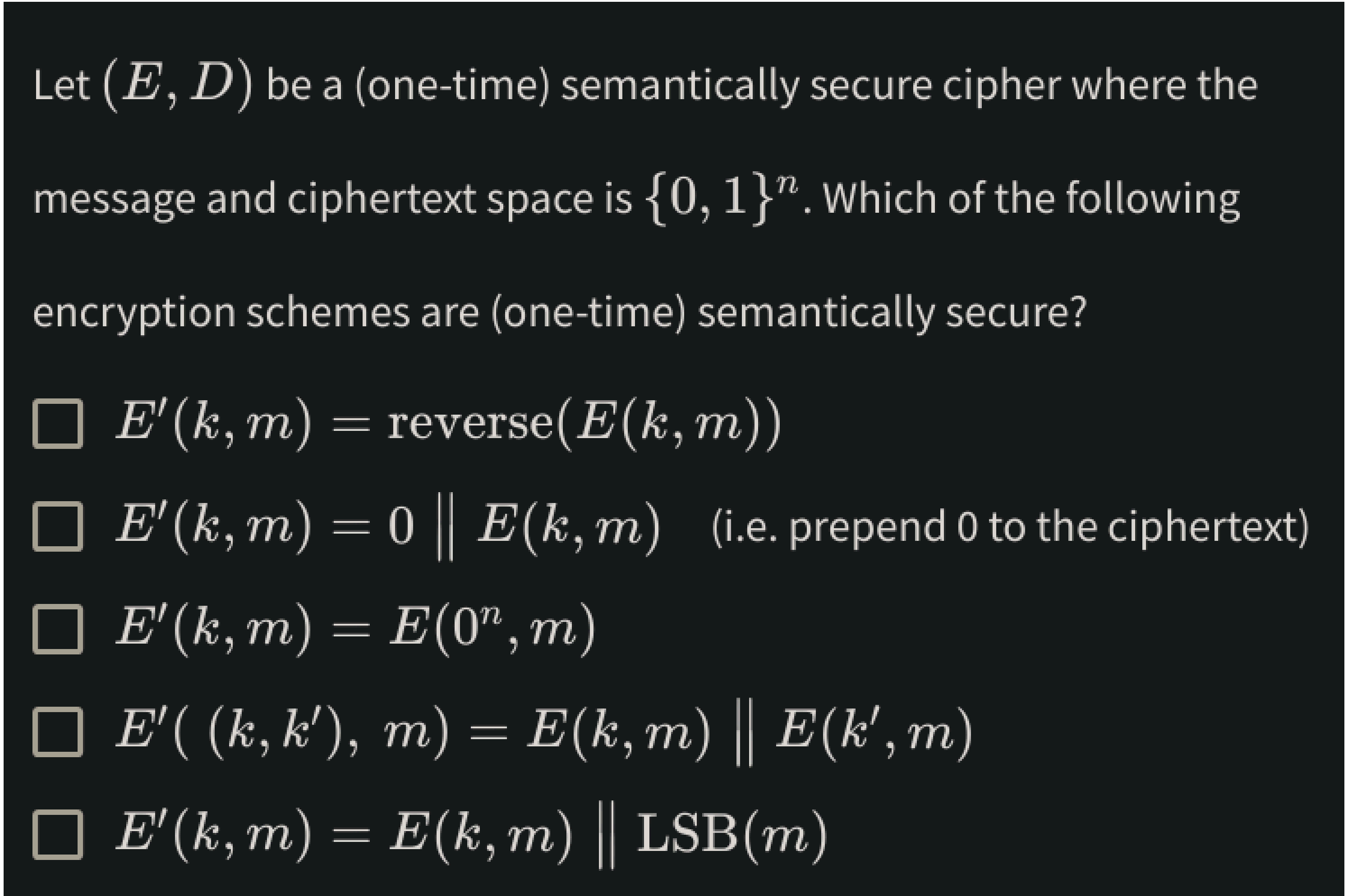
Let $G : K \rightarrow \{0,1\}^n$ be a secure PRG.

Define $G'(k_1, k_2) = G(k_1) \ \bigwedge \ G(k_2)$ where $\bigwedge$ is the bit-wise AND function. Consider the following statistical test A on $\{0,1\}^n$:

$A(x)$ outputs $\mathrm{LSB}(x)$, the least significant bit of x.

What is $Adv_{\mathrm{PRG}}[A, G']$?

You may assume that $\mathrm{LSB}(G(k))$ is 0 for exactly half the seeds k in K.



Let (E, D) be a (one-time) semantically secure cipher where the message and ciphertext space is $\{0,1\}^n$. Which of the following encryption schemes are (one-time) semantically secure?

□ $E'(k,m) = \text{reverse}(E(k,m))$

□ $E'(k,m) = 0 \,\|\, E(k,m)$ (i.e. prepend 0 to the ciphertext)

□ $E'(k,m) = E(0^n, m)$

□ $E'(\,(k,k'),\ m) = E(k,m) \,\|\, E(k',m)$

□ $E'(k,m) = E(k,m) \,\|\, \text{LSB}(m)$

PRF and PRP

In previous slides we discussed about PRG, which is widely used in block ciphers. Next we’ll introduce PRF (pseudo random function) and PRP (pseudo random permutation).

PRF is defined over (K, X, Y):

$$F : K \times X \to Y$$

such that exists "eff" algorithm to evalutate $F(k, x)$.

PRP is defined over (K, X):

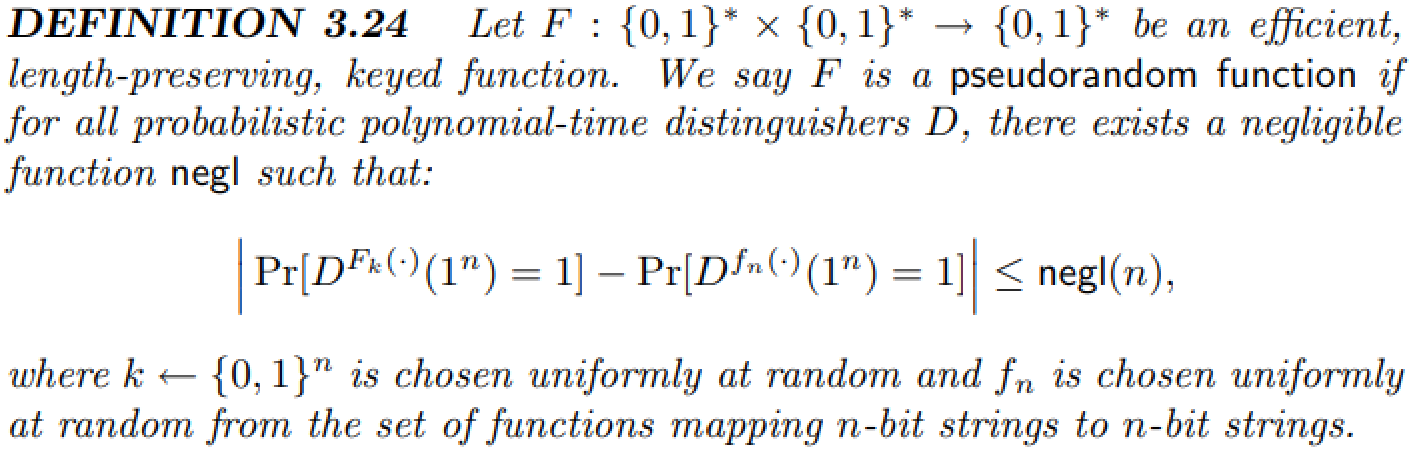
$$E : K \times X \to X$$

such that:

1. exists "eff" **deterministic** algorithm to evalutate $E(k, x)$.
2. $E(k, \cdot)$ is one-to-one, and exists "eff" inversion algorithm $D(k, y)$.

QUESTION: PRF can construct PRG easily, please provide an example.

Security requirements of PRF (*Introduction to Modern Cryptography, page 79*):



DEFINITION 3.24 *Let $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ be an efficient, length-preserving, keyed function. We say F is a* pseudorandom function *if for all probabilistic polynomial-time distinguishers D, there exists a negligible function* negl *such that:*

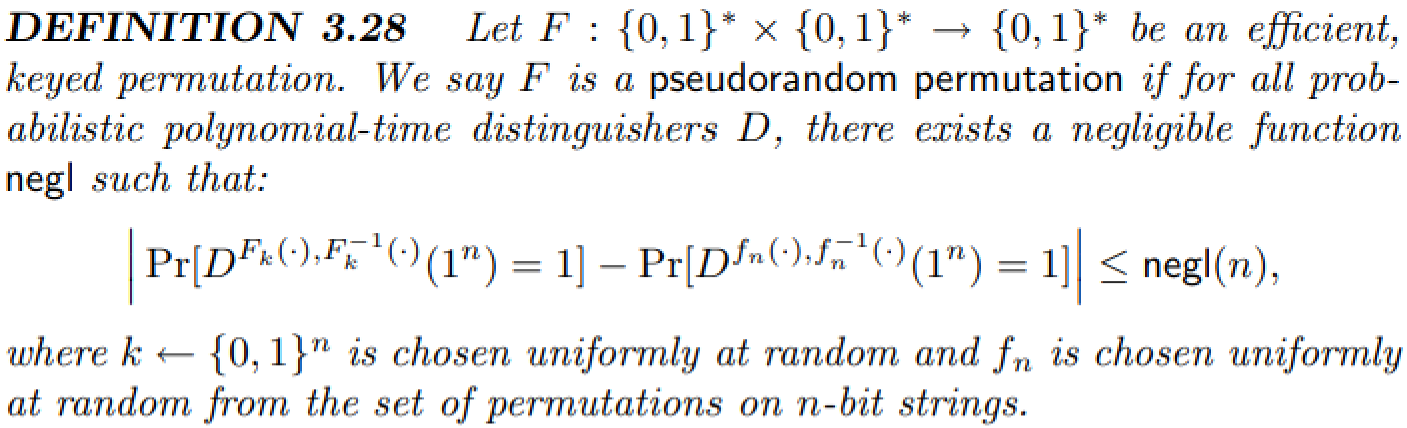
$$\left| \Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f_n(\cdot)}(1^n) = 1] \right| \leq \mathsf{negl}(n),$$

where $k \leftarrow \{0,1\}^n$ is chosen uniformly at random and f_n is chosen uniformly at random from the set of functions mapping n-bit strings to n-bit strings.

To be short, you can interpret $D(x)$ as binary classification function used in machine learning (only outputs 0 or 1), and $f_n(x)$ is a truly random function (TRF) chosen uniformly.

This equation means you can’t use any method to tell which is which (PRF and TRF), so the 1^n can be any n-bit string, and the right-hand side can be 0 or 1, as long as they are the same.

Security requirements of PRP (*Introduction to Modern Cryptography, page 80*):



DEFINITION 3.28 *Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, keyed permutation. We say F is a* pseudorandom permutation *if for all probabilistic polynomial-time distinguishers D, there exists a negligible function* negl *such that:*

$$\left| \Pr[D^{F_k(\cdot),F_k^{-1}(\cdot)}(1^n) = 1] - \Pr[D^{f_n(\cdot),f_n^{-1}(\cdot)}(1^n) = 1] \right| \leq \mathsf{negl}(n),$$

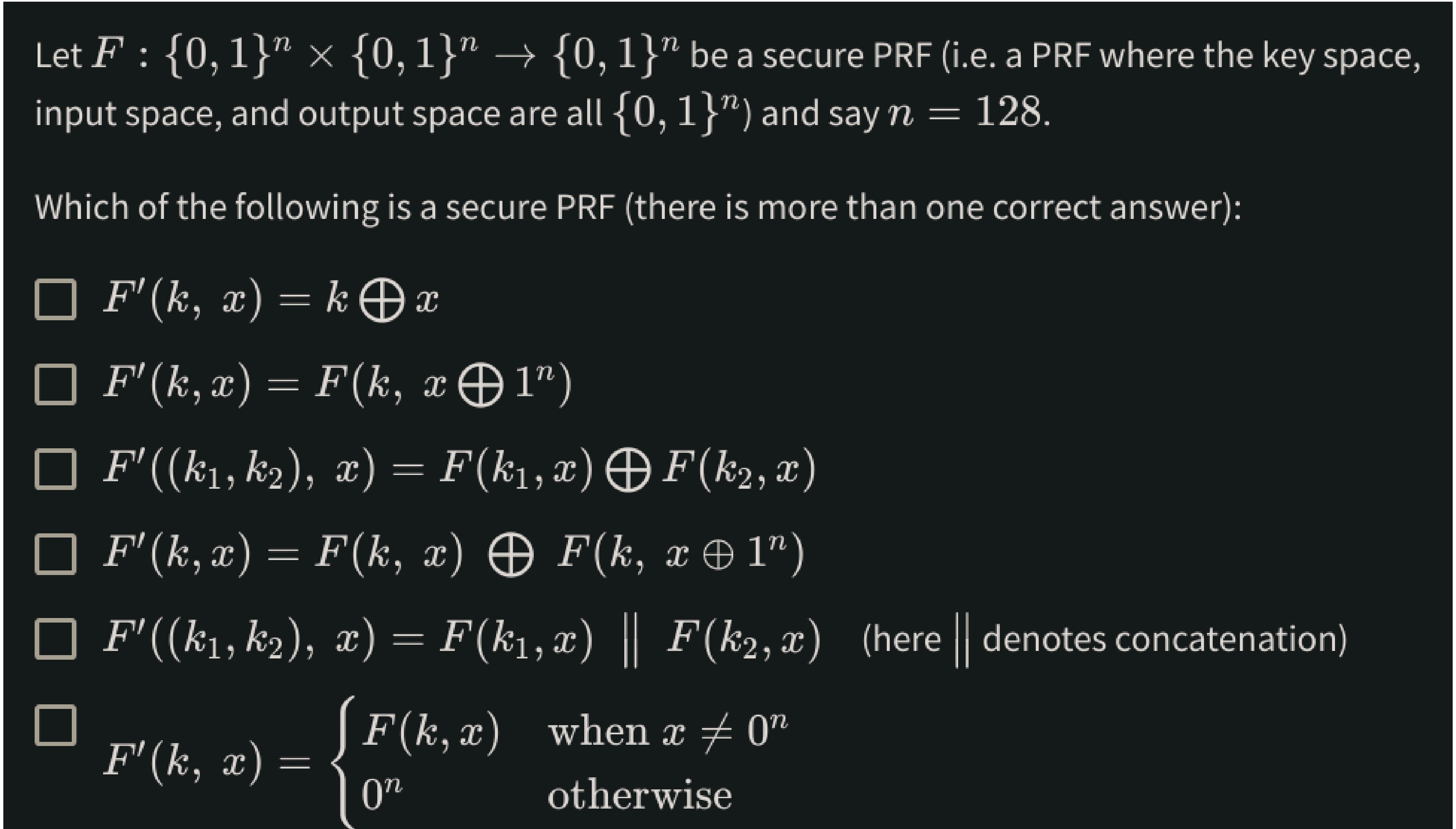
where $k \leftarrow \{0,1\}^n$ is chosen uniformly at random and f_n is chosen uniformly at random from the set of permutations on n-bit strings.

Just nearly the same as the PRF slide, since PRP is invertible, so we can add more constriants, like we may require the inverse function has the same property as the PRP.

Therefore, making the whole encrypt-decrypt procedure indistinguishable from the TRP is necessary.

Someone may say this is boring, but that’s what formality does.

QUESTION: Please write 2 real-world examples of secure PRP.



Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF (i.e. a PRF where the key space, input space, and output space are all $\{0,1\}^n$) and say $n = 128$.

Which of the following is a secure PRF (there is more than one correct answer):

□ $F'(k,\ x) = k \bigoplus x$

□ $F'(k, x) = F(k,\ x \bigoplus 1^n)$

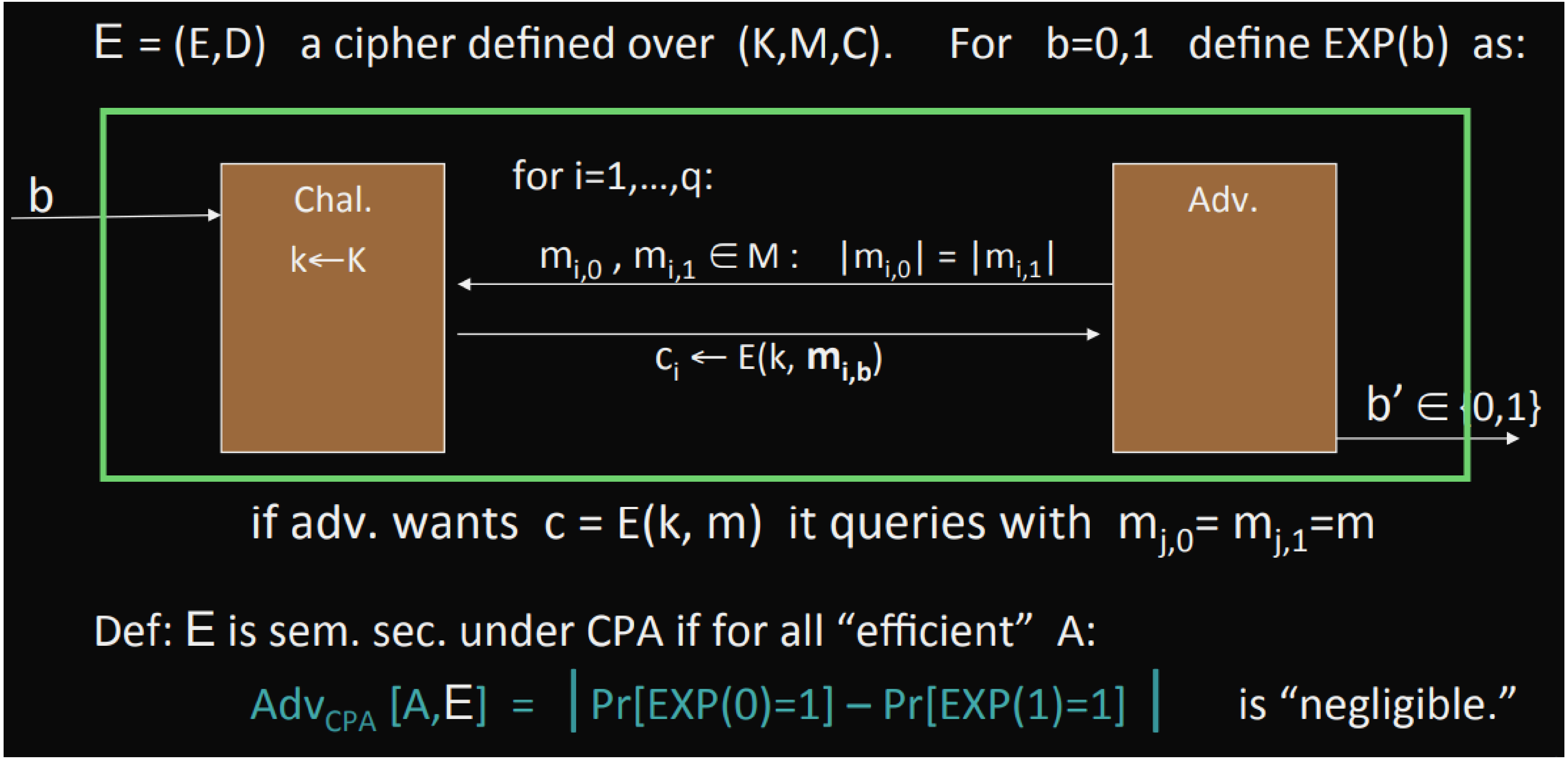
□ $F'((k_1, k_2),\ x) = F(k_1, x) \bigoplus F(k_2, x)$

□ $F'(k, x) = F(k,\ x)\ \bigoplus\ F(k,\ x \oplus 1^n)$

□ $F'((k_1, k_2),\ x) = F(k_1, x)\ \big\|\ F(k_2, x)$ (here $\big\|$ denotes concatenation)

□ $$F'(k,\ x) = \begin{cases} F(k, x) & \text{when } x \neq 0^n \\ 0^n & \text{otherwise} \end{cases}$$

CPA Security



$\mathbb{E}$ = (E,D) a cipher defined over (K,M,C). For b=0,1 define EXP(b) as:

b

Chal.

k←K

for i=1,…,q:

$m_{i,0}$, $m_{i,1}$ $\in$ M : $|m_{i,0}| = |m_{i,1}|$

$c_i \leftarrow E(k, \mathbf{m_{i,b}})$

Adv.

b′ $\in$ {0,1}

if adv. wants c = E(k, m) it queries with $m_{j,0}= m_{j,1}=m$

Def: $\mathbb{E}$ is sem. sec. under CPA if for all "efficient" A:

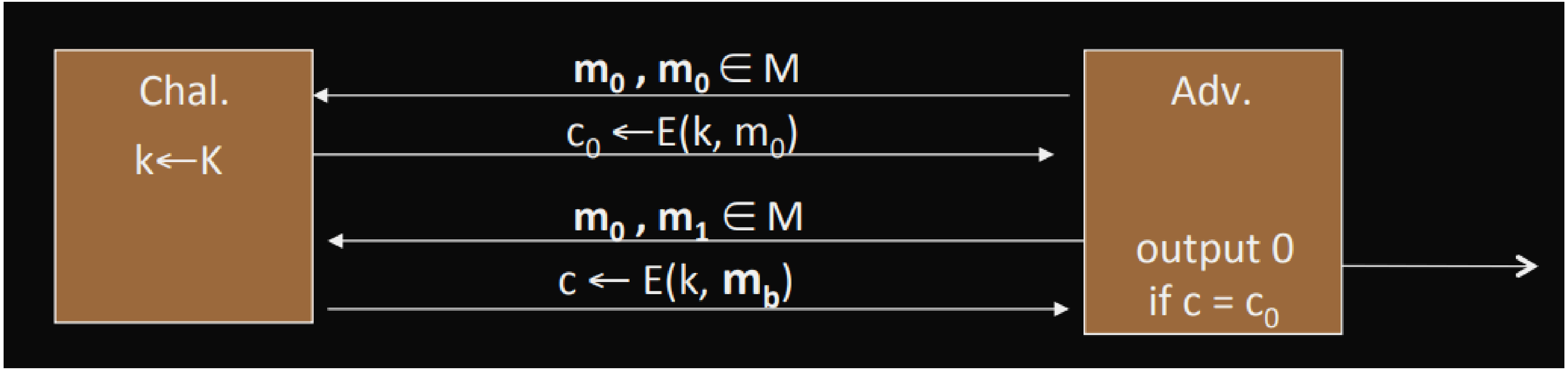
$\text{Adv}_{\text{CPA}}\,[A,\mathbb{E}] = \left|\Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1]\right|$ is "negligible."

As you can see, CPA security is semantic security for many-time key senario.

QUESTION: Does OTP satisfiy CPA security?

The answer is: NO!

In fact, all deterministic encryption algorithm don’t have CPA security. The proof is as follows:

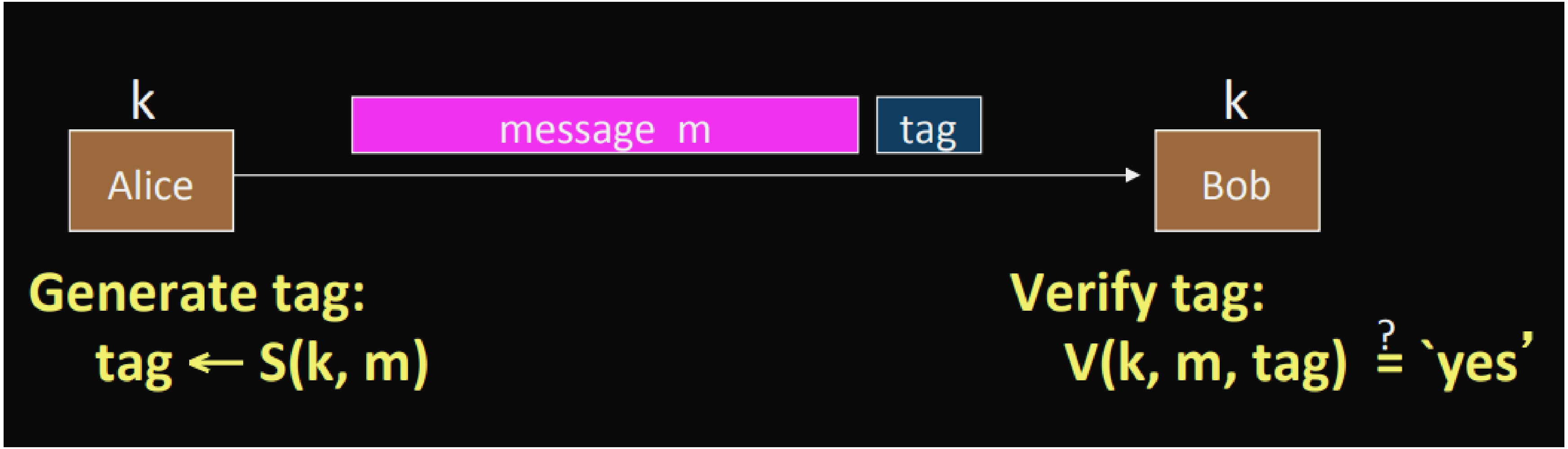

Chal.
$k \leftarrow K$
$\mathbf{m_0}$, $\mathbf{m_0} \in M$
$c_0 \leftarrow E(k, m_0)$
$\mathbf{m_0}$, $\mathbf{m_1} \in M$
$c \leftarrow E(k, \mathbf{m_b})$
Adv.
output 0
if $c = c_0$

So how to solve this?

1. Randomized encryption: encrypt the plaintext at random times, and decrypt according to entropy.
2. Nonce-based encryption: CBC, CFB, CTR mode.

Message Authentication Code (MAC)

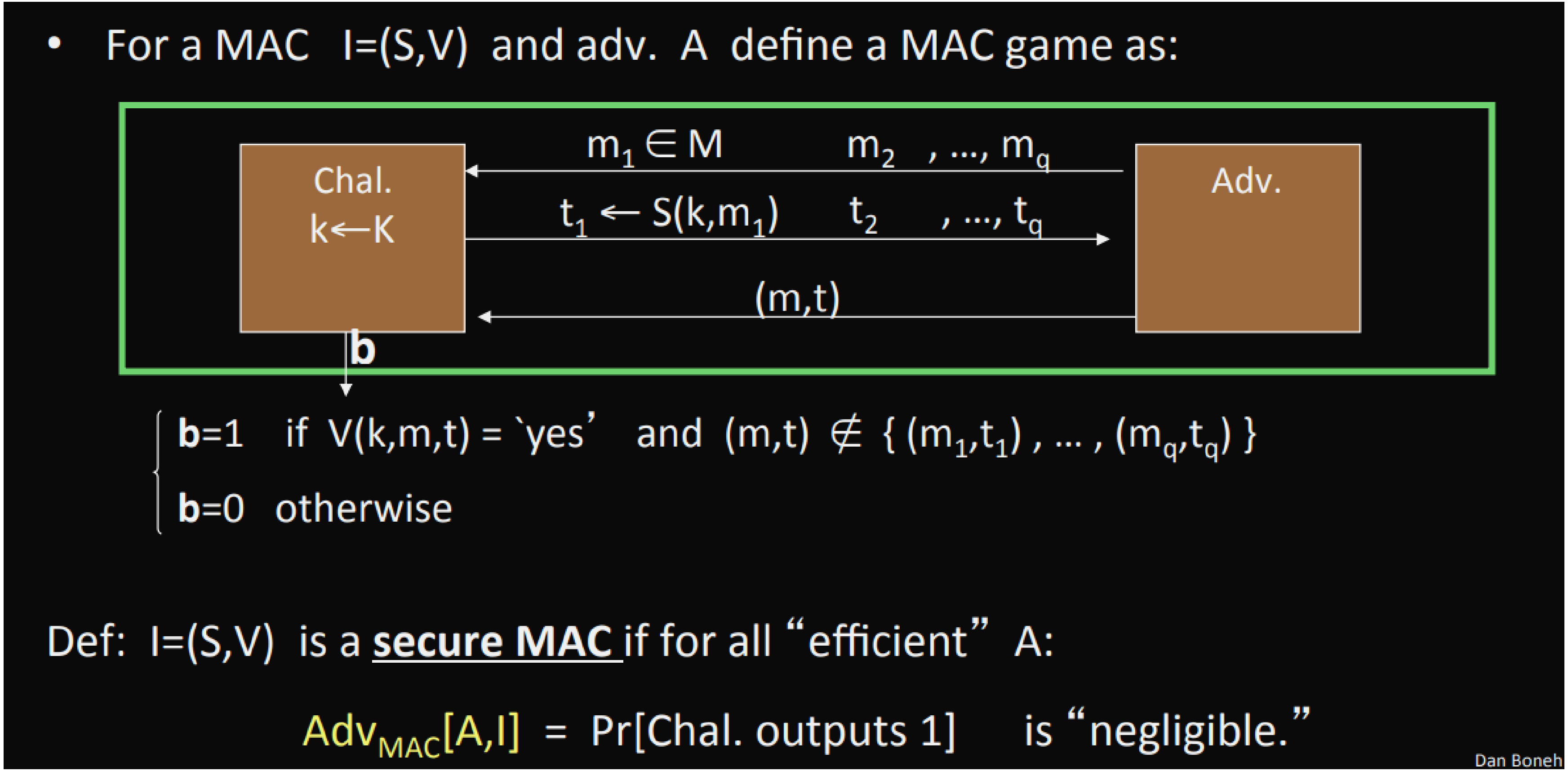
DEF: Message authentication code $I = (S, V)$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{T})$ is a pair of algs:



- $S(k, m)$ outputs t in $\mathcal{T}$
- $V(k, m, t)$ outputs "yes" or "no".

QUESTION: Can MAC be constructed by PRF?

And it comes to the security definition:



- For a MAC $I=(S,V)$ and adv. A define a MAC game as:

$$b = \begin{cases} 1 & \text{if } V(k,m,t) = \text{`yes'} \text{ and } (m,t) \notin \{ (m_1,t_1), \ldots, (m_q,t_q) \} \\ 0 & \text{otherwise} \end{cases}$$

Def: $I=(S,V)$ is a **secure MAC** if for all "efficient" A:

$$\mathrm{Adv}_{\mathrm{MAC}}[A,I] = \Pr[\text{Chal. outputs } 1] \quad \text{is "negligible."}$$

Dan Boneh

Hash Function

DEF: Let $H : M \to T$ be a hash function. ($|M| \gg |T|$)

A collsion for H is a pair $m_0, m_1 \in M \text{ s.t:}$

$$H(m_0) = H(m_1) \wedge m_0 \neq m_1$$

A function H is collsion resistant if for all "eff" algs. A:

$$Adv_{CR}[A, H] = Pr[A \text{ outputs collsion for } H] \leq \text{negl}(|M|)$$

Hash function only provide collsion resistance, not existential forgery. However we can use hash functions to build secure MACs (HMAC).

QUESTION: Is hash function a PRF?

Generic Birthday Attack

First let me introduce the birthday paradox:

DEF (birthday paradox): Let $r_i \in \{1, \cdots, B\}$ be independent identically distributed integers. $Pr[\exists i \neq j : r_i = r_j] \geq \frac{1}{2}$ when $n = 1.2 \times \sqrt{B}$.

Proof: (for uniform indep. r_i)

$$Pr[\exists i \neq j : r_i = r_j] = 1 - Pr[\forall i \neq j : r_i \neq r_j] = 1 - (\frac{B-1}{B})(\frac{B-2}{B}) \cdots (\frac{B-n+1}{B})$$

$$= 1 - \prod_{i=1}^{n-1}(1 - \frac{i}{B}) \geq 1 - \prod_{i=1}^{n-1}(e^{-i/B}) = 1 - e^{-\frac{1}{B}\sum_{i=1}^{n-1} i} \geq 1 - e^{\frac{n^2}{2B}}$$

Because $n = 1.2 \times \sqrt{B}$, $1 - e^{\frac{n^2}{2B}} = 1 - e^{0.72} = 0.53 > \frac{1}{2}$.

Q.E.D.

The reason why it is called a "paradox" is the number of people are much smaller than our expectation to achieve this probability. However, we can understand this by intuition:

N numbers have roughly N^2 pairs, so if $N^2 > B$, a collsion may appear.

Now we can introduce generic birthday attack from this paradox:

Generic alg. to find a collision in time $O(\sqrt{T})$ hashes:

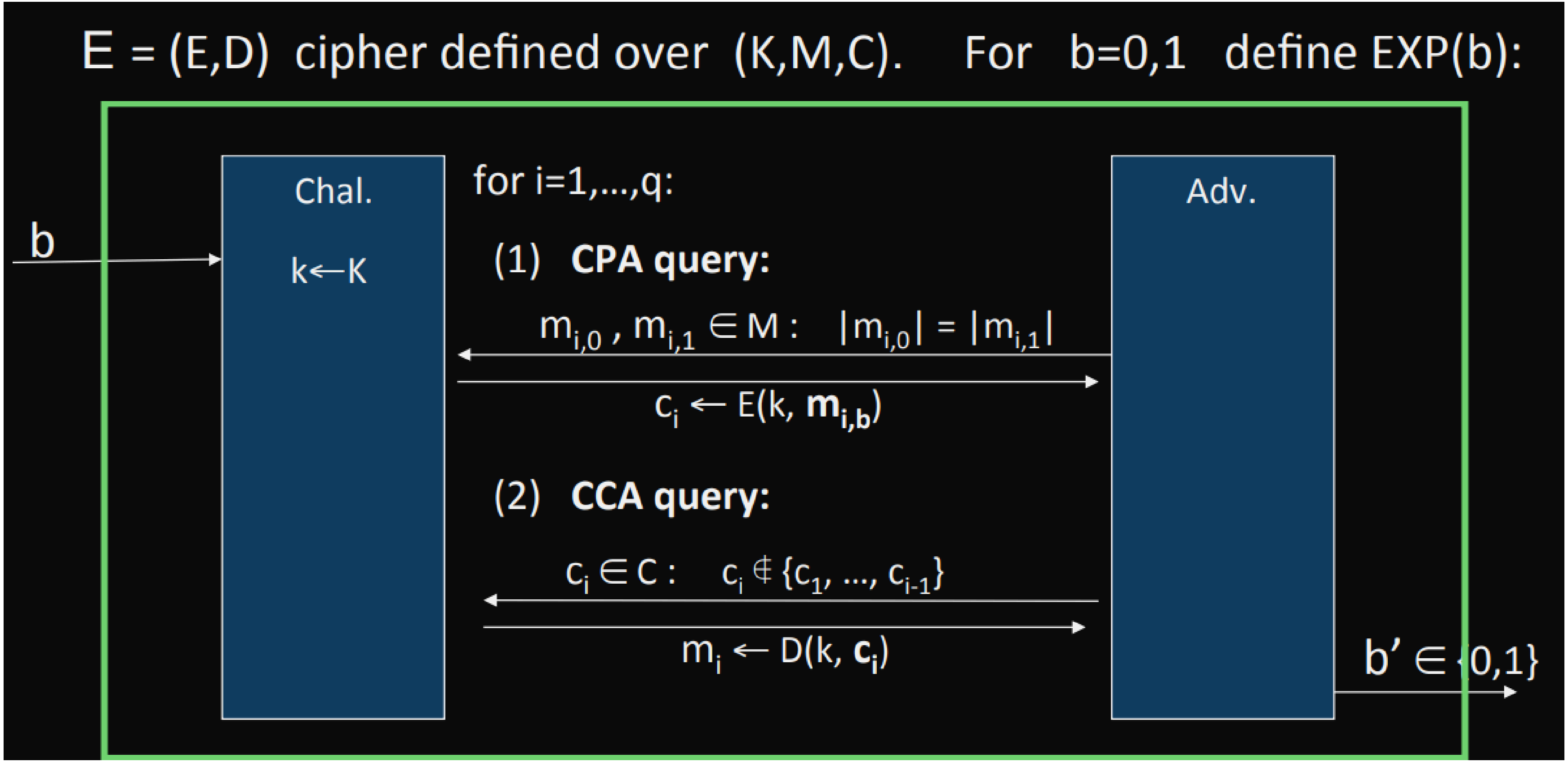
1. Choose $\sqrt{T}$ random messages in M: $\{m\}_{1,2,\cdots,\sqrt{T}}$.
2. Compute $t_i = H(m_i)$ for $i = 1, 2, \cdots, \sqrt{T}$.
3. Look for a collision $(t_i = t_j)$. If not found, go back to step 1.

QUESTION: What is the expected number of finding a collision using this algorithm?

HARDER QUESTION: How many integers (chosen from $[1, B]$) do you expect to choose so that the probability of finding a triple collision (having three same numbers) is higher than 50%?

TIME FOR A BREAK

CCA Security



$\mathcal{E}$ = (E,D) cipher defined over (K,M,C). For b=0,1 define EXP(b):

b

Chal.

k←K

for i=1,…,q:

(1) **CPA query:**

$m_{i,0}\,,\ m_{i,1} \in M: \quad |m_{i,0}| = |m_{i,1}|$

$c_i \leftarrow E(k, \mathbf{m_{i,b}})$

(2) **CCA query:**

$c_i \in C: \quad c_i \notin \{c_1, \ldots, c_{i-1}\}$

$m_i \leftarrow D(k, \mathbf{c_i})$

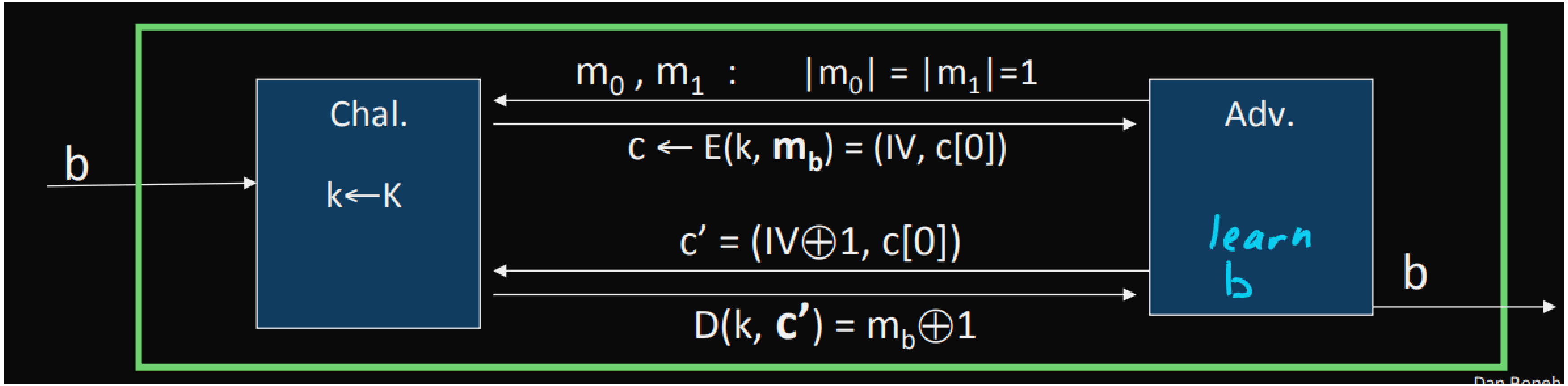
Adv.

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

E is CCA secure if for all "eff" A:

$$Adv_{CCA}[A, E] = |Pr[EXP(0) = 1] - Pr[EXP(1) = 1]| \leq \text{negl}$$

And we have a concrete example: CBC with rand. IV is not CCA-secure (insecure under active attacks).



b
Chal.
k←K
m_0 , m_1 : $|m_0| = |m_1| = 1$
c ← E(k, $\mathbf{m_b}$) = (IV, c[0])
c' = (IV⊕1, c[0])
D(k, **c'**) = m_b⊕1
Adv.
learn
b
b
Dan Boneh

Suppose Alice wants to send Bob `Borrow me $10000` (16 bytes), Eve wants to modify the 12th bit from `0x31` to `0x32`.

To do this, Eve can intercept encryption message and let `IV[11] = IV[11] ^ 0x31 ^ 0x32`, then Bob will get `Borrow me $20000` after decryption using the modified IV.

Authenticated Encryption (AE)

To solve this problem, authenticated encryption was invented to provide confidentiality against active attacks (i.e. provide CCA security).

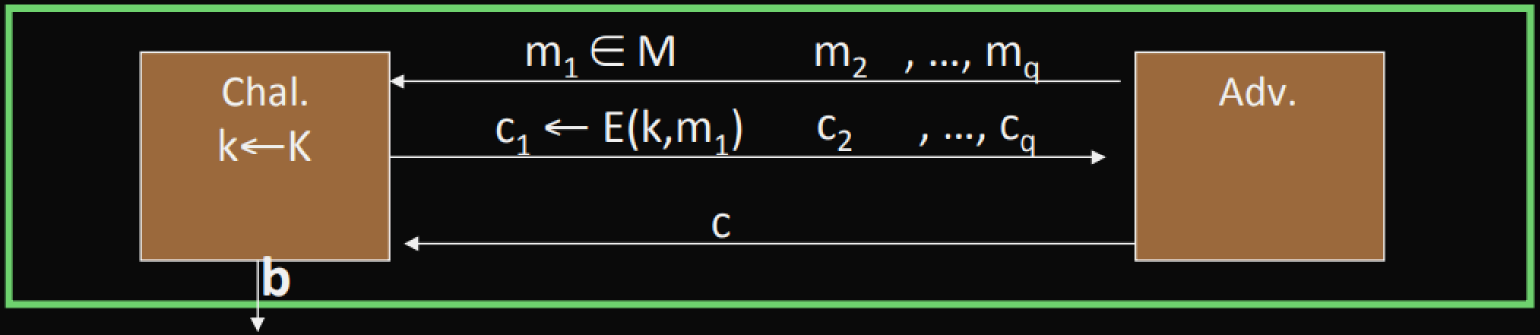
An authenticated encryption (AE) system is a cipher where:

- $E : K \times M \times N \to C$
- $D : K \times C \times N \to M \cup \{\perp\}$

DEF: cipher (E, D) provides AE if it is:

- sem. security under a CPA attack.
- ciphertext integrity: attack cannot create new ciphertexts that decrypt properly.

Let (E,D) be a cipher with message space M.


Chal.
$k \leftarrow K$
$m_1 \in M$ m_2 , ..., m_q
$c_1 \leftarrow E(k,m_1)$ c_2 , ..., c_q
Adv.
c
b

$$\begin{cases} \mathbf{b}=1 & \text{if } D(k,c) \neq \perp \quad \text{and } c \notin \{ c_1 , \dots , c_q \} \\ \mathbf{b}=0 & \text{otherwise} \end{cases}$$

Def: (E,D) has **ciphertext integrity** if for all "efficient" A:

$\text{Adv}_{\text{CI}}[A,E] = \Pr[\text{Chal. outputs } 1]$ is "negligible."

Construction of AE

Since we didn’t learn AE, the implementation is a little tricky for newcomers. Here is a question for you:

QUESTION: We often combine MAC and ENC together to achieve AE, so which of the implementations provide AE? Suppose encryption key is k_E and MAC key is k_I.

1. (SSL): let $tag = S(k_I, m), res = E(k_E, m||tag)$.
2. (IPSec): let $c = E(k_E, m), tag = S(k_I, c), res = c||tag$.
3. (SSH): let $c = E(k_E, m), tag = S(k_I, m), res = c||tag$.

The SSH way is obviously wrong, because let $tag = S(k_I, m)$ already exposes some information about m.

The SSL way is sometimes correct if (E, D) provides randomized encryption like rand-CTR or rand-CBC. Otherwise it may be insecure (image (E, D) using OTP).

Therefore, the answer is IPSec (encrypt then MAC), it is always correct.

QUESTION: Let (E, D) be an encryption system with key space K, message space $\{0,1\}^n$ and ciphertext space $\{0,1\}^s$. Suppose (E, D) provides authenticated encryption. Which of the following systems provide authenticated encryption: (as usual, we use $\|$ to denote string concatenation)

1. $$E'(k, m) = (E(k, m), H(m)) \text{ and } D'(k, (c, h)) = \begin{cases} D(k, c) & \text{if } H(D(k, c)) = h \\ \perp & \text{otherwise} \end{cases}$$

2. $$E'(k, m) = [c \leftarrow E(k, m), \text{output } (c, c)] \text{ and } D'(k, (c_1, c_2)) = \begin{cases} D(k, c_1) & \text{if } c_1 = c_2 \\ \perp & \text{otherwise} \end{cases}$$

3. $$E'(k, m) = (E(k, m), E(k, m)) \text{ and } D'(k, (c_1, c_2)) = D(k, c_1)$$

4. $$E'((k_1, k_2), m) = E(k_2, E(k_1, m)) \text{ and } D'((k_1, k_2), c) = \begin{cases} D(k_1, D(k_2, c)) & \text{if } D(k_2, c) \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

Deterministic Encryption (DE)

Suppose we have a database containing sensitive user information, such as government ID. We want to encrypt the IDs to protect user privacy, but we still need to be able to efficiently search for specific IDs in the database. That’s when deterministic encryption takes place.

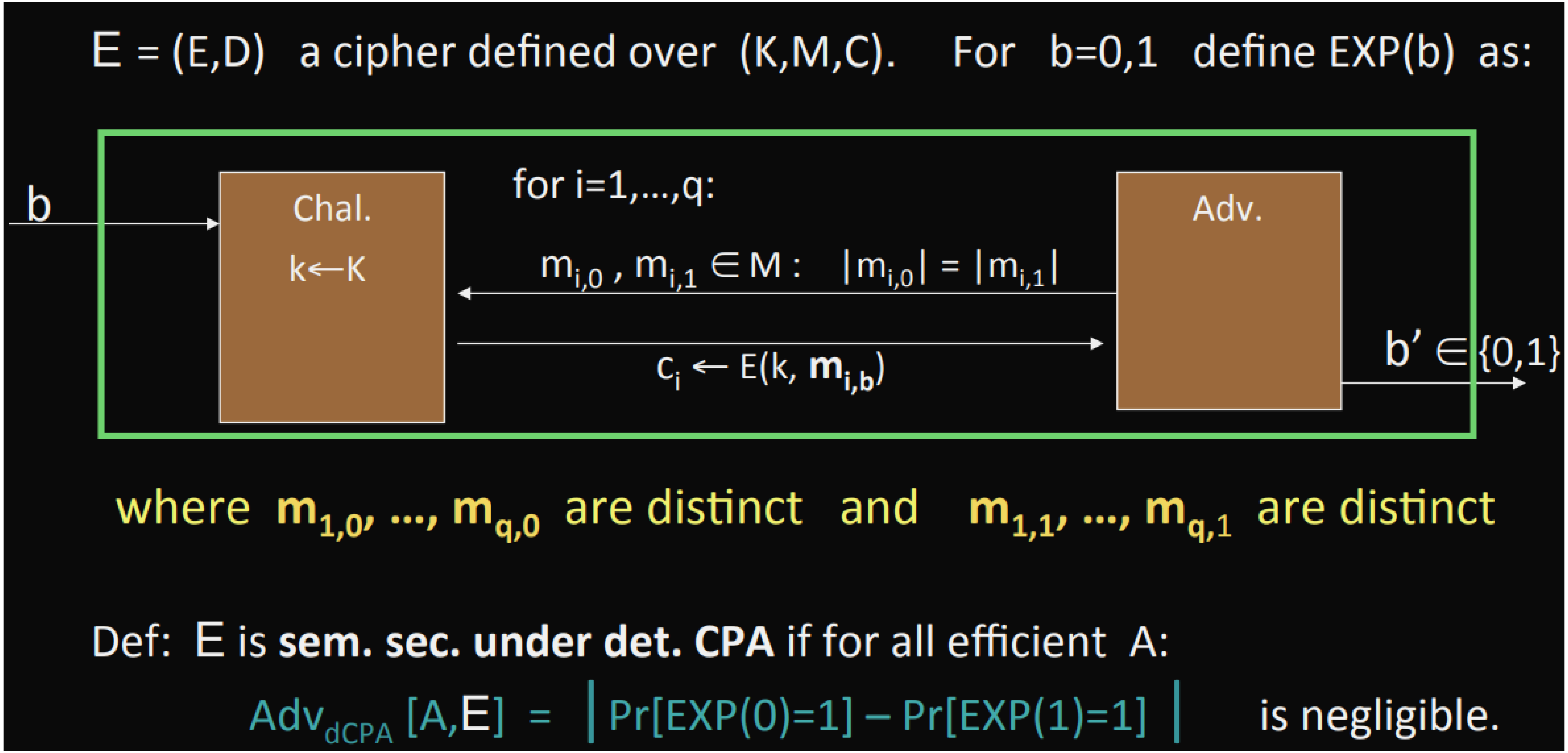
Of course, deterministic encryption cannot be CPA secure, the adversary can perform this attack:

Attacker can first choose m_0, m_0 and the server returns c_0.

After that, attacker choose m_0, m_1 and server will return c_0 or c_1. Since attacker knows the content of c_0, it can win CPA game.

The solution to this problem is **never encrypts same message twice**. This happens when encryptor:

- Chooses messages at random from a large message space $\mathcal{M}$.
- Message structure ensures uniqueness.



$\mathbb{E} = (E,D)$ a cipher defined over (K,M,C). For $b=0,1$ define EXP(b) as:

b
Chal.
$k \leftarrow K$
for i=1,…,q:
$m_{i,0}\,,\ m_{i,1} \in M: \quad |m_{i,0}| = |m_{i,1}|$
$c_i \leftarrow E(k,\ \mathbf{m_{i,b}})$
Adv.
$b' \in \{0,1\}$

where $\mathbf{m_{1,0}, \ldots, m_{q,0}}$ are distinct and $\mathbf{m_{1,1}, \ldots, m_{q,1}}$ are distinct

Def: $\mathbb{E}$ is **sem. sec. under det. CPA** if for all efficient A:

$$\mathrm{Adv}_{\mathrm{dCPA}}[A,\mathbb{E}] = \left|\Pr[\mathrm{EXP}(0)=1] - \Pr[\mathrm{EXP}(1)=1]\right| \quad \text{is negligible.}$$

QUESTION: Is CBC or CTR mode with fixed IV det. CPA secure?

QUESTION: Is PRP det. CPA secure?

Synthetic IV (SIV)

Although PRP is capable of building a det. encryption, PRP is not flexible because you need to build an n-bit PRP for n-bit messages. This is not convenient if you have a message with legnth 2^{20} bits.

Synthetic IV (SIV) can solve this problem, to be specific:

Let (E, D) be a CPA secure encryption: $E(k, m; r) \to c$

Let $F : K \times M \to R$ be a secure PRF.

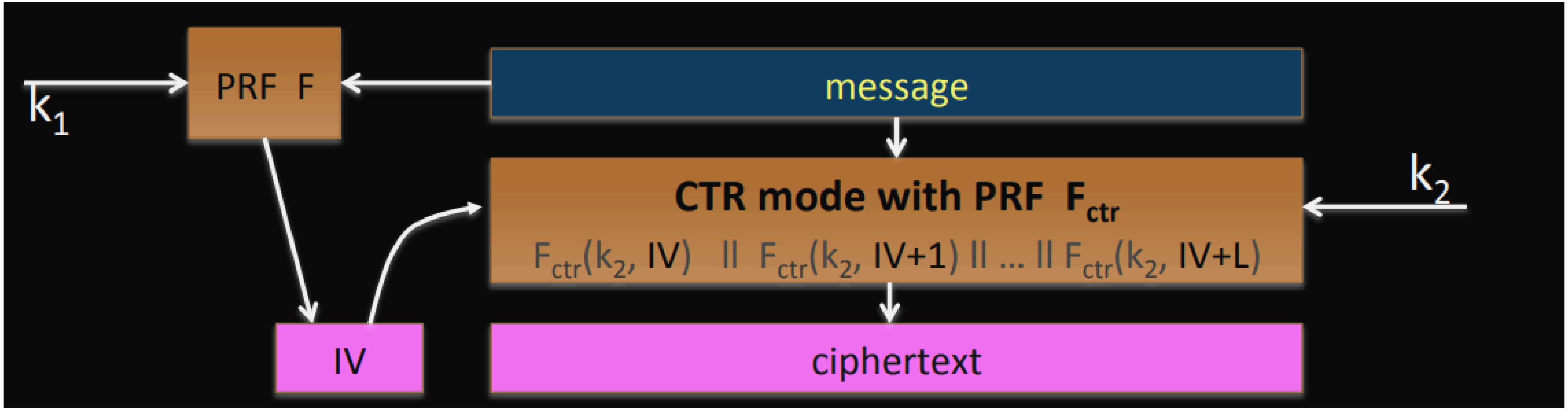
DEF:

$$E_{det}((k_1, k_2), m) = \begin{cases} r \leftarrow F(k_1, m) \\ c \leftarrow E(k_2, m; r) \\ \text{output r} \end{cases}$$

Then E_{det} is sem. sec. under det. CPA.

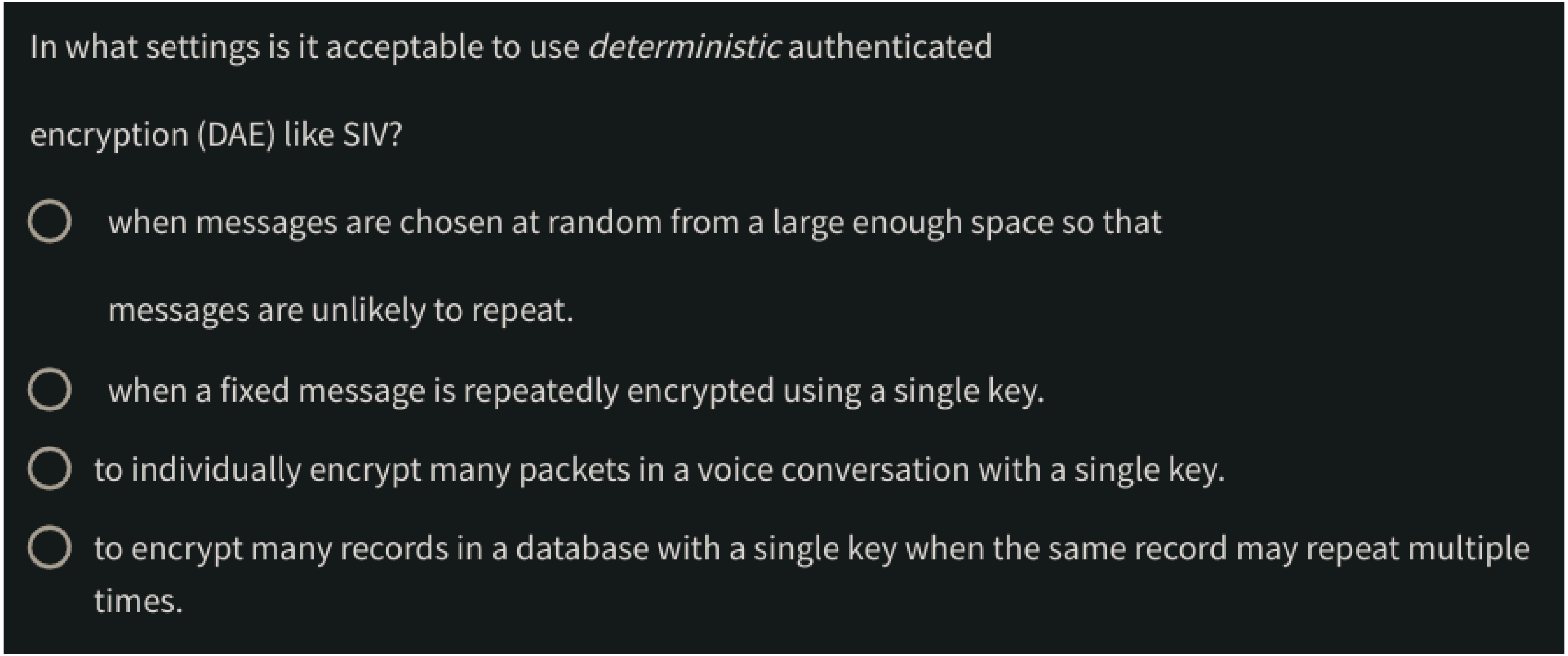
Deterministic Authenticated Encryption (DAE)

To modify SIV a little bit, we can get deterministic authenticated encryption (DAE) using CTR mode.


k1
PRF F
message
CTR mode with PRF F_{ctr}
k2
$F_{ctr}(k_2, IV)$ ‖ $F_{ctr}(k_2, IV+1)$ ‖ … ‖ $F_{ctr}(k_2, IV+L)$
IV
ciphertext

How about using PRP to do DAE? The easiet method is to append λ bits of zeros, and check if the decryption has λ bits of zeros at the end. Since PRP is pseudorandom, the probability of faking an encryption is $\frac{1}{2^\lambda}$, which is negligible.

There are also ways to expand PRP from $\{0,1\}^n$ to $\{0,1\}^N$ where $N \gg n$ like EME. However it is 2x slower than SIV so I'm not going to talk about it in this lecture.



In what settings is it acceptable to use *deterministic* authenticated encryption (DAE) like SIV?

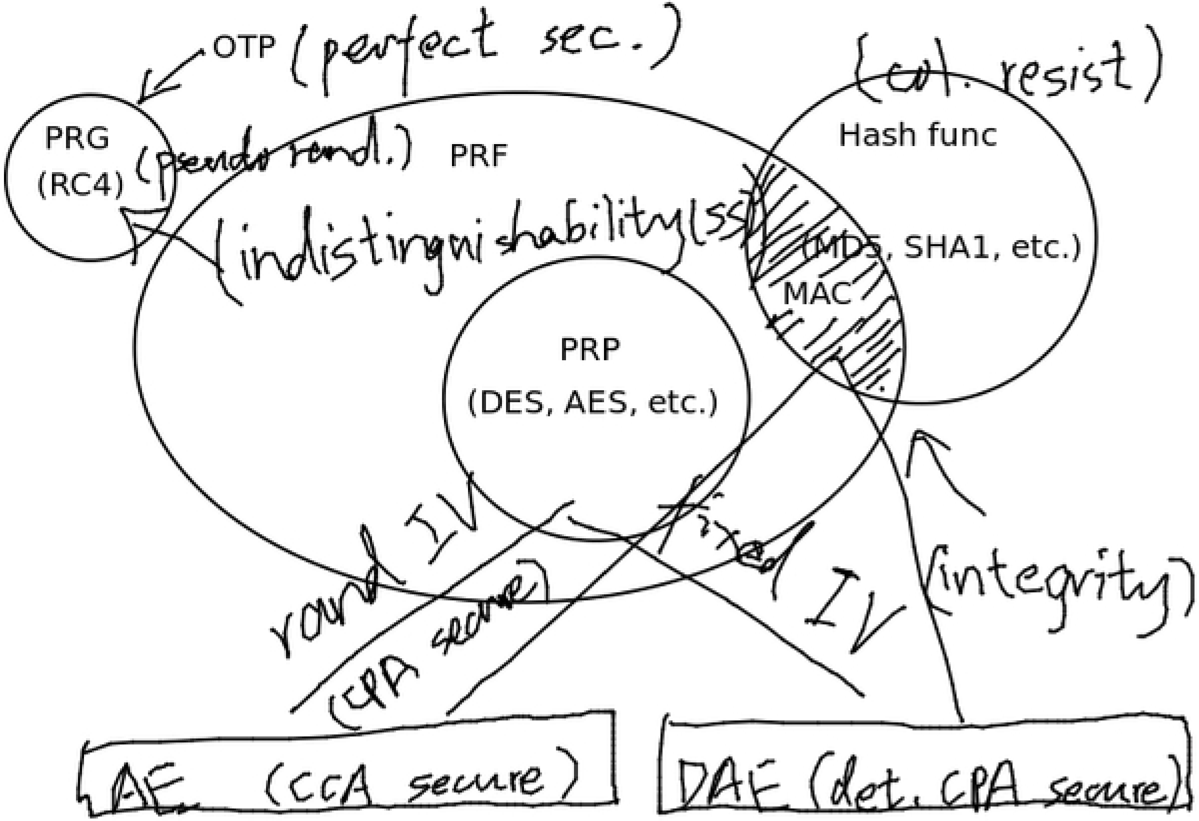
○ when messages are chosen at random from a large enough space so that messages are unlikely to repeat.

○ when a fixed message is repeatedly encrypted using a single key.

○ to individually encrypt many packets in a voice conversation with a single key.

○ to encrypt many records in a database with a single key when the same record may repeat multiple times.

Symmetric Encryption Summary



Public Key Encryption Intro

Suppose N people wants to share information with each other without eavesdropping.

The basic solution is to use a separate symmetric key for each pair, $O(N^2)$ keys in total.

However this is of course inefficient.

A better solution is to use a trusted third party (TTP), if Alice and Bob wants to communicate with each other. They first store their symmetric key in TTP. In this way, they only need to save their own key, $O(N)$ keys in total.

Alice can encrypt her message $c = Enc(k_A, m)$, TTP uses k_A to decrypt c and uses Bob's key k_B to send $c' = Enc(k_B, m)$ to Bob again. Bob then use his own key to decrypt c' and get m.

QUESTION: So, can we do it without using TTP?

Merkle Puzzles

Suppose we have a symmetric cipher $E(k, m)$ with $k \in \{0,1\}^{128}$.

Alice:

- Prepare 2^{32} puzzles, for $i = 1, 2, \cdots, 32$, choose rand. $P_i \in \{0,1\}^{32}$ and $(x_i, k_i) \in \{0,1\}^{128}$.
- Set $puzzle_i = E(0^{96} || P_i, \text{ "Puzzle } \#x_i\text{" } || k_i)$.
- Send $\{puzzle_i\}_{1,2,\cdots,2^{32}}$ to Bob.

Bob:

- Choose a random puzzle $puzzle_j$ to solve it and obtain (x_j, k_j).
- Send x_j to Alice.

Complexity:

- Alice: $O(n)$ time and $O(n)$ space.
- Bob: $O(n)$ time.
- Eve: $O(n^2)$ time.

Diffie-Hellman Key Exchange (DHKE)

However, the efficiency of merkle puzzles is too low, so we have Diffie-Hellman key exchange.

- SETUP(): a large prime p (e.g. 600 digits), an integers g (usually primitive root).
- Alice: choose random a in $[1, p-1]$, send $A = g^a$ to Bob.
- Bob: choose random b in $[1, p-1]$, send $B = g^b$ to Alice.
- Alice calculate $k_{AB} = B^a$, Bob calculate $k_{AB} = A^b$, then they can communicate using k_{AB}.

Since $B^a = (g^b)^a = g^{ab} = (g^a)^b = A^b$, this algorithm is valid.

Since knowing g^a and g^b is known to be hard to calculate g^{ab} (Dlog problem), this algorithm provide computation security when p is large enough.

The known best algorithm to solve Dlog problem, GNFS, has a complexity $O(\exp(n^{1/3}))$, where n is the digit number of p.

However, this key exchange algorithm only provides eavesdropping security. It cannot prevent active attacks, such as MitM.

Public Key Encryption

A different approach for key exchange is public key encryption. Here is the definition:

DEF: a public-key encryption system is a triple of algs (G, E, D):

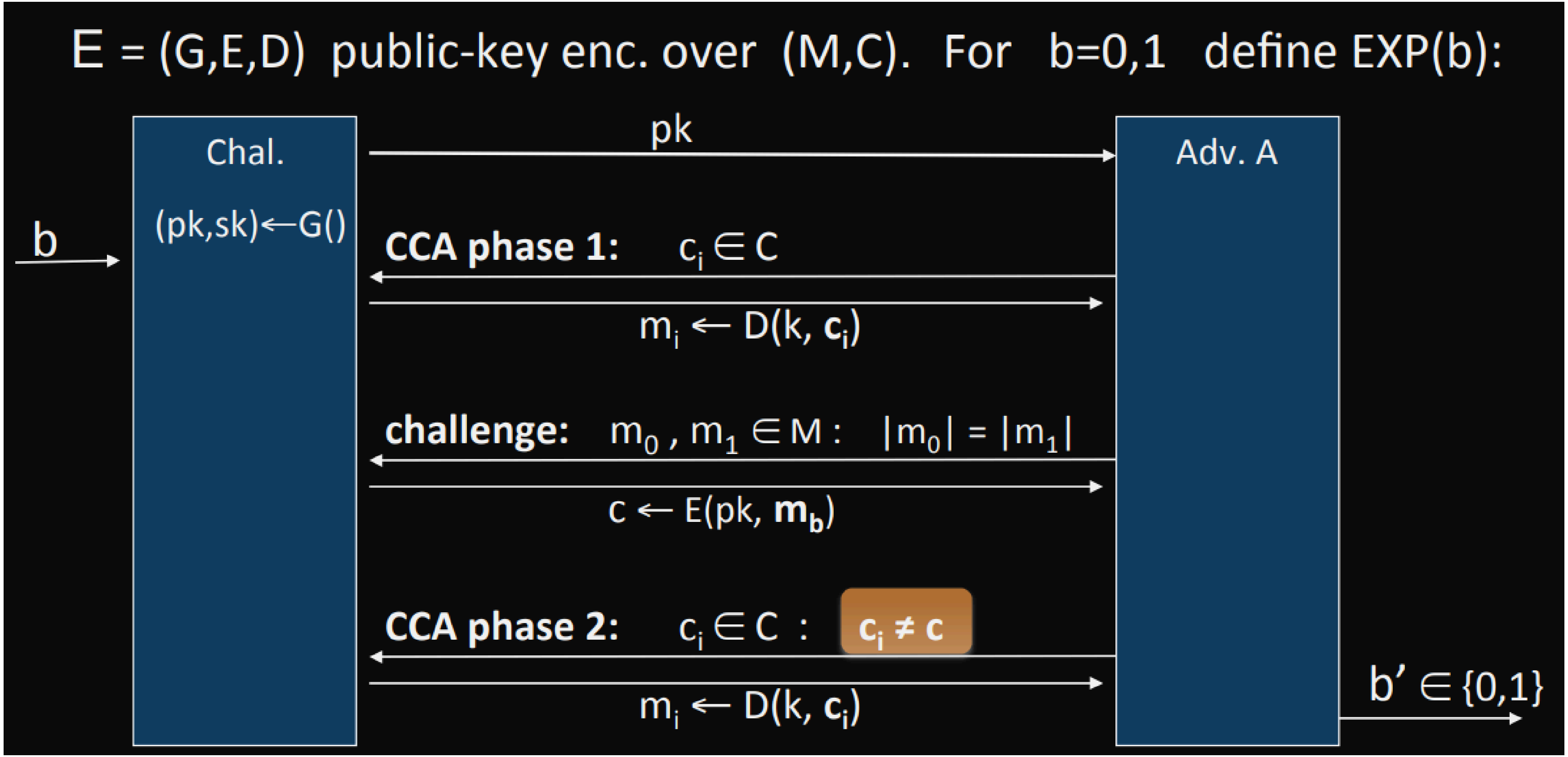
- $G()$: randomized alg. outputs a key pair (pk, sk).
- $E(pk, m)$: **randomized** alg. that takes $m \in \mathcal{M}$ and outputs $c \in \mathcal{C}$.
- $D(sk, c)$: det. alg. that $c \in \mathcal{C}$ and outputs $m \in \mathcal{M}$ or $\perp$.
- $\forall m \in \mathcal{M}, D(sk, E(pk, m)) = m.$

QUESTION: Is public key encryption able to solve MitM problem?

QUESTION: What level of security does public key encryption at least satisfiy?

- semantic security
- CPA security
- CCA security
- ind. CPA security

CCA Security for Public Key Encryption



E = (G,E,D) public-key enc. over (M,C). For b=0,1 define EXP(b):

CCA phase 1: $c_i \in C$

$m_i \leftarrow D(k, \mathbf{c_i})$

challenge: $m_0, m_1 \in M : \quad |m_0| = |m_1|$

$c \leftarrow E(pk, \mathbf{m_b})$

CCA phase 2: $c_i \in C$: $\mathbf{c_i \neq c}$

$m_i \leftarrow D(k, \mathbf{c_i})$

DEF: E is CCA secure if for all "eff" A: $Adv_{CCA}[A, E] = |Pr[\text{EXP}(0) = 1] - [Pr[\text{EXP}(1) = 1]| \leq$

Trapdoor Function (TDF)

Next we consider specific implementations of public key encryption, one of them is using trapdoor function:

DEF: a trapdoor function $X \to Y$ is a triple of "eff" algs (G, F, F^{-1}):

- $G()$: randomized alg. outputs a key pair (pk, sk).
- $F(pk, \cdot)$: **det.** alg. that defines a function $X \to Y$.
- $D(sk, \cdot)$: defines a function $Y \to X$ that inverts $F(pk, \cdot)$.
- $\forall x \in X, F^{-1}(sk, F(pk, x)) = x$.

QUESTION:

- Can you find the major difference from public key encryption?
- Can we use TDF directly to build public key encryption?

RSA Trapdoor Permutation

- $G()$:
  - choose random primes p, q 1024 bits. Set $N = pq$.
  - choose integers e, d s.t $ed = 1 \pmod{\varphi(N)}$.
  - output $pk = (N, E), sk = (N, d)$.
- $F(pk, x)$: $y = x^e$
- $D(sk, y)$: $y^d = x^{ed} = x^{ed \mod \varphi(N)} = x^1 = x$.

The security statement is: for all efficient algs. A:

$$Pr[A(N, e, y) = y^{1/e}] < \mathbf{negl}(N)$$

which is supported by the difficulty of factoring big integers.

By using RSA as a TDF and adding some randomness, we can construct RSA public key encryption easily.

RSA Public Key Encryption

(E_s, D_s) symmetric enc. scheme provide AE, $H : Z_N \to K$ where K is keyspace of (E, D).

- $G()$:
  - choose random primes p, q 1024 bits. Set $N = pq$.
  - choose integers e, d s.t $ed = 1 \pmod{\varphi(N)}$.
  - output $pk = (N, E), sk = (N, d)$.
- $F(pk, x)$:
  - choose random x in Z_N
  - $y = x^e$, $k \leftarrow H(x)$.
  - output $y, E_s(k, m)$.
- $D(sk, (y, c))$:
  - output $D_s(H(RSA^{-1}(y)), c)$.

Wiener’s Attack

However, as a TDF, RSA itself isn’t secure enough, Wiener proves that if $d \leq N^{0.25}/3$ then RSA is insecure.

Proof:

$$ed = 1 \pmod{\varphi(N)} \Rightarrow \exists k \in \mathbb{Z}, ed = k \cdot \varphi(N) + 1$$

$$|\frac{e}{\varphi(N)} - \frac{k}{d}| = \frac{1}{d \cdot \varphi(N)} \leq \frac{1}{\sqrt{N}}$$

since $\varphi(N) = N - p - q + 1$, $|N - \varphi(N)| \leq p + q \leq 3\sqrt{N}$

$$d \leq N^{0.25}/3 \Rightarrow |\frac{e}{N} - \frac{k}{d}| \leq |\frac{e}{N} - \frac{e}{\varphi(N)}| + |\frac{e}{\varphi(N)} - \frac{k}{d}| = |\frac{e(\varphi(N) - N)}{N \cdot \varphi(N)}| + |\frac{e}{\varphi(N)} - \frac{k}{d}|$$

$$<= \frac{3\sqrt{N}}{N} + \frac{\sqrt{N}}{N} = \frac{4}{\sqrt{N}} < \frac{1}{2d^2}$$

As you can see, $|\frac{e}{N} - \frac{k}{d}|$ is very small, so continued fraction of $\frac{e}{N}$ gives $\frac{k}{d}$. Q.E.D.

Suppose $N = 90581, e = 17993$, we can try to write the continued fraction of $\frac{e}{N}$.

- $17993 = 0 \times 90581 + 17993$, then $q_0 = 0$.
- $90581 = 5 \times 17993 + 616$, then $q_1 = 5$.
- $17993 = 29 \times 616 + 128$, then $q_2 = 29$.
- $616 = 4 \times 128 + 104$, then $q_3 = 4$.

Then we have e/N = $[0, 5, 29, 4, 1, 3, 2, 4, 3]$.

We can have convergents like:

0, $\frac{1}{5}$, $\frac{1}{5+\frac{1}{29}} = \frac{29}{146}, \frac{1}{5+\frac{1}{29+\frac{1}{4}}} = \frac{117}{589}, \frac{146}{735}, \frac{555}{2794}, \frac{1256}{6323}, \frac{5579}{28086}, \frac{17993}{90581}$.

And we can try $\varphi(N) = \frac{ed-1}{k}$: $k = 1, d = 5$ satisfies, we got $\varphi(N) = 89964$, other pairs can’t be an integer.

Solve $x^2 - ((N - \varphi(N)) + 1)x + N = 0$, we got $(239, 379)$, and we’ve found factorization of N, which breaks the whole RSA system.

ElGamal Public Key system

Since RSA has some defects, ElGamal, which is based on Diffie-Hellman Key Exchange, came out of place.

First, let’s recap DHKE:

> - SETUP(): a large prime p (e.g. 600 digits), an integers g (usually primitive root).
> - Alice: choose random a in $[1, p-1]$, send $A = g^a$ to Bob.
> - Bob: choose random b in $[1, p-1]$, send $B = g^b$ to Alice.
> - Alice calculate $k_{AB} = B^a$, Bob calculate $k_{AB} = A^b$, then they can communicate using k_{AB}.

QUESTION: what will happen if we treat A as pk?

Therefore, a basic ElGamal protocol is invented.

- SETUP(): a large prime p (e.g. 600 digits), an integers g (usually primitive root).
- Alice: choose random a in $[1, p-1]$, send $A = g^a$ to Bob.
- Bob: choose random b in $[1, p-1]$, encrypt m using $k_{AB} = A^b$, send $(B = g^b, E_s(k_{AB}, m))$ to Alice.
- Alice calculate $k_{AB} = B^a$, decrypt $m = D_s(k_{AB}, c)$.

And here is a more modern version:

- G : finite cyclic group of order n, random generator $g \in G$.
- $\mathrm{sk} = a \overset{R}{\leftarrow} Z_n$, $\mathrm{pk} = (g, h = g^a)$.
- (E_s, D_s) : symmetric AE defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$.
- $H : G^2 \to K$ a hash function.

$E(\mathrm{pk} = (g, h), m)$:

- $b \overset{R}{\leftarrow} Z_n, u \leftarrow g^b, v \leftarrow h^b$.
- $k \leftarrow H(u, v), c \leftarrow E_s(k, m)$.
- $\text{output } (u, c)$.

$D(\mathrm{sk} = a, (u, c))$:

- $v = u^a$.
- $k = H(u, v), m \leftarrow D_s(k, c)$.
- $\text{output } m$.

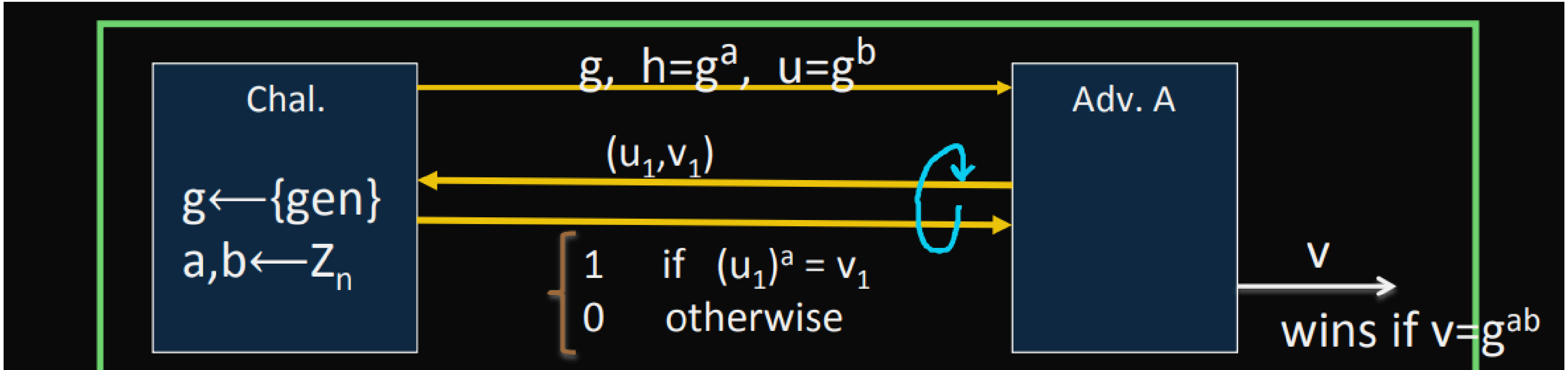
ElGamal CCA Security

In previous sections, we know basic ElGamal protocol relies on Dlog problem, just as same as Diffie-Hellman key exchange. However, it is vulnerable to MitM attacks. So the question is:

QUESTION: Does modern version of ElGamal provide CCA security?

To solve this question, let’s look at several security assumptions:

- Computational DH (CDH): for all "eff" algs. A, $Pr[A(g, g^a, g^b) = g^{ab}] < \text{negl.}$
- Hash DH (HDH): $H : G^2 \to K$ as a hash function, $(g, g^a, g^b, H(g^b, g^{ab})) \approx_p (g, g^a, g^b, R)$.
- Interactive DH (IDH): for all "eff" A, $Pr[A \text{ outputs } g^{ab}] < \text{negl.}$



In fact, if:

- IDH holds in the group G;
- (E_s, D_s) provides AE;
- $H : G^2 \to K$ is a "random oracle".

then ElGamal is CCA^{ro} secure.

QUESTION:

- Can we prove CCA security based on CDH?
- Can we prove CCA security without random oracles?

ElGamal Variants

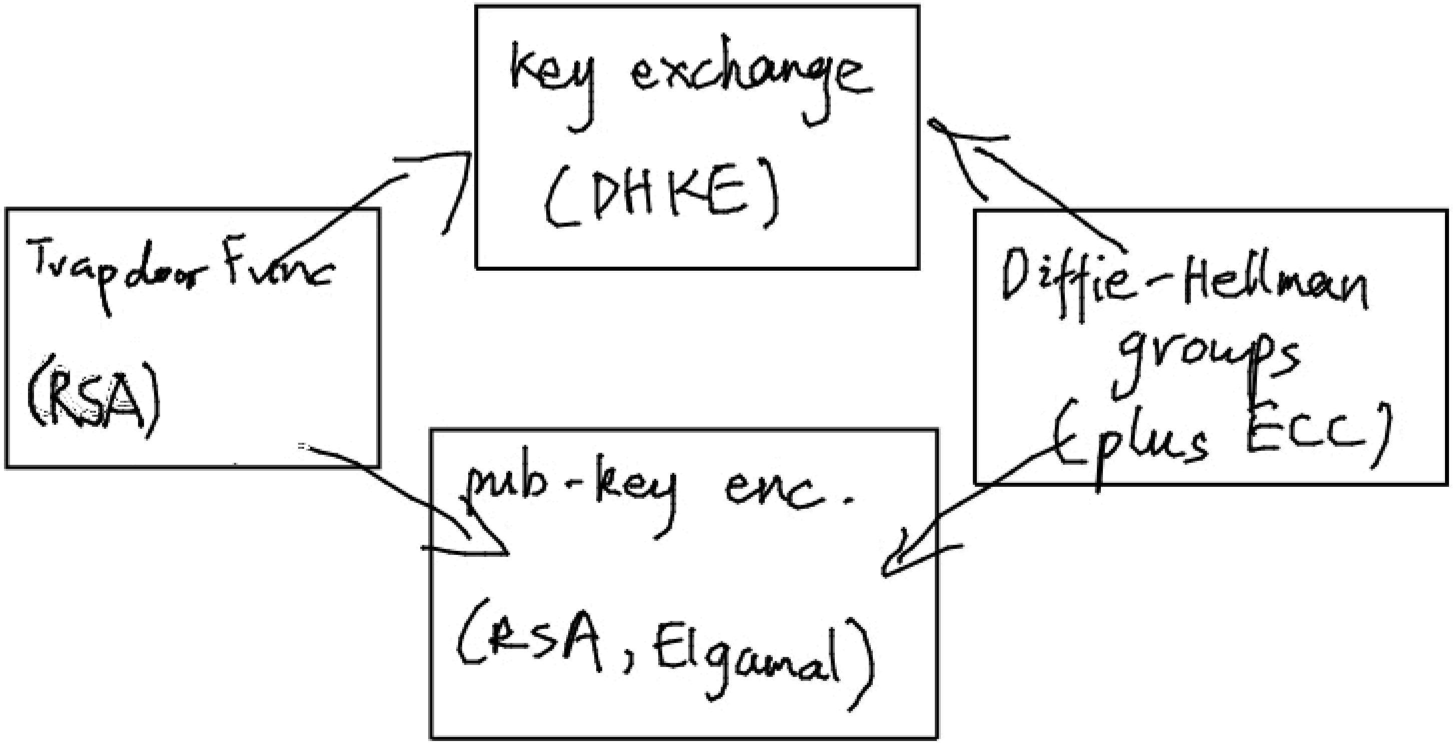
Twin ElGamal:

- SETUP(): $g \in G \text{ and } a_1, a_2 \in Z_n$, output $\text{pk} = (g, h_1 = g^{a_1}, h_2 = g^{a_2})$, $\text{sk} = (a_1, a_2)$.
- $E(\text{pk} = (g, h_1, h_2), m)) : b \leftarrow Z_n$
  - $k \leftarrow H(g^b, h_1^b, h_2^b)$.
  - $c \leftarrow E_s(k, m)$.
  - $\text{output } (g^b, c)$.
- $D(\text{sk} = (a_1, a_2), (u, c)) :$
  - $k \leftarrow H(u, u^{a_1}, u^{a_2})$.
  - $m \leftarrow D_s(k, c)$.
  - $\text{output } m$.

You can observe this algorithm "has a beauty of symmetry".

Another popular variant is ElGamal on ECC (ECELG), you will learn/have already learnt in your textbook.

Public Key Encryption Summary



THANKS FOR YOUR LISTENING!