Cryptography, Master MICAS, Part I

Michèle Wigger

Telecom Paris, October 23, 2020

Outline of the Course: Part I

Michèle Wigger (3C58) and Mehrasa Ahmadipour (3C61)

- Use Cases of Cryptography;
- Historical Cyphers, Stream Ciphers and Block Ciphers;
- Perfect and Semantic Security against Chosen Plaintext Attacks; Active Attacks: Tempering and Chosen Ciphertext Attacks;
- Message Authentication Codes (MAC) and Hash Functions;
- Authenticated Encryption;
- Pseudo Random Number Generators, Pseudo Random Functions, Pseudo Random Permutations, The Random Oracle Model;
- Basics in Number Theory and Algebra;

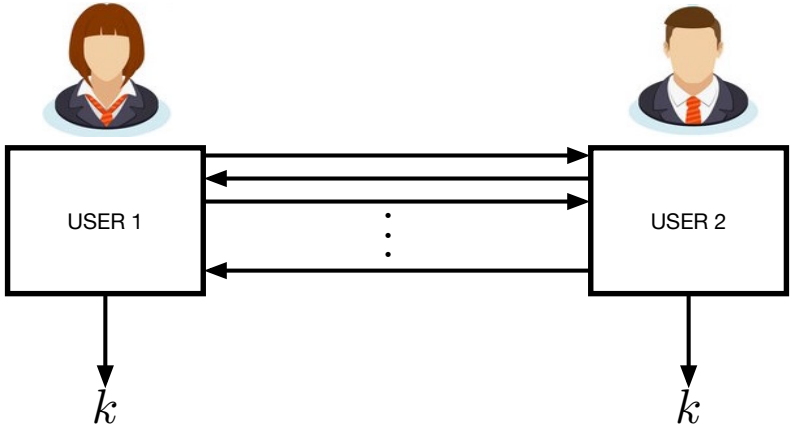
Outline of the Course: Part II

Hieu Phan

- Public Key Encryption Systems
- Digital Signatures
- Zero-Knowledge Proofs
- Decentralized Cryptosystems

Use Cases of Cryptography

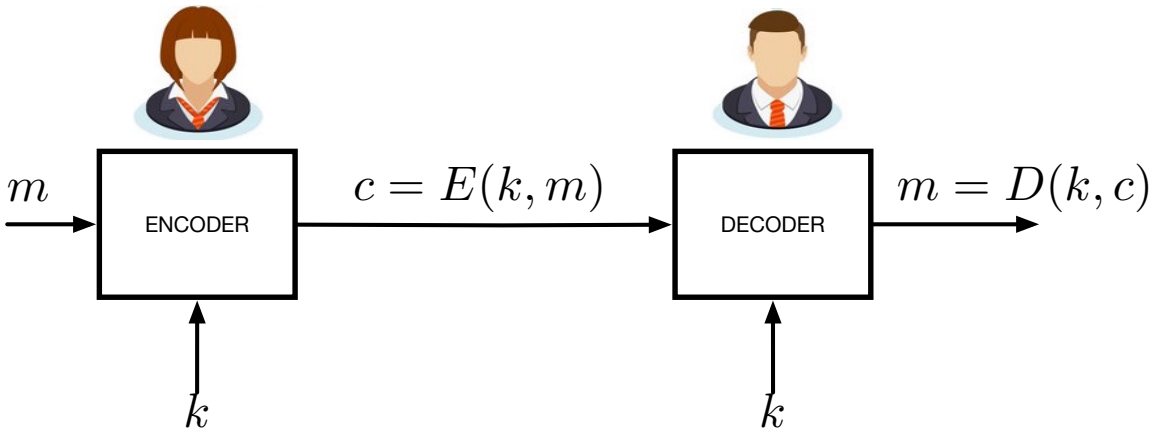
Establishing a Common Secret Key



- k: secret key

The two parties wish to create a common key k that looks completely random to other users.

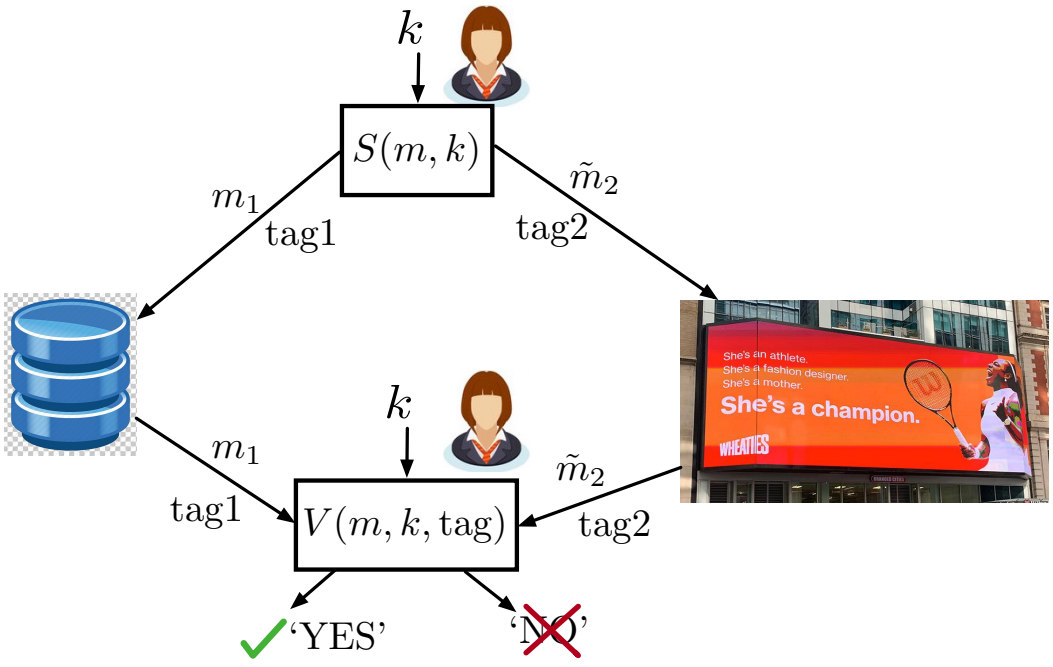
Encryption



- m: plaintext (message)
- k: secret key
- c: cipher text
- (**E**,**D**) is called a cipher

Ciphertext c should leak "no" information about plaintext m even after observing c or different (m_i, c_i) pairs.

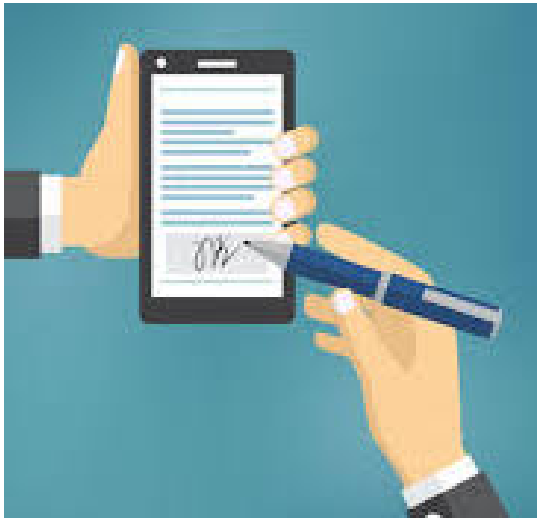
Authentication



- t: tag
- (**S**,**V**) is called an authentication code

For an adversary it should be hard to forge a message-tag pair (m, tag) without knowing the secret key k.

Digital Signatures



- sk: secret key
- pk: public key
- $\sigma(sk, m)$: signature on a message m
- $\mathbf{v}(pk, m, \sigma) \in \{\text{valid, false}\}$: verification of an algorithm
- $(\sigma,\mathbf{v})$ is called a digital signature algorithm

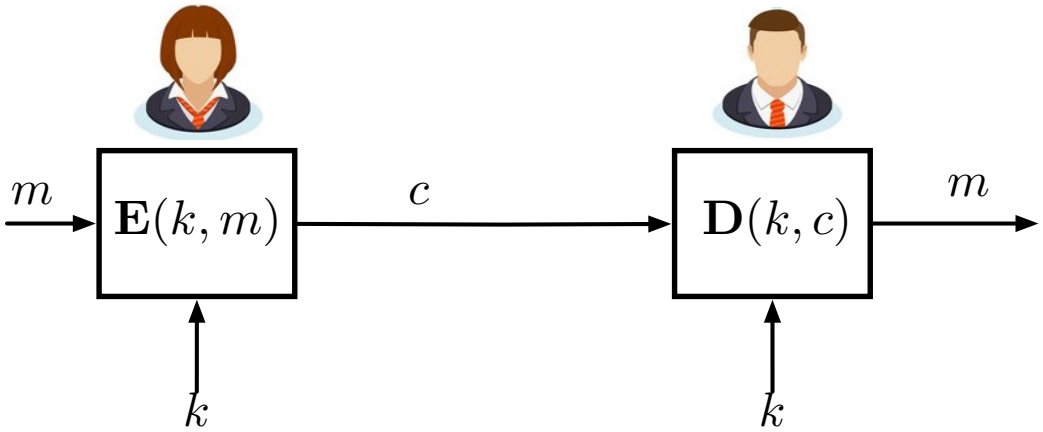
For an adversary it should be hard to forge a *valid* signature $\sigma(m, sk)$ on a document m without knowing the secret key sk.

Multi-Party Protocols

- For example blockchains
- Perform distributed agreements/computations based on distributed data and in the presence of malicious users

Historical Ciphers

Symmetric Ciphers



- m: plaintext (message)
- k: secret key known at the encoder and the decoder
- c: cipher text
- (**E**,**D**) is called a cipher

Substitution Cipher

- Key k is a substitution table

Plaintext	···	d	e	···	h	···	l	m	n	o	···	r	w	···
Ciphertext	···	u	t	···	q	···	j	k	m	l	···	a	b	···

- Encryption: Apply the substitution character by character

  m = hello world $\Longrightarrow$ c = qtkkl blaju

- Decryption: Reverse the substitution character by character

Caesar Cipher

- Key k determines the shift of the substitution:

E.g., $k = 3$:

Plaintext	$\cdots$	d	e	$\cdots$	h	$\cdots$	l	m	n	o	$\cdots$	r	w	$\cdots$
Ciphertext	$\cdots$	g	h	$\cdots$	k	$\cdots$	o	p	q	r	$\cdots$	u	z	$\cdots$

- Encryption for $K = 3$:

  $m =$ hello world $\quad\Longrightarrow\quad$ $c =$ khoor zruog

- Decryption: Reverse the shift character by character

How to Break the Substitution or Caesar Cipher

- Frequency of letters in a given language. E.g., *e* is the most frequent letter, than *t* and *a*
- Use frequency of pairs

We call this a cypher-text (CT) only attack.

Vigenère Cipher (16th–19th century)

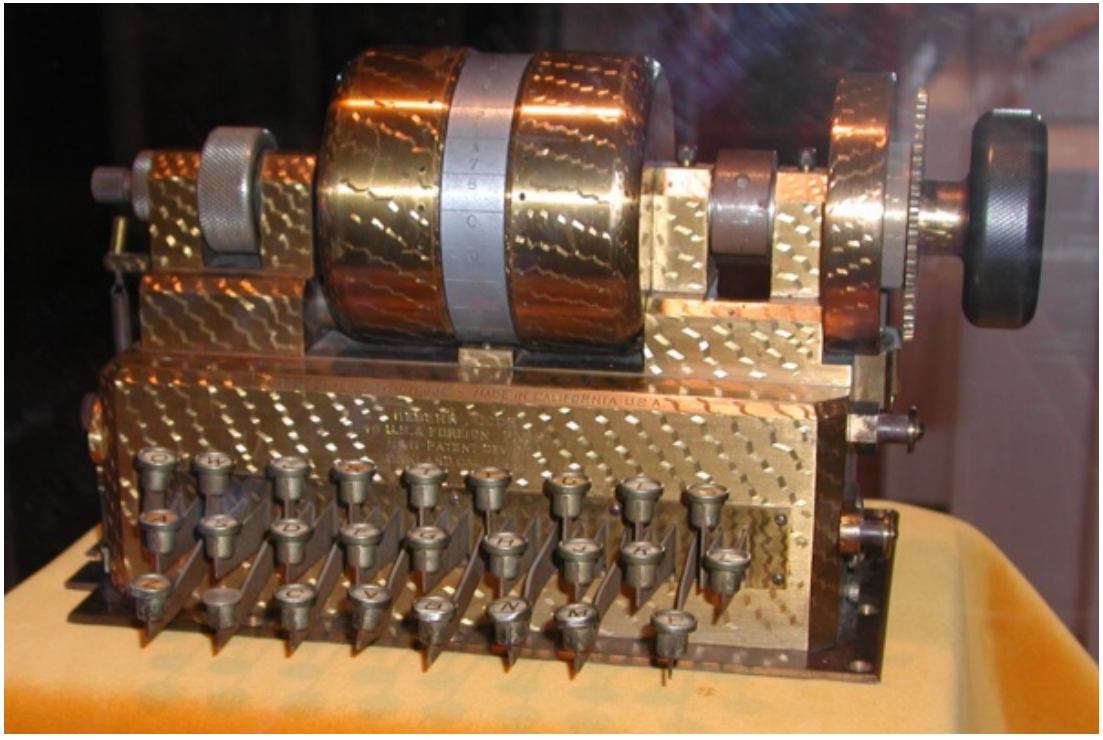
- Key: $k =$ "bonbon"
- Encryption: Repeat the key, and then sum ($a = 0$) this repeated key with the plaintext modulo 26

Keytext	b	o	n	b	o	n	b	o	n	b	o	n	b	o	n
Plaintext	h	e	l	l	o	w	o	r	l	d	c	h	u	c	k
Ciphertext	i	s	y	m	c	j	p	f	y	e	q	u	v	q	x
	↑						↑						↑		

- Assuming one knows the key length, one can do the same attack as above, simply by considering periodic sequences of letters

Rotor Machines: The Hebern Machine

- Rotor Machines from 1870–1943
- The key determines the initial substitution table, afterwards the machine shifts the substitution table by one symbol after each letter.



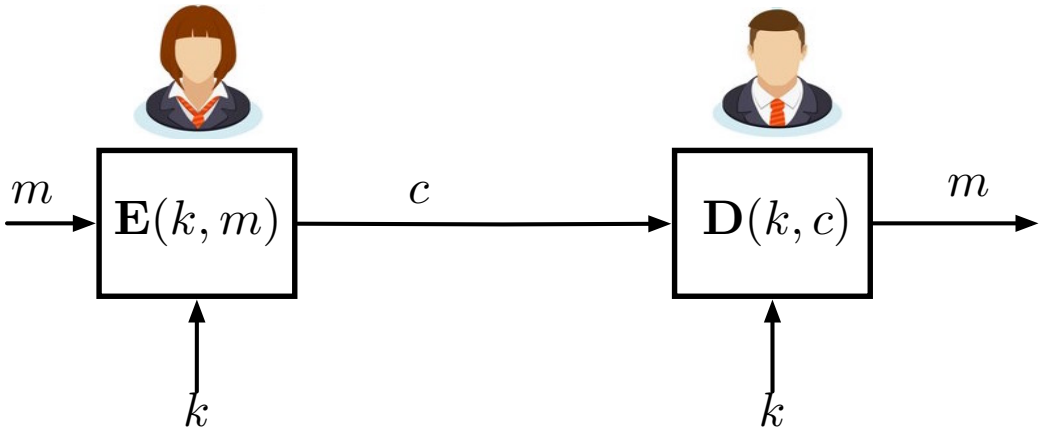
- Easy to break

Rotor Machines: The Enigma Machine

- First part of the last century
- Multiple rotors changing with different frequencies
- Handy and portable $\rightarrow$ used in wars
- Books used to determine the initial keys and the wirings
- Encrypted ciphertext/decrypted plaintext shown with lamps

Stream Ciphers, Pseudo Random Number Generators, Semantic Security

Symmetric Ciphers



Definition

A symmtric cipher of a key alphabet $\mathcal{K}$, a message space $\mathcal{M}$, and a cipher space $\mathcal{C}$ is a pair of "efficient" algorithms

$$\begin{aligned} \mathbf{E}: &\quad \mathcal{K} \times \mathcal{M} \to \mathcal{C} \\ \mathbf{D}: &\quad \mathcal{K} \times \mathcal{C} \to \mathcal{M} \end{aligned}$$

such that for all $k \in \mathcal{K}, m \in \mathcal{M}$:

$$\mathbf{D}(k, \mathbf{E}(k, m)) = m$$

Many modern ciphers have key size $|\mathcal{K}| = 2^{128} \approx 10^{38,4}$

OneTime Pad (OTP)

- Key k is as long as the plaintext (used only once!)
- Ciphertext obtained by XORing the plaintext and the key

Plaintext	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0
Keytext	1	1	0	0	0	1	0	1	0	0	1	0	0	0	1
Ciphertext	0	1	1	1	0	0	1	1	0	0	0	1	0	1	1

- Decryption performs the XOR in the reverse direction
- Very fast encryption/decryption
- Requires a very long key

Information-Theoretic Security

Definition (Shannon 1949)

A cipher $(\mathbf{E}, \mathbf{D})$ is *perfectly secure* if for $K \overset{\mathcal{U}}{\sim} \mathcal{K}$ independent of $M \sim \mathcal{M}$:

$$I(M\,;\,\mathbf{E}(K,M))) = 0 \qquad \Leftrightarrow \qquad C = \mathbf{E}(K,M) \text{ independent of } M.$$

Here, $I(\cdot;\cdot)$ denotes mutual information.

- Even the most powerful adversary learns nothing from a cyphertext (CT) only attack!
- The OTP is the only information-theoretic secure cipher!

Information-Theoretic Security

Definition (Shannon 1949)

A cipher $(\mathbf{E}, \mathbf{D})$ is *perfectly secure* if for $K \overset{\mathcal{U}}{\sim} \mathcal{K}$ independent of $M \sim \mathcal{M}$:

$$I(M\,;\,\mathbf{E}(K,M))) = 0 \qquad \Leftrightarrow \qquad C = \mathbf{E}(K,M) \text{ independent of } M.$$

Here, $I(\cdot;\cdot)$ denotes mutual information.

- Even the most powerful adversary learns nothing from a cyphertext (CT) only attack!
- The OTP is the only information-theoretic secure cipher!
  $\rightarrow$ Problem: key length too large!

Stream Ciphers

- Idea: Replace the secret key by a *Pseudo Random Sequence*

PR Seq.	1	1	0	0	1	1	0	1	0	0	1	0	0	0	1
Plaintext	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0
Ciphertext	0	1	1	1	0	0	0	1	0	0	0	1	0	1	1

Stream Ciphers and Pseudo Random Number Generators (PRG)

- Idea: Replace the secret key by a *Pseudo Random Sequence*

PR Seq.	1	1	0	0	1	1	0	1	0	0	1	0	0	0	1
Plaintext	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0
Ciphertext	0	1	1	1	0	0	0	1	0	0	0	1	0	1	1

Definition

A *Pseudo Random Number Generator (PRG)* is a function

$$\mathbf{G}_\lambda : \mathcal{K}_\lambda \to \{0,1\}^{n(\lambda)}$$

where λ is a security parameter; $n(\lambda)$ is an increasing function in λ; and $\mathcal{K}_\lambda$ denotes the key space, which also grows with λ but typically $|\mathcal{K}_\lambda| \ll 2^{n(\lambda)}$.

Stream ciphers are not perfectly secure!

Unpredictable PRGs

Definition (Unpredictable PRGs)

A PRG $\mathbf{G}_\lambda : \mathcal{K} \to \{0,1\}^n$ is *unpredictable* if, given a uniform key $K \overset{\mathcal{U}}{\sim} \mathcal{K}$, for any index $i \in \{1, \ldots, n\}$ and any "efficient" adversary $\mathbf{A}_i : \{0,1\}^{i-1} \to \{0,1\}$, the adversary's *advantage* $\mathsf{Adv}_{\mathsf{Pred}}(\mathbf{A}_i, \mathbf{G}_\lambda)$ of *predicting* the i-th bit from the previous $i-1$ bits,

$$\mathsf{Adv}_{\mathsf{Pred}}(\mathbf{A}_i, \mathbf{G}_\lambda) := \Pr\left[\mathbf{A}_i(\mathbf{G}_\lambda(K)[1 : i-1]) = \mathbf{G}_\lambda(K)[i]\right]$$

satisfies

$$\mathsf{Adv}_{\mathsf{Pred}}(\mathbf{A}_i, \mathbf{G}_\lambda) \leq \frac{1}{2} + \epsilon(\lambda), \quad \text{for a negligible function } \epsilon(\lambda).$$

If $\mathbf{G}_\lambda$ produced an independent and identically distributed (i.i.d.) Bernoulli-$1/2$ sequence $\mathbf{B}_R$, the advantage of any adversary is exactly $1/2$.

Negligible Terms and Efficient Algorithms

Negligible $\epsilon(\lambda) > 0$:

- In theory: The function $\epsilon(\lambda)$ vanishes faster in the security parameter λ than any polynomial function:

$$\forall d > 0: \quad \epsilon(\lambda) \leq \lambda^{-d}, \qquad \text{whenever } \lambda > \lambda_{0,d}.$$

- In practice: $\epsilon \leq \frac{1}{2^{80}} \rightarrow$ is not more likely than correctly guessing the key if $|\mathcal{K}| \approx 2^{80}$

Efficient algorithms $\mathbf{A}(\cdot)$:

- In theory: except for negligible probability it runs in *polynomial time* $t(\lambda)$:

$$\exists c, d, \lambda_0 > 0: \quad t(\lambda) \leq c^{\lambda} + d, \quad \forall \lambda > \lambda_0$$

- In practice: an honest user can run an algorithm in a few minutes on a computer; an attacker can run an algorithm in less than a few years on 10000 parallel computers

Security Definition of PRGs

Definition (Secure PRGs)

A PRG $\mathbf{G}$ is *secure* if for any "efficient" adversary $\mathbf{A}\colon \{0,1\}^n \to \{0,1\}$, the adversary's advantage compared to observing a truly random Bernoulli-$1/2$ sequence $\mathbf{B}_R$,

$$\mathrm{Adv}_{\mathrm{PRG}}(\mathbf{A},\mathbf{G}) := \left|\Pr\left[\mathbf{A}(\mathbf{G}(K)) = 1\right] - \Pr\left[\mathbf{A}(\mathbf{B}_R) = 1\right]\right|,$$

is negligible:

$$\mathrm{Adv}_{\mathrm{PRG}}(\mathbf{A},\mathbf{G}) \leq \epsilon, \qquad \text{for a negligible } \epsilon.$$

- A secure PRG is *computationally indistinguishable* from a purely random number sequence.
- No provably secure PRGs are known. However, some PRGs are believed to be "secure", i.e., no better attack than exhaustive search is known.

An Unpredictable PRG is Secure and Vice Versa

- For a predictable PRG we can construct an adversary that has non-negligible advantage. Simply use the predictor $\mathbf{A}_i$ of the predictable bit i as the output of the adversary:

$$\mathsf{Adv}_{\mathsf{PRG}}(\mathbf{A}_i, \mathbf{G}) = \left| \underbrace{\Pr\left[\mathbf{A}(\mathbf{G}(K)) = 1\right]}_{\geq 1/2+\epsilon} - \underbrace{\Pr\left[\mathbf{A}(\mathbf{B}_R) = 1\right]}_{=1/2} \right| \geq |1/2+\epsilon-1/2| = \epsilon.$$

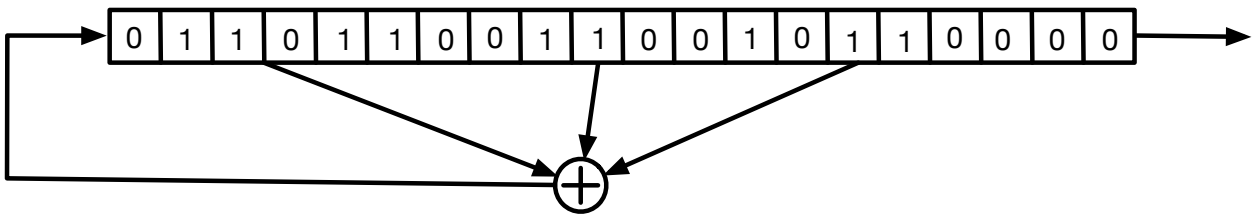
- It can also be shown that an unpredictable PRG is secure.

Some Real-World Stream Ciphers

- RC4: Used to secure web-traffic. Fast when implemented in software. But can be broken by exploiting some biases $\rightarrow$ second byte is more likely to be $\mathbf{0}$ and the probability of having $(\mathbf{0}, \mathbf{0})$ is also larger
- CSS: is also broken
- Salsa 20/12
- Sosemanuk

Linear Feedback Shift Registers (LFSR)

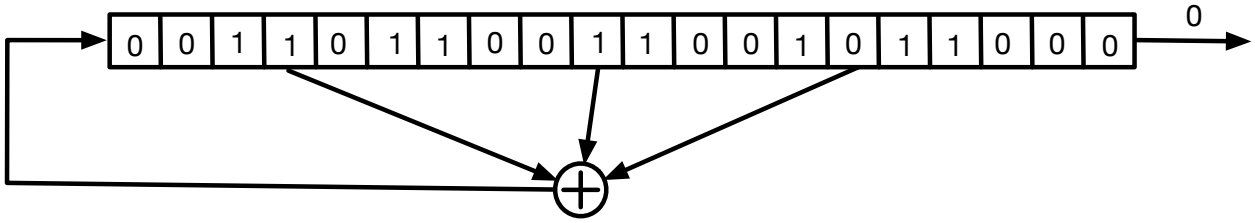
- The initial values are called *seed (key)* and can be padded with zeros



- Easy to implement in hardware

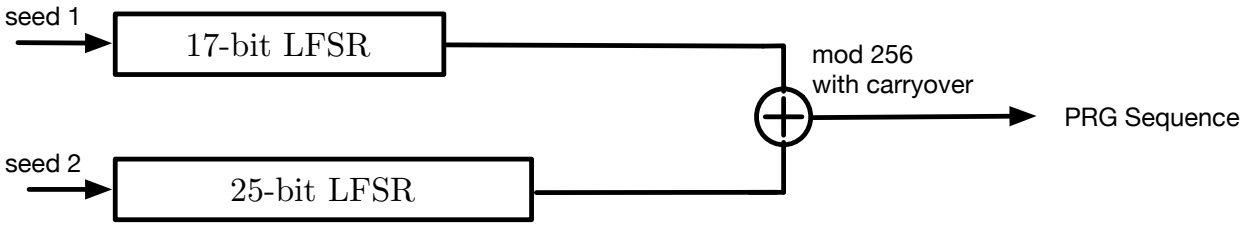
Linear Feedback Shift Registers (LFSR)

- The initial values are called *seed (key)* and can be padded with zeros



- Easy to implement in hardware

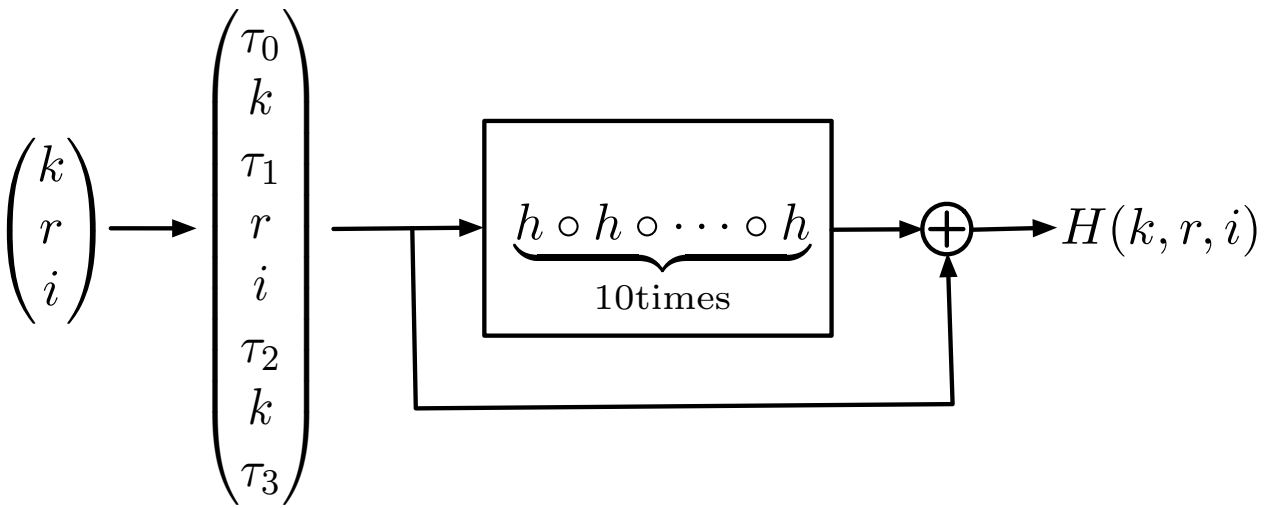
Content Scramble System (CSS)



- Encryption/Decryption: XOR the PRG sequence with the movie.
- Key size 5 bytes=40 bits: seed 1 has 16 bits and seed 2 has 24 bits
- But actual security is no more than 16 bits. Here is the attack:
  - The first 20 bytes of the movie are a known prefix.
  - XORing this prefix with the encrypted file, yields the first 20 bytes of the PRG sequence.
  - Run through all possibilities for seed 1, and XOR the LFSR-1 outputs with the first 20 bytes of the PRG sequence.
  - If the result is consistent with seed 2 $\rightarrow$ found seed 1 *and* seed2; Otherwise try with the next seed 1.

eStream: Salsa 20 (no better attack known than exhaustive search)

- Encryption: $(k, r) \mapsto s$ where k is denotes a 128 or 256 bits long seed, r a 64 bits once, and s a maximally 2^{73} bits long PRG output sequence
- Each (k, r) pair is used only once$\rightarrow$ the nonce allows to use a key for a longer time
- $s = H(k, r, 0)||H(k, r, 1)||H(k, r, 2)|| \cdots$ where H is defined as in the following figure:



- for h an invertible function and $\tau_0, \tau_1, \tau_2, \tau_3$ given numbers

Semantic Security for Ciphertext (CT) Only Attack

- *Ciphertext (CT) only Attack*: An adversary sees a ciphertext and wishes to determine the message, the key, or any other related information
  $\rightarrow$ only one ciphertext is observed and key only used once

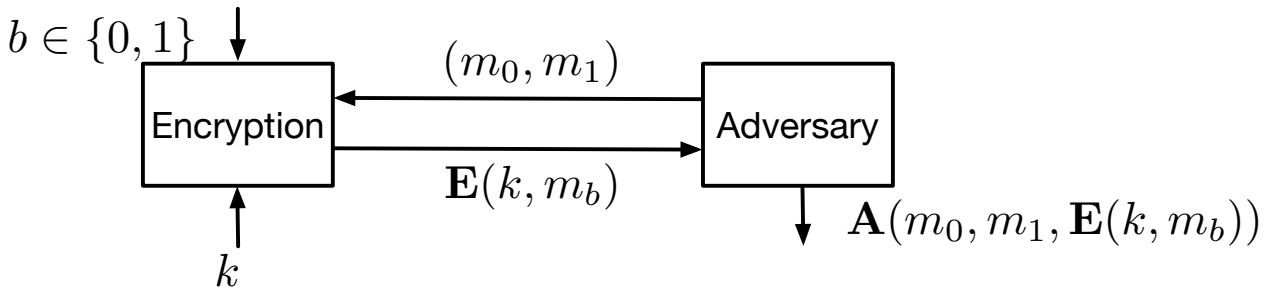
Definition (Semantic Security against CT Attacks)

A cipher $(\mathbf{E}, \mathbf{D})$ is *semantically secure* against CT attacks if for all "efficient" algorithms $\mathbf{A}$ and all $m_0, m_1 \in \mathcal{M}$ of same length, the advantage

$$\mathsf{Adv}_{SS-CT}(\mathbf{A}) := \left| \Pr\left[\mathbf{A}(m_0, m_1, \mathbf{E}(k, m_1)) = 1\right] - \Pr\left[\mathbf{A}(m_0, m_1, \mathbf{E}(k, m_0)) = 1\right] \right|$$

is negligible:

$$\mathsf{Adv}_{SS-CT}(\mathbf{A}) \leq \epsilon.$$



Question:

1. Is the One Time Pad semantically secure under a ciphertext attack?

2. Consider a Cipher $(\mathbf{E}, \mathbf{D})$ for which there exists an efficient algorithm $\mathbf{A}$ that based on a ciphertext always finds the XOR of all bits of the plaintext. Is this cipher semantically secure under a ciphertext attack?

A Stream Cipher with a Secure PRG is Semantically Secure

Theorem

A stream cipher with a secure PRG $\mathbf{G}$ *is semantically secure under a CT attack.*

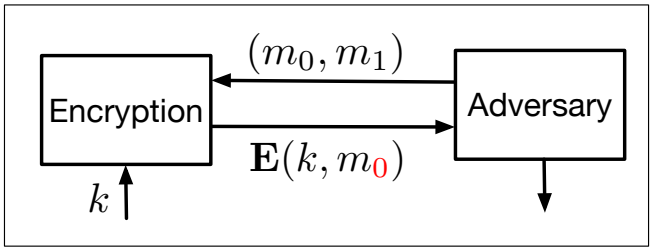
Proof: For any SS-CT attacker $\mathbf{A}$ there exists PRG attackers $\mathbf{F}_0$ and $\mathbf{F}_1$ s.t.:

$$\mathsf{Adv}_{SS-CT}(\mathbf{A}) \leq \mathsf{Adv}_{PRG}(\mathbf{F}_0, \mathbf{G}) + \mathsf{Adv}_{PRG}(\mathbf{F}_1, \mathbf{G}).$$

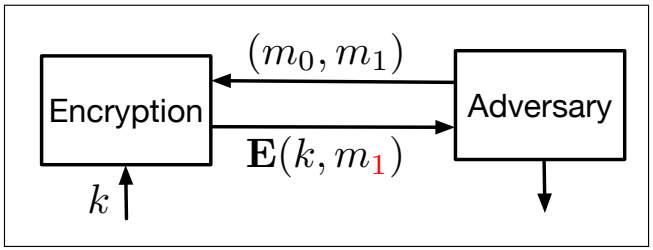
The inequality is proved using the triangle Inequality.

$$\left|\mathsf{Pr}_1[\mathbf{A}=1] - \mathsf{Pr}_0[\mathbf{A}=1]\right| \leq \underbrace{\left|\mathsf{Pr}_1[\mathbf{A}=1] - \mathsf{Pr}_{S1}[\mathbf{A}=1]\right|}_{\leq \mathsf{Adv}_{PRG}(\mathbf{F}_1,\mathbf{G})} + \underbrace{\left|\mathsf{Pr}_0[\mathbf{A}=1] - \mathsf{Pr}_{S0}[\mathbf{A}=1]\right|}_{\leq \mathsf{Adv}_{PRG}(\mathbf{F}_0,\mathbf{G})}$$

$\mathrm{Pr}_0[\boldsymbol{A}=1]$



$\mathrm{Pr}_1[\boldsymbol{A}=1]$



A Stream Cipher with a Secure PRG is Semantically Secure

Theorem

A stream cipher with a secure PRG $\mathbf{G}$ *is semantically secure under a CT attack.*

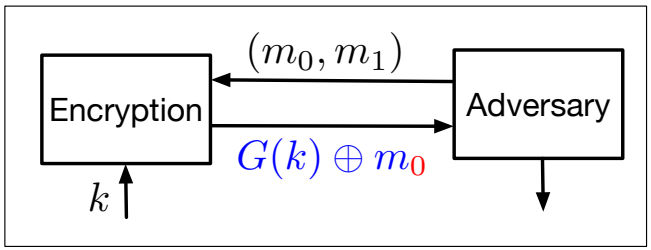
Proof: For any SS-CT attacker $\mathbf{A}$ there exists PRG attackers $\mathbf{F}_0$ and $\mathbf{F}_1$ s.t.:

$$\mathsf{Adv}_{SS-CT}(\mathbf{A}) \leq \mathsf{Adv}_{PRG}(\mathbf{F}_0, \mathbf{G}) + \mathsf{Adv}_{PRG}(\mathbf{F}_1, \mathbf{G}).$$

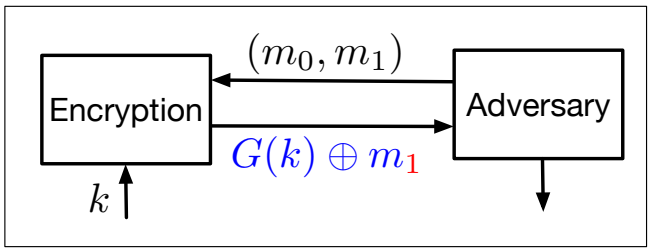
The inequality is proved using the triangle Inequality.

$$\left|\Pr_1[\mathbf{A}=1] - \Pr_0[\mathbf{A}=1]\right| \leq \underbrace{\left|\Pr_1[\mathbf{A}=1] - \Pr_{S1}[\mathbf{A}=1]\right|}_{\leq \mathsf{Adv}_{PRG}(\mathbf{F}_1,\mathbf{G})} + \underbrace{\left|\Pr_0[\mathbf{A}=1] - \Pr_{S0}[\mathbf{A}=1]\right|}_{\leq \mathsf{Adv}_{PRG}(\mathbf{F}_0,\mathbf{G})}$$

$\Pr_0[\boldsymbol{A}=1]$



$\Pr_1[\boldsymbol{A}=1]$



A Stream Cipher with a Secure PRG is Semantically Secure

Theorem

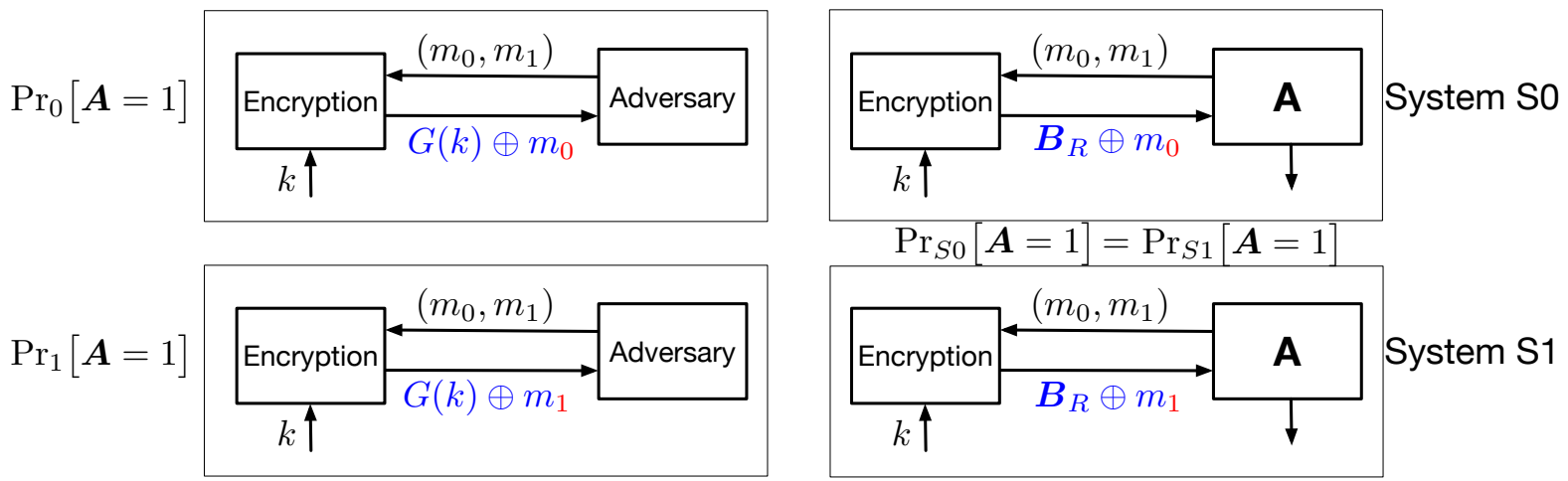
A stream cipher with a secure PRG $\mathbf{G}$ *is semantically secure under a CT attack.*

Proof: For any SS-CT attacker $\mathbf{A}$ there exists PRG attackers $\mathbf{F}_0$ and $\mathbf{F}_1$ s.t.:

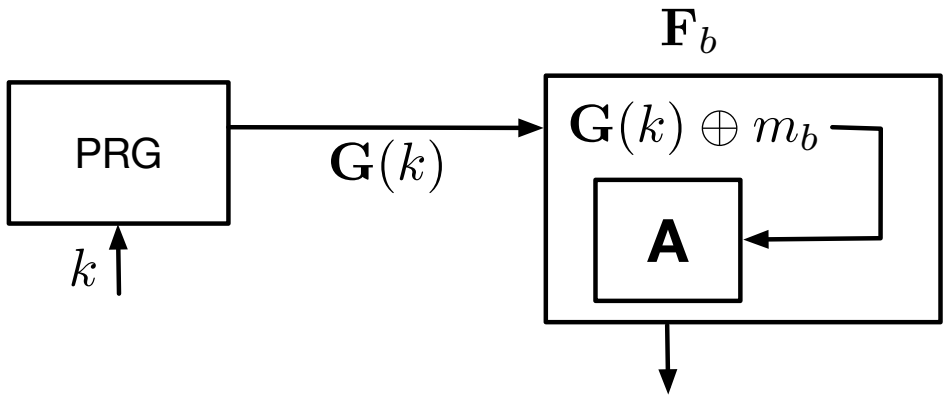
$$\mathsf{Adv}_{SS-CT}(\mathbf{A}) \leq \mathsf{Adv}_{PRG}(\mathbf{F}_0, \mathbf{G}) + \mathsf{Adv}_{PRG}(\mathbf{F}_1, \mathbf{G}).$$

The inequality is proved using the triangle Inequality.

$$\left|\Pr_1[\mathbf{A}=1] - \Pr_0[\mathbf{A}=1]\right| \leq \underbrace{\left|\Pr_1[\mathbf{A}=1] - \Pr_{S1}[\mathbf{A}=1]\right|}_{\leq \mathsf{Adv}_{PRG}(\mathbf{F}_1,\mathbf{G})} + \underbrace{\left|\Pr_0[\mathbf{A}=1] - \Pr_{S0}[\mathbf{A}=1]\right|}_{\leq \mathsf{Adv}_{PRG}(\mathbf{F}_0,\mathbf{G})}$$



The PRG Attackers $\mathbf{F}_0$ and $\mathbf{F}_1$



- Notice that for each $b \in \{0, 1\}$:

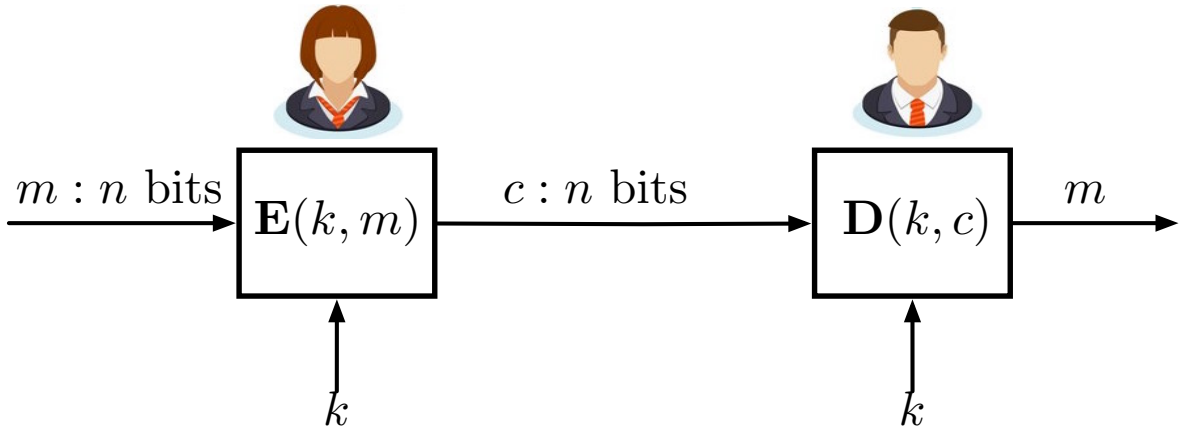
$$\Pr_b[\mathbf{A} = 1] = \Pr[\mathbf{F}_b(\mathbf{G}(K)) = 1]$$

and

$$\Pr_{Sb}[\mathbf{A} = 1] = \Pr[\mathbf{F}_b(\mathbf{B}_R) = 1]$$

Block Ciphers

Block Ciphers



- Encryption/Decryption in blocks not in streams
- Examples: DES, 3DES, AES
- Slowlier than stream ciphers but more secure
- Abstraction: A block cipher is a *Pseudorandom Function/Permutation (PRF/PRP)*

Pseudo Random Function/Pseudorandom Permutations (PRF/PRP)

Definition

A *Pseudo Random Function (PRF)* over a key space $\mathcal{K}$ and a domain $\mathcal{X}$ is a function

$$\mathbf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$$

that can "efficiently" be evaluated by a practical algorithm.

Definition

A *Pseudo Random Permuation (PRP)* over a key space $\mathcal{K}$ and a domain $\mathcal{X}$ is a function

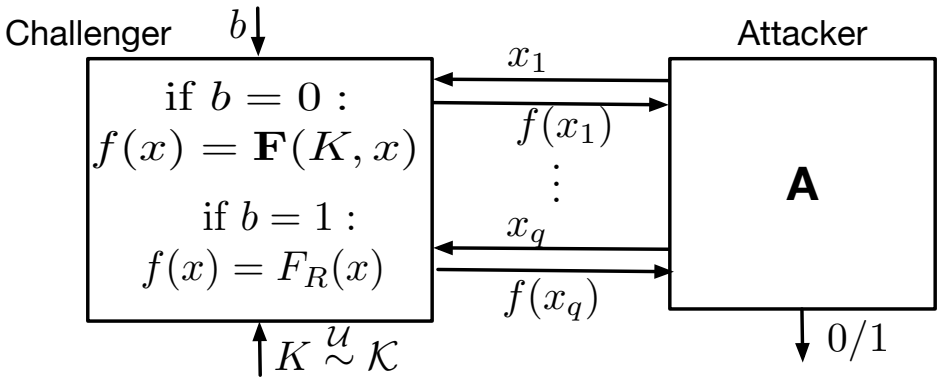
$$\mathbf{P}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

that can "efficiently" be evaluated and for each $k \in \mathcal{K}$ the function $\mathbf{P}(k, \cdot)$ is one-to-one and can efficiently be *inverted*.

Security Definitions of PRFs and PRPs

Definition

A *PRF* $\mathbf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ or a *PRP* $\mathbf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ is called *secure* if it is *computationally indistinguishable* from a random function/permutation that is picked uniformly at random from the set of *all possible* functions/permutations on the given alphabets.



Secure PRF $\mathbf{F}$:

$$\mathsf{Adv}_{PRF}(\mathbf{A}, \mathbf{F}) := \Big|\Pr\big[\mathbf{A}\big(x_1, \mathbf{F}(K, x_1), \ldots, x_q, \mathbf{F}(K, x_q)\big) = 1\big] - \Pr\big[\mathbf{A}\big(x_1, \mathbf{F}_R(x_1), \ldots, x_q, \mathbf{F}_R(x_q)\big) = 1\big]\Big| \leq \epsilon, \quad \text{negligible}$$

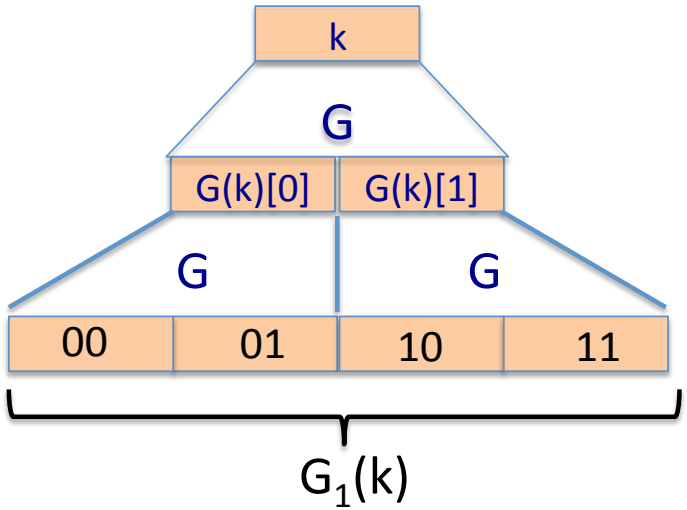
Generating a Secure PRG from a Secure PRF

- Let $\mathbf{F}\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF.
- From $\mathbf{F}$ one can construct the secure PRG $\mathbf{G}\colon \mathcal{K} \to \{0,1\}^{tn}$

$$\mathbf{G}(k) = \mathbf{F}(k,0) \,||\, \mathbf{F}(k,1) \,||\, \mathbf{F}(k,2) \,||\, \mathbf{F}(k,3) \cdots \,||\, \mathbf{F}(k,t-1)$$

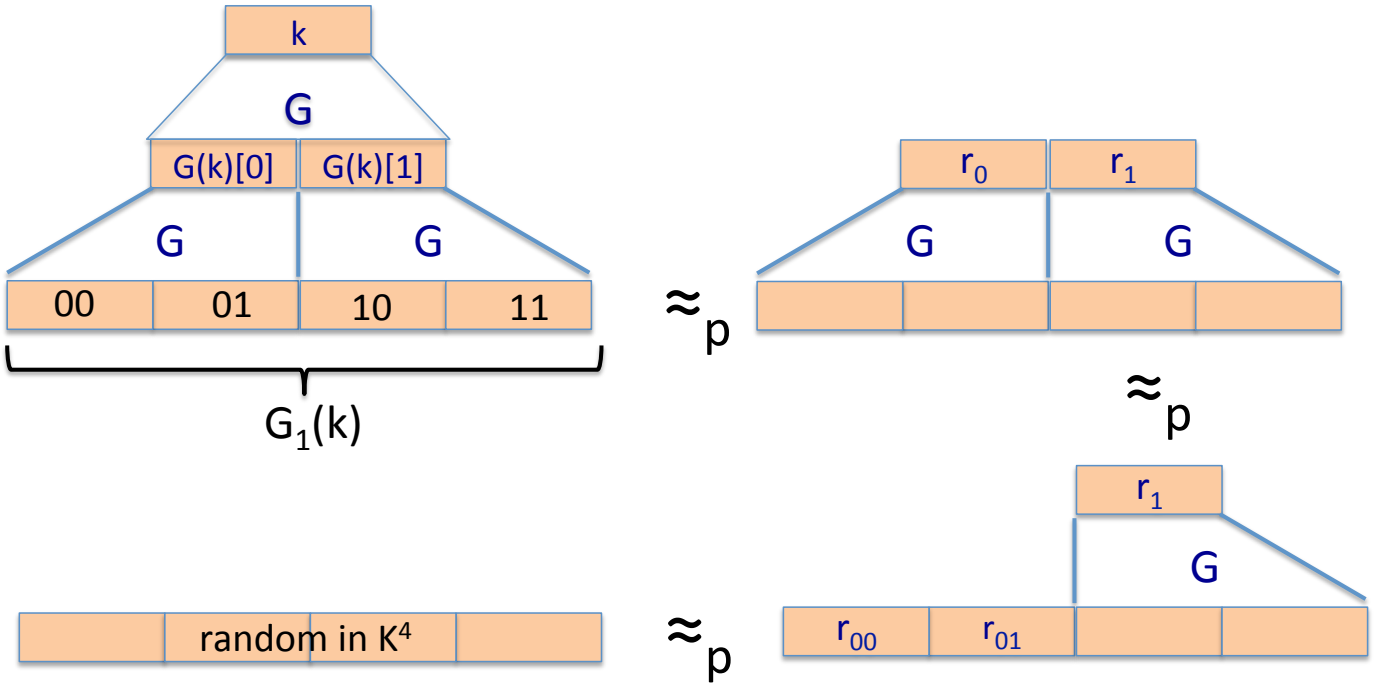
Generating a Secure PRF from a Secure PRG

- Let $\mathbf{G}\colon \mathcal{K} \to \mathcal{K}^2$ be a secure PRG
- A secure PRF $\mathbf{G}\colon \mathcal{K} \times \{0,1\}^2 \to \mathcal{K}$ is obtained by:



- Applying $\mathbf{G}$ m times $\to$ arbitrary secure PRFs $\mathcal{K} \times \{0,1\}^m \to \mathcal{K}$
- Too slow for practice
- A secure PRF can be transformed into a secure PRP (block cipher) by the Luby-Rackoff Theorem (which states that each secure PRF can be transformed into a secure PRP and is discussed shortly)

Security Proof of PRG-to-PRF construction



PRF Switching Lemma: Secure PRPs that are also Secure PRFs

- Recall that a PRP is also a PRF.

Lemma

Consider the PRP $\mathbf{P}: \mathcal{K} \times \mathcal{X} \to \mathcal{X}$. *For any adversary* $\mathbf{A}$ *asking* q *queries:*

$$\left|\mathsf{Adv}_{PRP}(\mathbf{A}, \mathbf{P}) - \mathsf{Adv}_{PRF}(\mathbf{A}, \mathbf{P})\right| < \frac{q^2}{2|\mathcal{X}|}$$

So any secure PRP with sufficiently large domain $|\mathcal{X}|$, is also a secure PRF. The intuitive reason is that for large domains, very likely the set of function values that an efficient adversary can evaluate will all be different. For purely random functions, the probability that q different function values are all different can be bounded by $\frac{q^2}{2|\mathcal{X}|}$.

Proof of the PRF Switching Lemma

$$
\begin{aligned}
&\left|\mathrm{Adv}_{PRP}(\mathbf{A},\mathbf{P}) - \mathrm{Adv}_{PRF}(\mathbf{A},\mathbf{P})\right| \\
&= \Big|\big|\underbrace{\Pr\big[\mathbf{A}\big(\{x_i,\mathbf{P}(K,x_i)\}_{i=1}^q\big)=1\big]}_{p_0} - \underbrace{\Pr\big[\mathbf{A}\big(\{x_i,\mathbf{P}_R(x_i)\}_{i=1}^q\big)=1\big]}_{p_1}\big| \\
&\qquad -\big|\underbrace{\Pr\big[\mathbf{A}\big(\{x_i,\mathbf{P}(K,x_i)\}_{i=1}^q\big)=1\big]}_{p_0} - \underbrace{\Pr\big[\mathbf{A}\big(\{x_i,\mathbf{F}_R(x_i)\}_{i=1}^q\big)=1\big]}_{p_2}\big|\Big| \\
&\leq |p_1 - p_2| \\
&\overset{(a)}{\leq} \Pr\big[\{\mathbf{F}_R(x_i)\}_{i=1}^q \text{ are not all distinct}\big] \\
&\leq \sum_{1\leq i<j\leq q} \Pr\big[\mathbf{F}_R(x_i)=\mathbf{F}_R(x_j)\big] \leq \frac{q(q-1)}{2}\cdot\frac{1}{|\mathcal{X}|}
\end{aligned}
$$

Here (a) follows from the difference lemma on the next slide where we set:

- $\epsilon_1 : \{\mathbf{A}\big(\{x_i,\mathbf{P}_R(K,x_i)\}_{i=1}^q\big)=1\}$
- $\epsilon_2 : \{\mathbf{A}\big(\{x_i,\mathbf{F}_R(K,x_i)\}_{i=1}^q\big)=1\}$
- $\epsilon_Z : \{\{\mathbf{F}_R(x_i)\}_{i=1}^q \text{ are not all distinct}\}$

The Difference Lemma

Lemma

Let $\epsilon_1, \epsilon_2, \epsilon_Z$ be events on the same probability space satisfying

$$\epsilon_1 \cap \epsilon_z^c = \epsilon_2 \cap \epsilon_z^c,$$

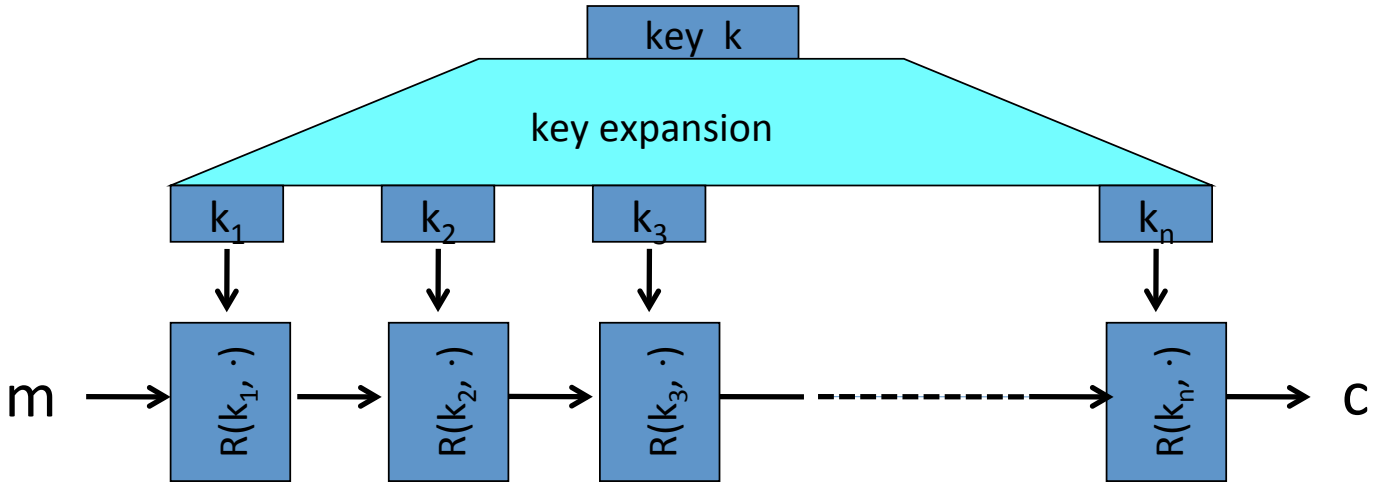
where ϵ_z^c denotes the complement of event ϵ_z.
Then,

$$\big|\mathbb{P}[\epsilon_1] - \mathbb{P}[\epsilon_2]\big| \leq \mathbb{P}[\epsilon_z]$$

Proof:

$$\begin{aligned}
&\big|\mathbb{P}[\epsilon_1] - \mathbb{P}[\epsilon_2]\big| \\
&= \big|\mathbb{P}[\epsilon_1 \cap \epsilon_z^c] - \mathbb{P}[\epsilon_2 \cap \epsilon_z^c] + \mathbb{P}[\epsilon_1 \cap \epsilon_z] - \mathbb{P}[\epsilon_2 \cap \epsilon_z]\big| \\
&= \mathbb{P}[\epsilon_1 \cap \epsilon_z] - \mathbb{P}[\epsilon_2 \cap \epsilon_z]\big| \\
&= \big|\mathbb{P}[\epsilon_1 \cap \epsilon_z^c \cap \epsilon_2] - \mathbb{P}[\epsilon_2 \cap \epsilon_z^c \cap \epsilon_1] + \mathbb{P}[\epsilon_1 \cap \epsilon_z \cap \epsilon_2^c] - \mathbb{P}[\epsilon_2 \cap \epsilon_z \cap \epsilon_1^c]\big| \\
&= \big|\mathbb{P}[\epsilon_1 \cap \epsilon_z \cap \epsilon_2^c] - \mathbb{P}[\epsilon_2 \cap \epsilon_z \cap \epsilon_1^c]\big| \\
&\leq \mathbb{P}[\epsilon_1 \cap \epsilon_z \cap \epsilon_2^c] + \mathbb{P}[\epsilon_2 \cap \epsilon_z \cap \epsilon_1^c] \\
&\leq \mathbb{P}[\epsilon_z]
\end{aligned}$$

Block Ciphers are Built by Iterations



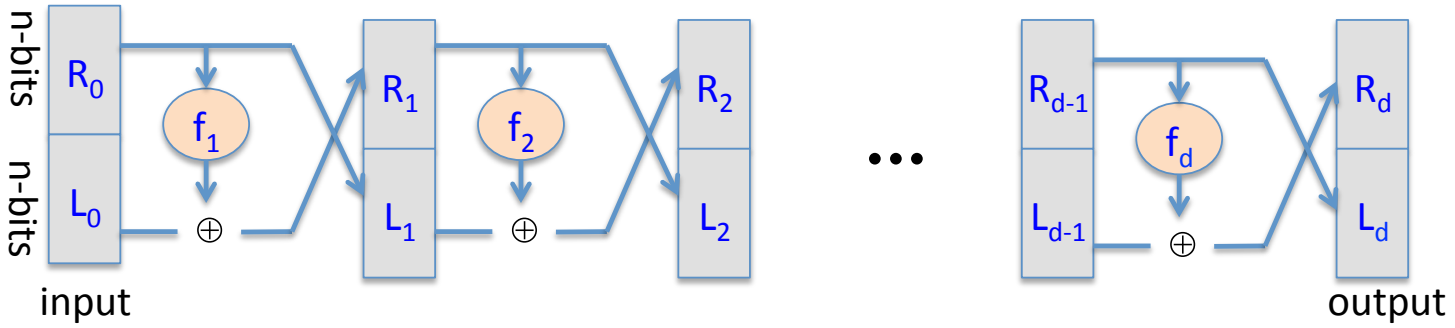
Round function $R(\cdot, \cdot)$.

Data Encryption Standard (DES)

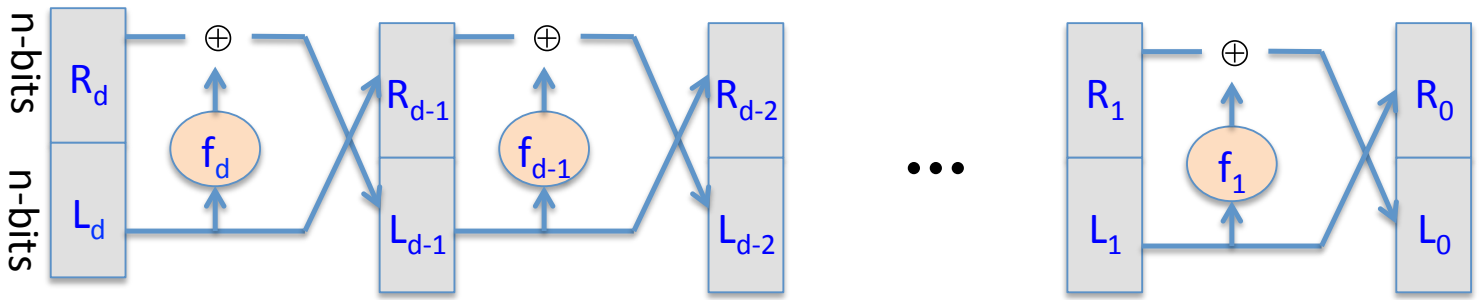
- 1970: Invented at IBM with 128 keylength and blocklength ($n = k = 128$)
- 1976: Adopted as a federal standard with 56 bits keylength and 64 bits blocklength ($n = 64$ and $k = 56$)
- 1997: DES broken by exhaustive search
- 2000 Replaced as a standard by AES

Feistel Network

- Arbitrary functions $f_1, \ldots, f_d : \{0,1\}^n \to \{0,1\}^n$



- Easy to invert:



- Method used in many block ciphers

Theorem of Luby-Rackoff'85

Theorem

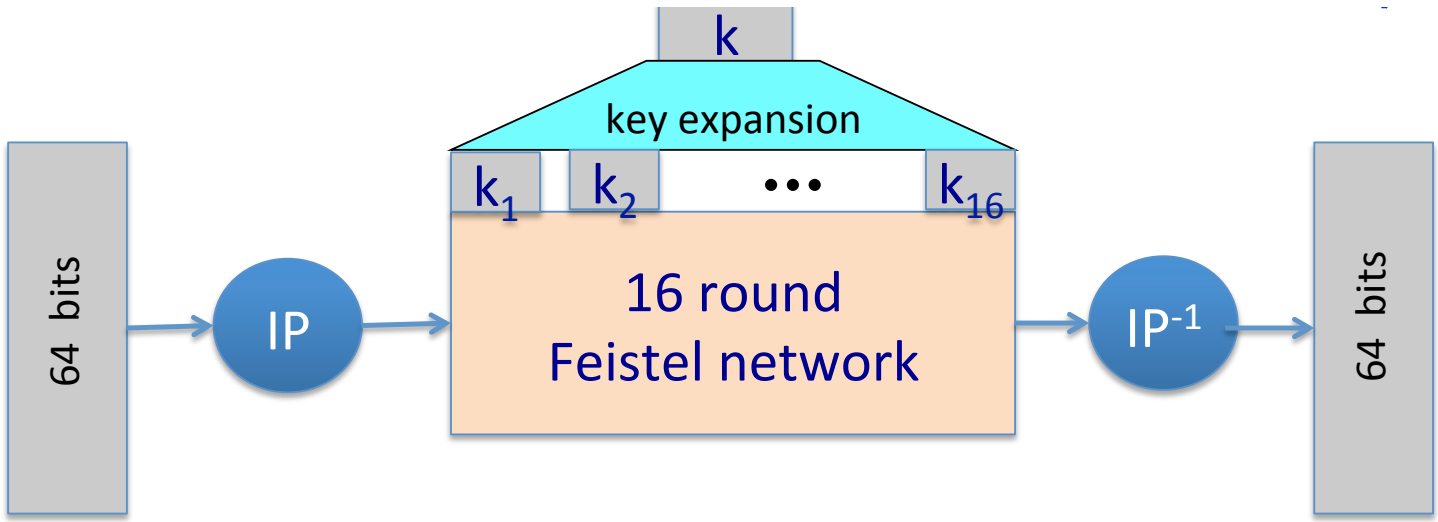
Consider a Feistel network with 3 rounds and
$f_1 = f_2 = f_3 : \mathcal{K} \times \{0,1\}^n \to \{0,1\}$ *a secure PRF.*
If a different random key is used for each function f_1, f_2, f_3, *the Feistel Network implements a secure PRP.*

$\to$ So from a secure PRF we can construct a secure PRP!

Back to DES

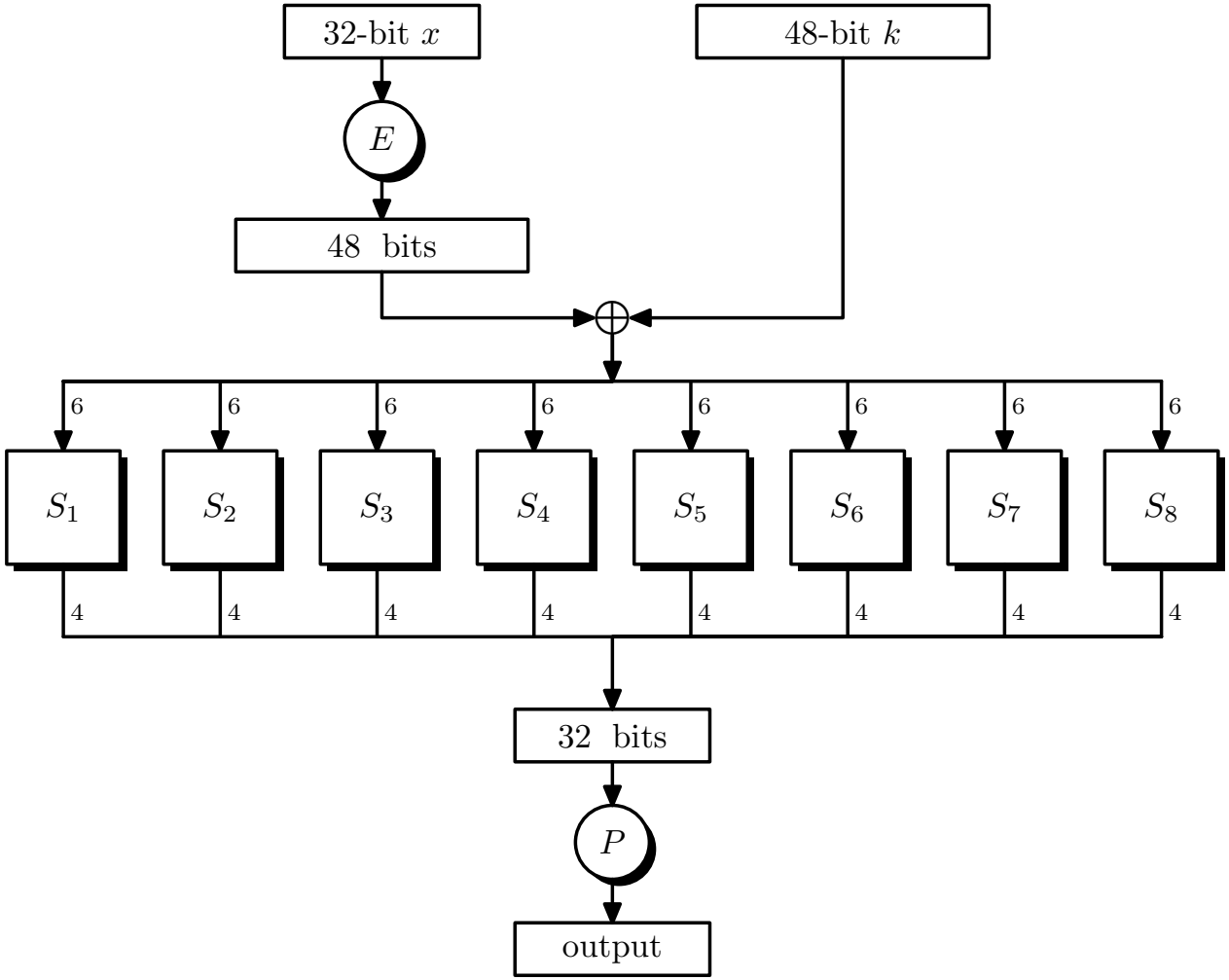
- DES is a 16-round Feistel network with functions $f_i : \{0,1\}^{32} \to \{0,1\}^{32}$

$$f_i(x) = \mathbf{F}(k_i, x) \qquad \mathbf{F} \text{ described on next slide}$$



- To invert, use keys in reverse order

The **F** function of DES



- Permutations E and P and S-boxes are carefully designed using lookup tables and were revealed only in 1994.

Design Criteria

- Choosing P and S-boxes at random would result in an insecure cipher.
- If S-boxes are linear, the whole cipher is linear.
- S-boxes should be 4-to-1.

Exhaustive Search Attack on DES

- Exhaustive search attacks with only 3 plaintext-ciphertext pairs is possible in few days (2007) because 56 bits keys are too short!

- Hint: If DES was a purely random permutation, then the probability of two keys leading to the same three plaintext-ciphertext pairs is:

$$\Pr\left[\exists k' \neq k : DES(m_i, k) = DES(m_i, k'), \forall i = 1, 2, 3\right]$$

$$= \sum_{k' \neq k} \prod_{i=1}^{3} \Pr\left[DES(k, m_i) = DES(k', m_i)\right] = 2^{56} \left(\frac{1}{2^{64}}\right)^3 = 2^{-136}.$$

$\rightarrow$ So most likely, 3 plaintext-ciphertext pairs allow for an exhaustif search attack on the key.

Linear and Differential Attacks on DES

Theorem

Assume that the probability (over random message m and key k)

$$\Pr\Big[m(i_1) \oplus \cdots \oplus m(i_r) \bigoplus c(j_1) \oplus \cdots \oplus c(j_v) = k(l_1) \oplus \cdots \oplus k(l_u)\Big] = \frac{1}{2} + \epsilon,$$

then with ϵ^{-2} plaintext-ciphertext pairs $\{(m_\xi, c_\xi = DES(k, m_\xi))\}_{\xi=1}^{\epsilon^{-2}}$:

$$\Pr\Big[k(l_1) \oplus \cdots k(l_u) = \mathrm{MAJ}_{\xi=1,\ldots,\epsilon^{-2}}\big(m_\xi(i_1) \oplus \cdots m_\xi(i_r) \bigoplus c_\xi(j_1) \oplus \cdots c_\xi(j_v)\big)\Big] \geq 0.977$$

$\rightarrow$ can find $k(l_1) \oplus \cdots k(l_u)$ with high probability in ϵ^{-2} time

- For DES:
  - $\epsilon = 2^{-21}$
  - this way we can find 14 key bits in time 2^{42}
  - find the rest by brute force in time 2^{42}

Triple DES (3DES)

- Split initial key into 3: $k = (k^{(1)}, k^{(2)}, k^{(3)})$ (all purely random keys)
- Use original DES cipher $(\mathbf{E}, \mathbf{D})$ to create the Triple-DES cipher $(\mathbf{3E}, \mathbf{3D})$

$$\begin{aligned} \mathbf{3E}(k, m) &= \mathbf{E}(k^{(1)}, \mathbf{D}(k^{(2)}, \mathbf{E}(k^{(3)}, m))) \\ \mathbf{3D}(k, m) &= \end{aligned}$$

- Keylength is $3 \cdot 56$ bits $= 168$bits

$\rightarrow$ Effective security seems to be 118 bits of key, i.e., $\exists$ an attack in 2^{118} time

Meet-in-the-Middle Attack on Double DES

- Split initial key into 2: $k = (k^{(1)}, k^{(2)}$ (all purely random keys)

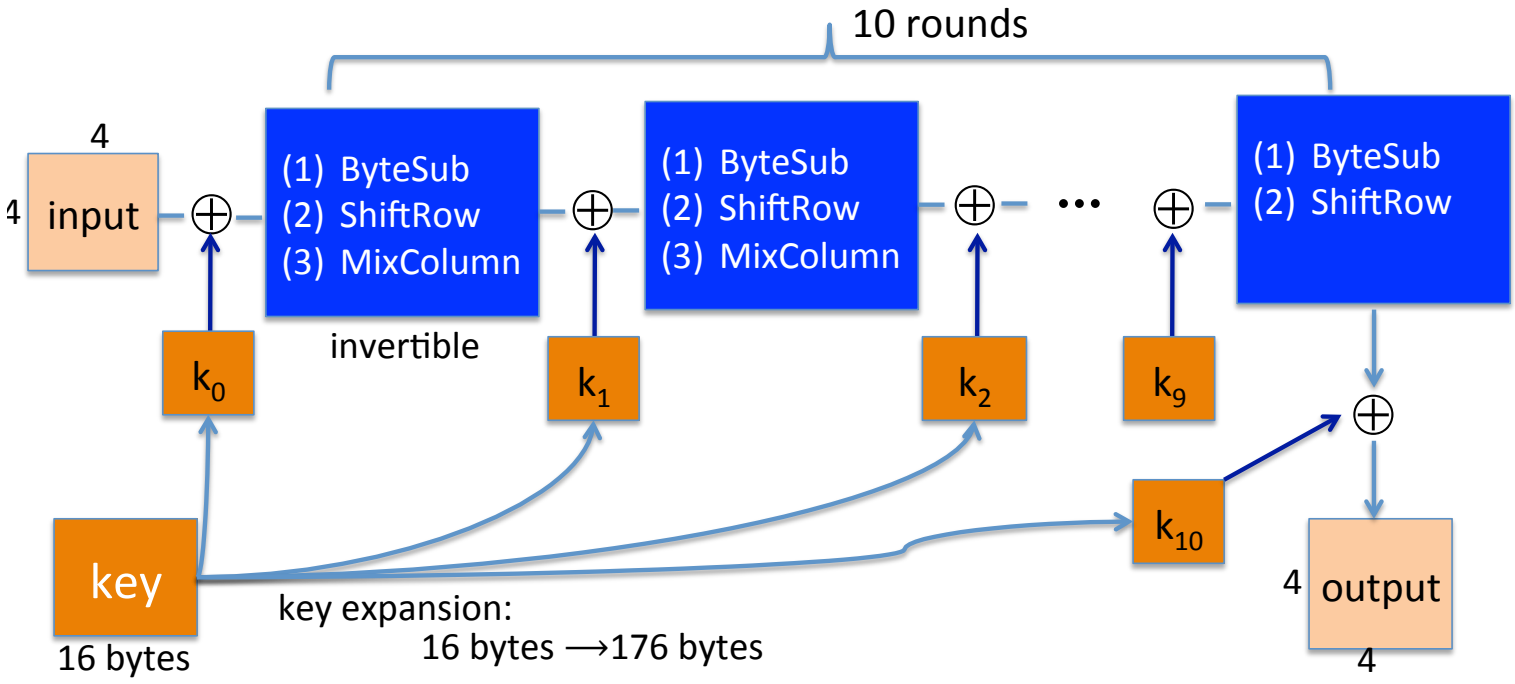
- Use original DES cipher $(\mathbf{E}, \mathbf{D})$ to create the Double-DES cipher $(\mathbf{2E}, \mathbf{2D})$

$$\begin{aligned} \mathbf{2E}(k, m) &= \mathbf{E}(k^{(1)}, \mathbf{E}(k^{(2)}, m)) \\ \mathbf{2D}(k, m) &= \end{aligned}$$

- Chosen plain-text attack in 2^{56} time with 2^{56} space:
  - Precalculate a table with 2^{56} entries $\mathbf{E}(k^{(1)}, m)$, for each possible key $k^{(1)}$
  - After receiving c, calculate $\mathbf{D}(k^{(2)}, c)$ for all keys $k^{(2)}$ and search it in the table $\rightarrow$ gives $k^{(1)}$ and $k^{(2)}$

$\rightarrow$ same effective keylength as single DES!

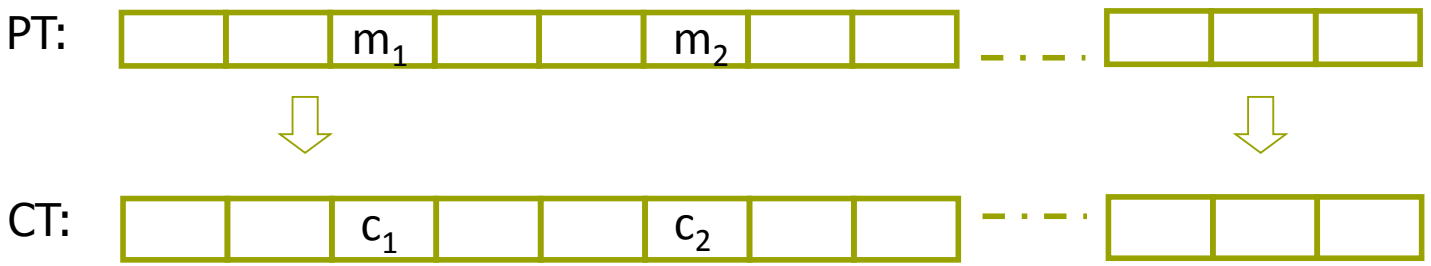
American Encryption Standard (AES)



- Byte Sub are simple lookup tables like the S-boxes:
  each byte is permuted according to this S-box
- Effective security $\approx$ 99 bits of keylength
- MixColumn omitted from the last round
  (to make decryption algorithm more similar to encryption)

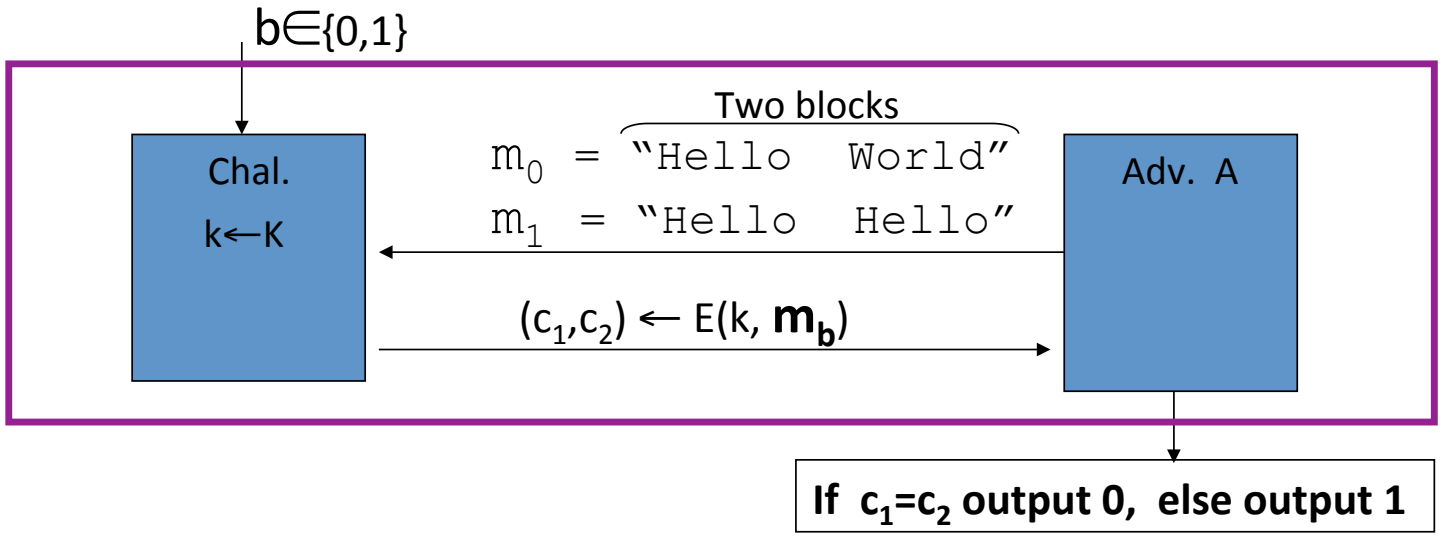
Using Block Ciphers: Electronic Code Book (ECB) Mode

- Use same key k and same block cipher to encode multiple messages



- If $m_1 = m_2$ then $c_1 = c_2$ $\rightarrow$ cannot be semantically secure!

ECB Mode is not Semantically Secure

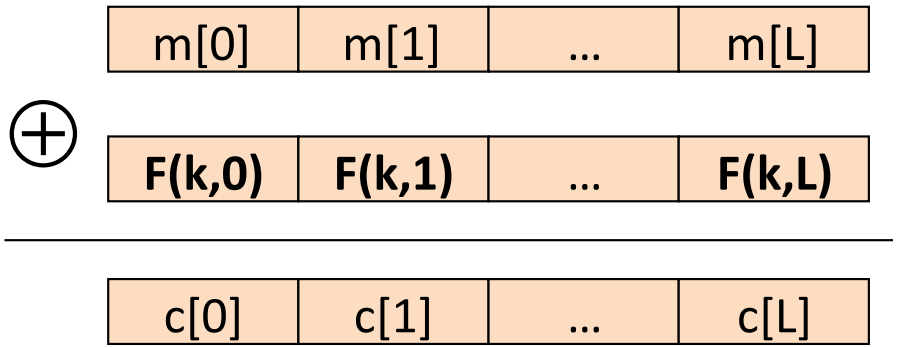


- If $(\mathbf{E}, \mathbf{D})$ was a truly random permutation, the attacker's advantage $\mathrm{Adv}_{SS-CT}(\mathbf{A}) = 1$

Deterministic Counter Mode

- Build a stream cipher from a *secure PRF*

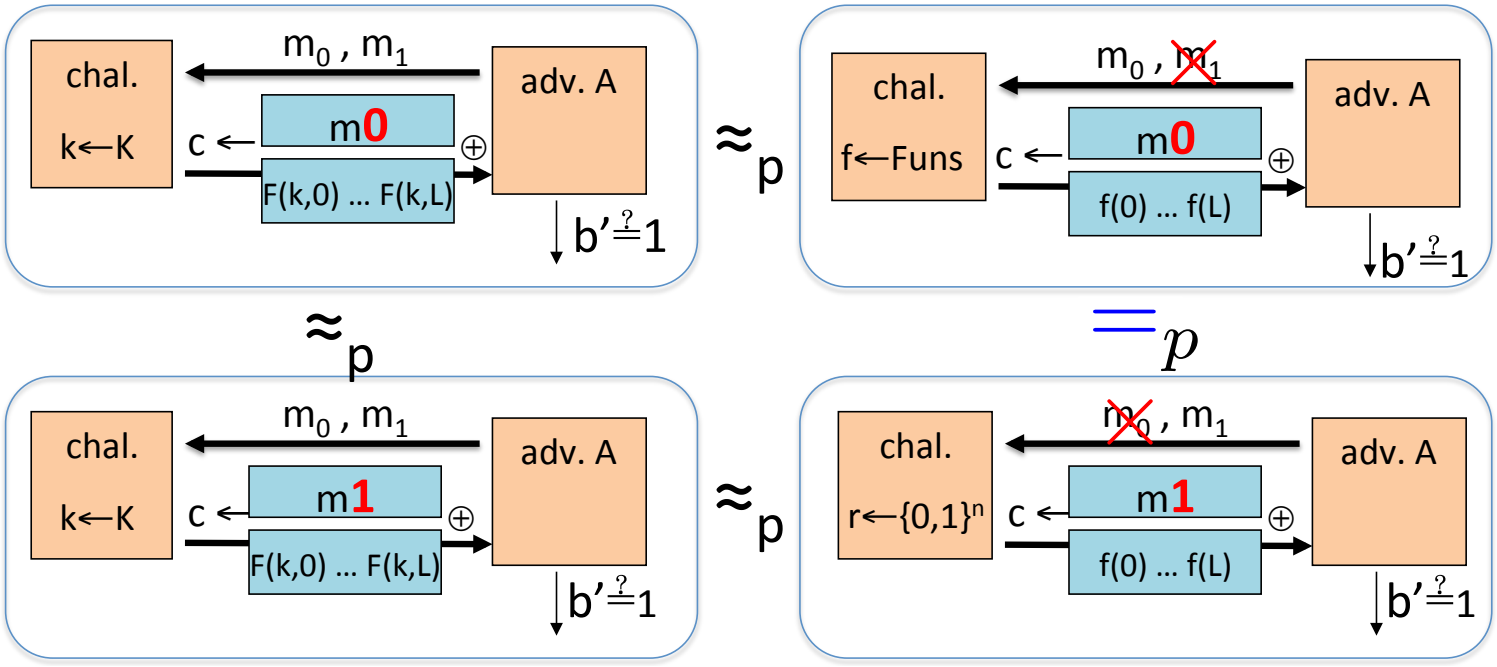
$$\mathbf{F}: \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$$



Theorem

For any $L > 0$ Determinstic Counter Mode produces a semantically secure cipher.

Deterministic Counter Mode is Secure



- f is a purely random function
- $\forall \mathbf{A}$ there $\exists$ PRF adversary $\mathbf{B}$ such that

$$\mathrm{Adv}^{ss}(\mathbf{A}, \mathbf{F}) = 2\mathrm{Adv}^{ss}(\mathbf{B}, \mathbf{F})$$

- $\mathbf{B}(x, \mathbf{F}(x)) = \mathbf{A}(x, \mathbf{F}(x) \oplus x)$

Semantic Security against Chosen Ciphertext Attacks (CPA)

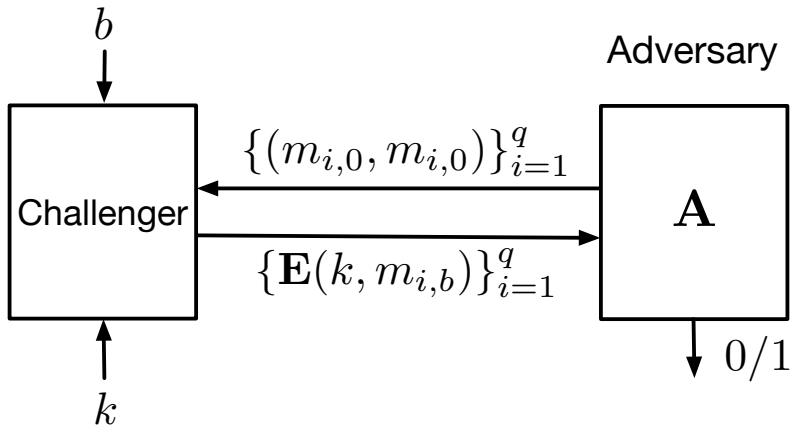
- *Chosen Ciphertext Attack (CPA)*: An adversary obtains ciphertexts for q different plaintexts under the same encryption key

Definition (Semantic Security against CT Attacks)

A cipher $(\mathbf{E}, \mathbf{D})$ is *semantically secure* against CPA if $\forall$ "efficient" algorithms $\mathbf{A}$ and $\{m_{i,0}, m_{i,1}\}_{i=1}^{q}$ of same length, the advantage

$$\mathsf{Adv}_{SS-CPA}(\mathbf{A}) := \Big|\Pr\big[\mathbf{A} = 1 | b = 1\big] - \Pr\big[\mathbf{A} = 1 | b = 0\big]\Big|$$

is negligible.



Semantic Security of Deterministic Ciphers

Theorem

Any deterministic cipher $(\mathbf{E}, \mathbf{D})$ *that is a secure PRP is not semantically secure under a multiple CPA.*

Proof:

- Randomize encryption, or add a *nonce* to the message

Randomized Encryption

- Consider a deterministic standard block cipher $(\mathbf{E}, \mathbf{D})$ and let r be an independent randomness
- Add randomness to the plaintext: $c = \mathbf{E}_r(k, m) = (r||\mathbf{E}(k, (m||r)))$
- Decryption of ciphertext $c = (r||\tilde{c})$: $m = \mathbf{D}_r(k, c) = \mathbf{D}(k, \tilde{c})$
- Drawback: for given blockcipher we reduce the message space $\mathcal{M}$ by the randomness space $\mathcal{R}$

Question: Let $\mathbf{F}\colon \mathcal{K} \times \mathcal{R}$ be a secure PRF. Is the randomized encryption

$$\mathbf{E}_r(k, m) = [r||\mathbf{F}(k, r) \oplus m]$$

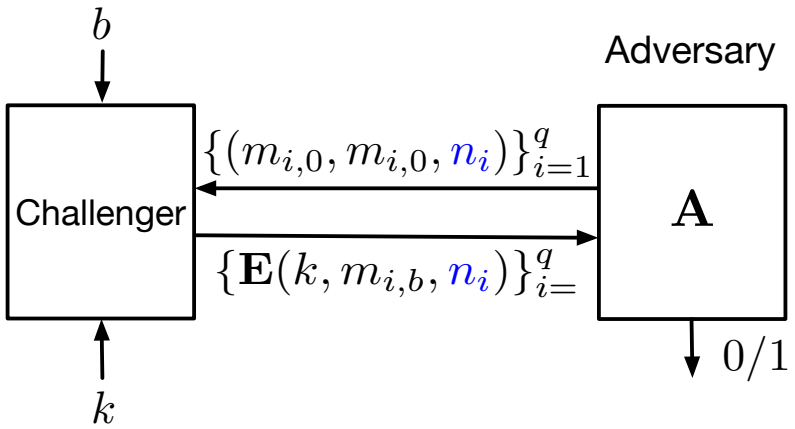
semantically secure against a CPA if each randomness is used only once?

Nonce-Based Encryption

- Similar to randomized encryption, but now we add a *nonce*, for example an increasing counter or a random nonce that is public

$$c = \mathbf{E}(k, m, n)$$

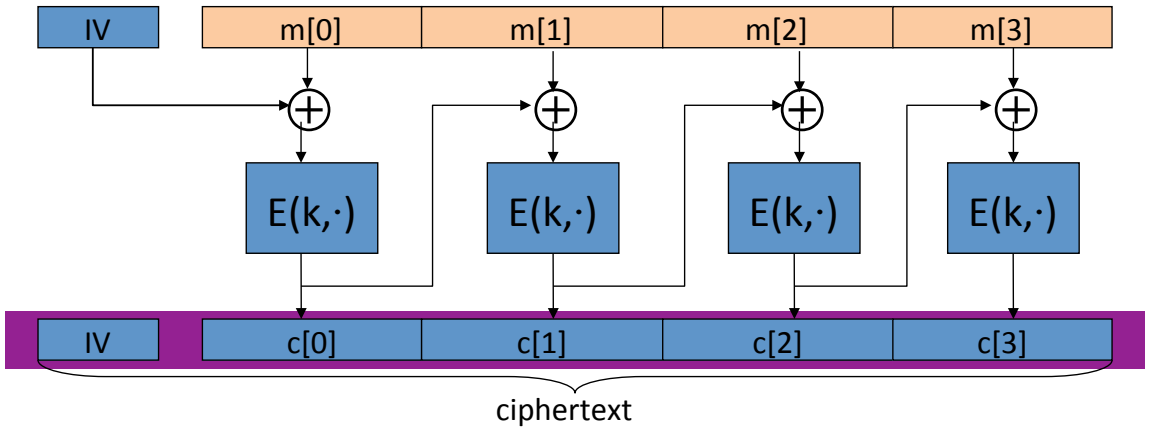
CPA game for nonce-based encryption:



- All nonces must be different.

CBC Mode: Chaining Block Ciphers

- Block cipher $(\mathbf{E}, \mathbf{D})$ (PRP) and random initialization value (IV)
- Use same key k in all L blocks!



- Decryption: $m(i) = \mathbf{D}(k, c(i)) \oplus c(i-1)$.

CBC Mode: CPA Analysis

Theorem

For any $L > 0$, if $\mathbf{E}(\cdot,\cdot)$ is a secure PRP over $\mathcal{X}$, then CBC is semantically secure against a q-query CPA if $\frac{q^2 L^2}{|\mathcal{X}|}$ is negligible

Proof Idea: If **A** is an attacker on the CBC, there exists an attacker **B** on the PRP such that:

$$\mathrm{Adv}_{SS-CPA}(\mathbf{A}) \leq 2\mathrm{Adv}_{PRP}(\mathbf{B}) + 2\frac{q^2 L^2}{|\mathcal{X}|}.$$

An Attack on CBC

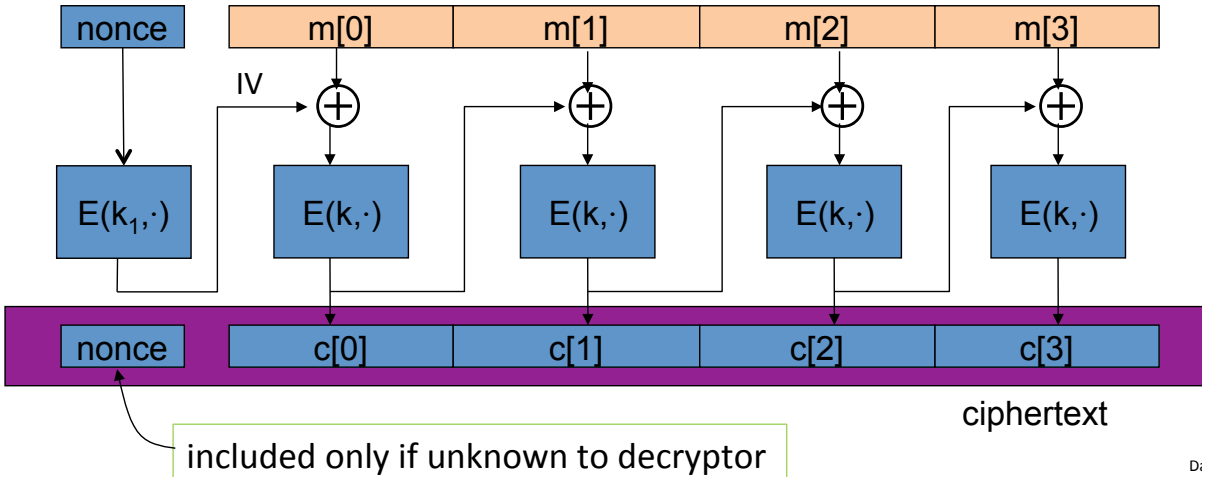
Theorem

When the attacker can predict the next IV, then CBC is not secure even if PRP is secure.

Proof:

- Bug in SSL/TLS 1.0: previous ciphertexts used for next IVs

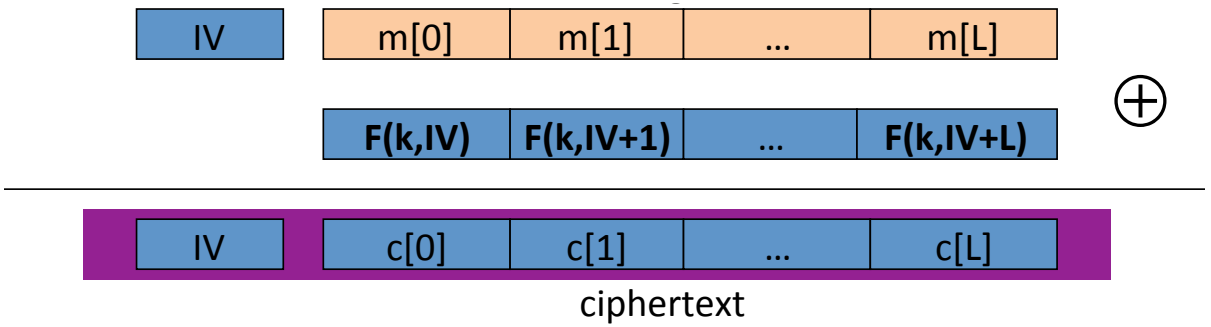
Nonce-Based CBC Mode



- Nonce needs to be encrypted!!
- Each nonce/key pair only used once!

Random Counter Mode for Block Ciphers or Stream Ciphers

- Let $\mathbf{F}\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF and IV random over $\{0,1\}^n$



- Can also be used with a nonce: IV = (nonce || counter); nonce typically does not have to be transmitted

Random Counter Mode: CPA Analysis

Theorem

For any $L > 0$, if $\mathbf{F}(\cdot,\cdot)$ is a secure PRF over $\mathcal{X}$, then Random Counter Mode is semantically secure against q-queries CPA whenever $\frac{q^2 L}{|\mathcal{X}|}$ is negligible

Proof Idea: If **A** is an attacker on the Random Counter Mode, then there exists an attacker **B** on the PRF **F** such that

$$\mathrm{Adv}_{SS-CPA}(\mathbf{A}) \leq 2\mathrm{Adv}_{PRF}(\mathbf{B}) + 2\frac{q^2 L}{|\mathcal{X}|}.$$

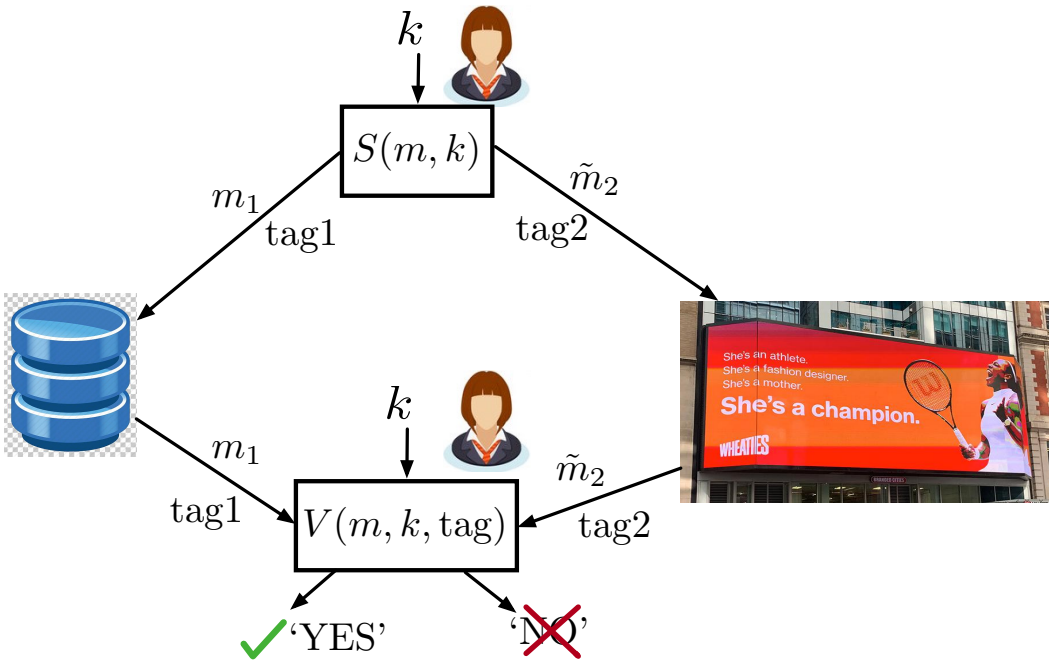
Comparison of CBC and Random Counter Mode

	CBC	Random CTR
Based on	PRP	PRF or PRP
Parallel Processing	no	yes
Secure if	$q^2 L^2 \ll \lvert\mathcal{X}\rvert$	$q^2 L \ll \lvert\mathcal{X}\rvert$

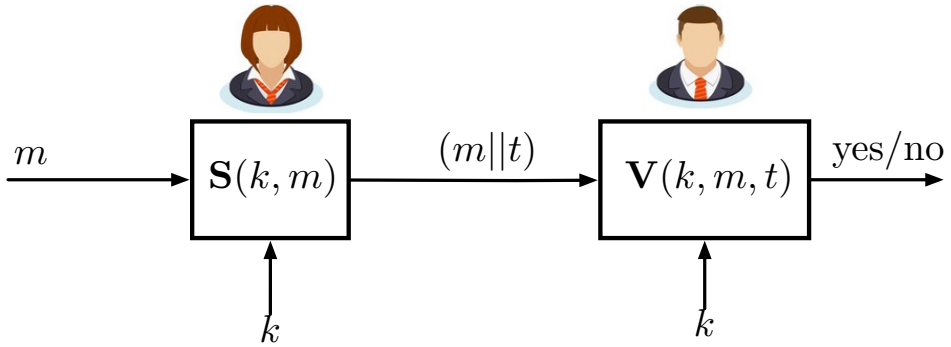
Key reuse in AES/DES when advantage should be $< 2^{32}$:

- AES has $|\mathcal{X}| = 2^{128}$ (16 bytes):
  - In CBC mode, the key needs to be changed all 2^{48} AES blocks. If we chain 2^{16} blocks we have to change the key all 2^{32} chains.
  - In Random CTR Mode, the key needs to be changed after 2^{40} chains, if each chain includes 2^{16} AES blocks
- For DES has $|\mathcal{X}| = 2^{64}$ (8 bytes):
  - In CBC mode we have to change the key all 2^{16} AES blocks. If we chain 2^8 blocks we have to change all 2^8 chains.
  - In Random CTR Mode, the key needs to be changed after 2^{12} chains, if each chain includes 2^8 AES blocks

Message Authentication Codes (MAC)



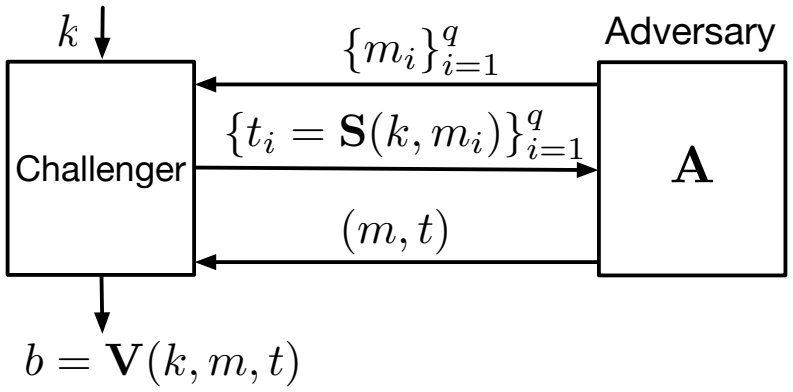
Message Authentication Codes



- Message Authentication Code (MAC) is a pair $(\mathbf{S}, \mathbf{V})$:
  - $\mathbf{S}$: $\mathcal{K} \times \mathcal{M} \to \mathcal{T}$ produces tag t
  - $\mathbf{V}$: $\mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{\text{'yes'}, \text{'no'}\}$ verifies the validity of a message-tag pair

- Existential forgery (active attack): produce an unseen (m, t) pair such that $\mathbf{V}(k, m, t) =$'yes'.

Security Definition of a MAC



Definition (Security for Message Authentication Codes)

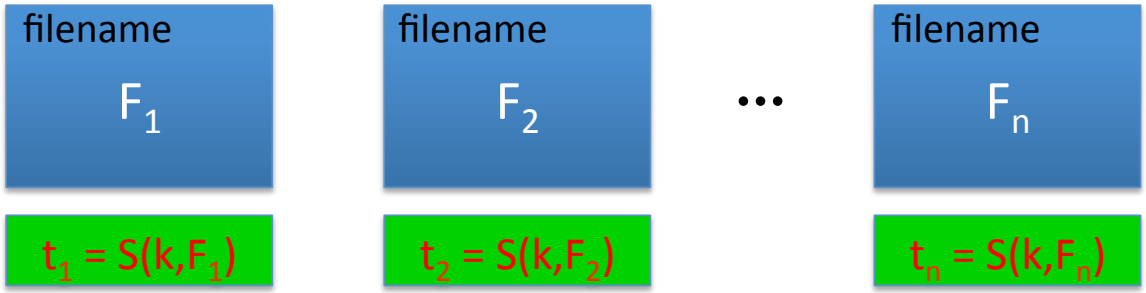
A MAC $(\mathbf{S}, \mathbf{V})$ is *secure against a chosen message attack (CMA)* if $\forall$ "efficient" algorithms $\mathbf{A}$ and $\{m_i\}_{i=1}^q$, the advantage

$$\mathrm{Adv}_{MAC}(\mathbf{A}) := \Pr\left[\mathbf{V}(K, m, t) = \text{'yes'}\right]$$

is negligible. (Here, $m_1, \ldots, m_q$ and m can be of arbitrary lengths and $(m, t) \notin \{(m_i, t_i)\}_{i=1}^q$)

- What if tag is always 3 bits, $\mathcal{T} = \{0,1\}^3$? Can such a MAC be secure?

Example: Authentication of a File System



- Key obtained from user password
- Often used to detect hardware failures or viruses at startup
- Important to also authenticate the file number, otherwise the system is not secure!

Building a (Short) MAC from a PRF

- Let $\mathbf{F}: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF
- Consider the MAC:

$$\mathbf{S}(k, m) = \mathbf{F}(k, m) \qquad \text{and} \qquad \mathbf{V}(k, m, t) = \begin{cases} \text{'yes'} & \text{if } t = \mathbf{F}(k, m) \\ \text{'no'} & \text{else} \end{cases}$$

Theorem

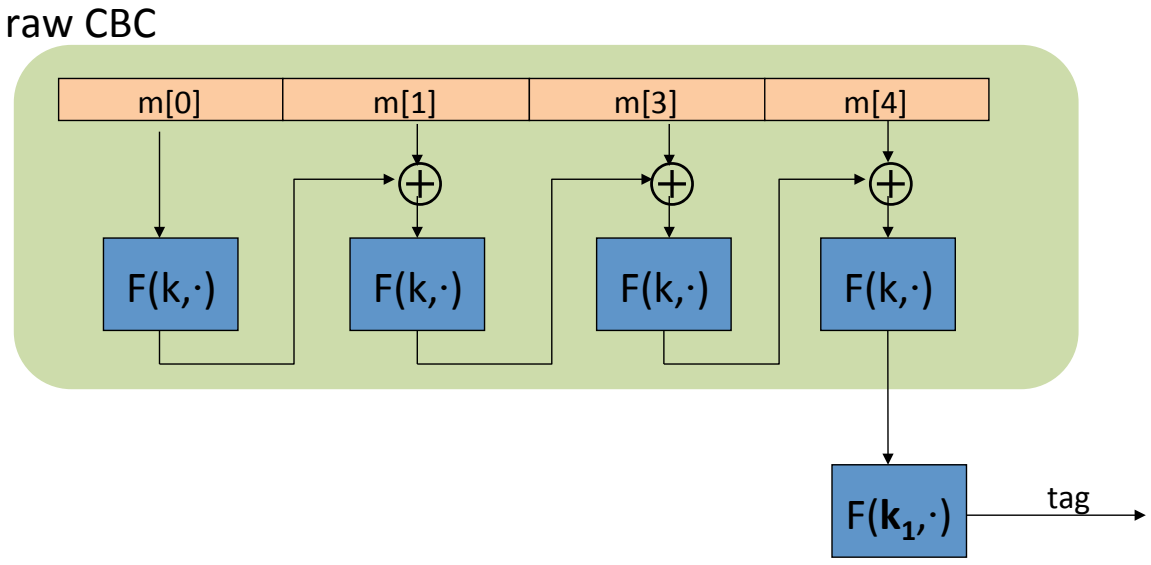
If $|\mathcal{Y}|^{-1}$ is negligible, above MAC is secure.

Proof idea: If **A** is an attacker on the MAC, there exists an attacker **B** on the PRF **F** such that

$$\text{Adv}_{MAC}(\mathbf{A}) \leq \text{Adv}_{PRF}(\mathbf{B}) + \frac{1}{|\mathcal{Y}|}.$$

- Next: How to construct long MACs

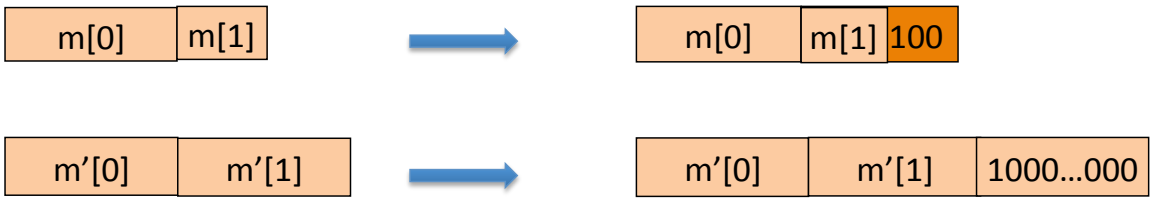
Encrypted CBC MAC



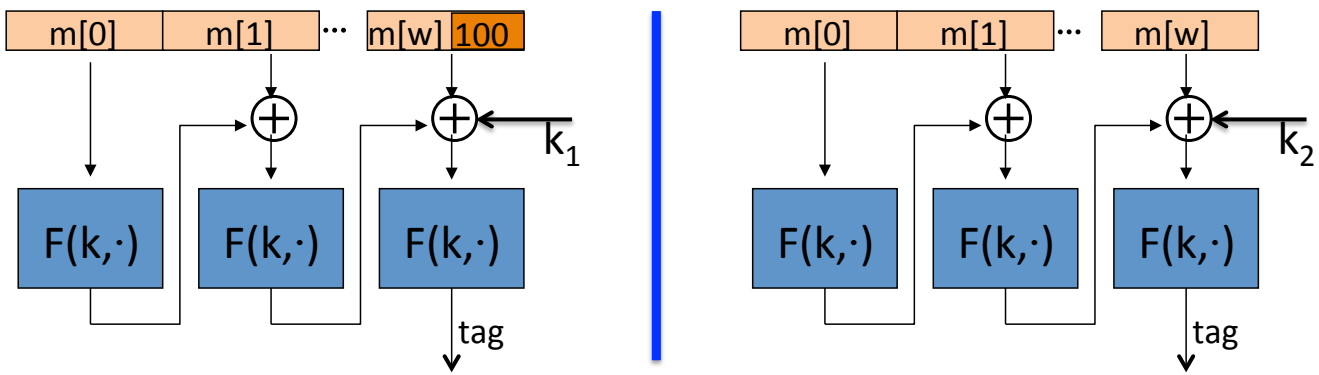
- Key (k, k_1)
- PRP $\mathbf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ used as often as needed (upper bound L)
- Not secure without final encryption step with key k_1!
- We constructed new (longer) PRF $\mathbf{F}_{\mathrm{ECBC}}\colon \mathcal{K} \times \mathcal{X}^{\leq L} \to \mathcal{X}$

Padding for CBC MACs

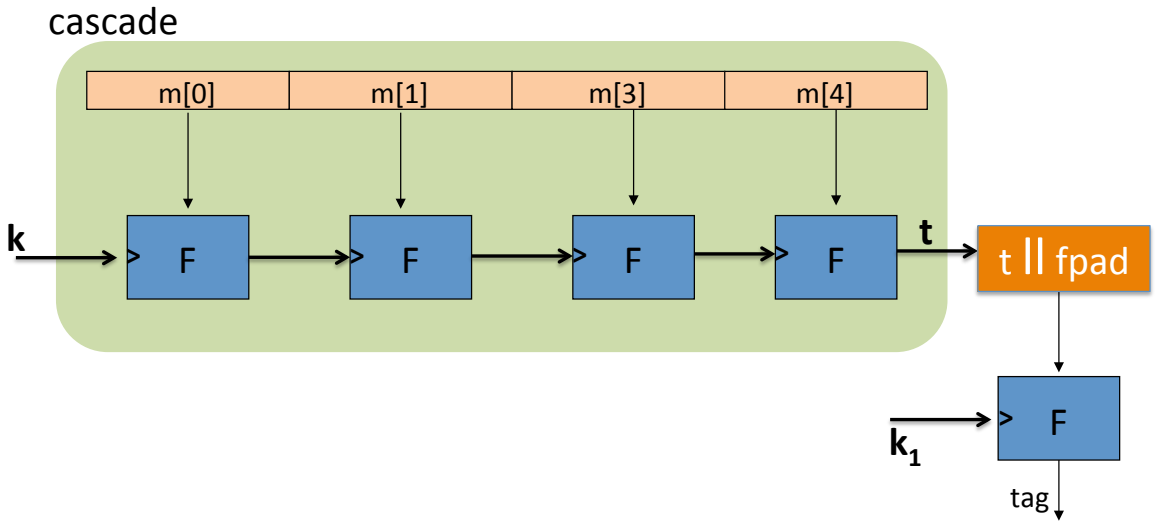
- Padding only with 0s allows for a simple single-message attack
- Instead pad as follows



- Alternatively, the CMA pads as follows (keys k_1 and k_2 are derived from k)



Encrypted NMAC



- Key (k, k_1)
- PRF $\mathbf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ used as often as needed (upper bound L)
- Not secure without final encryption step with key k_1!
- We constructed new (longer) PRF $\mathbf{F}_{\mathsf{NMAC}}\colon \mathcal{K} \times \mathcal{X}^{\leq L} \to \mathcal{K}$

Security Analysis of CBC MAC and NMAC

- Given PRP $\mathbf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ and PRF $\mathbf{F}'\colon \mathcal{K} \times \mathcal{X} \to \mathcal{K}$

Theorem

For any q-queries attacker $\mathbf{A}$ *on the PRF* $\mathbf{F}_{ECBC}$*, there exists an attacker* $\mathbf{B}$ *on the PRP* $\mathbf{F}$ *such that*

$$\mathrm{Adv}_{PRF}(\mathbf{A}, \mathbf{F}_{\mathrm{ECBC}}) \leq \mathrm{Adv}_{PRP}(\mathbf{B}, \mathbf{F}) + \frac{2q^2}{|\mathcal{X}|}$$

For any q-queries attacker $\mathbf{A}'$ *on the PRF* $\mathbf{F}_{NMAC}$*, there exists an attacker* $\mathbf{B}'$ *on the PRF* $\mathbf{F}'$ *such that*

$$\mathrm{Adv}_{PRF}(\mathbf{A}', \mathbf{F}_{\mathrm{NMAC}}) \leq qL \cdot \mathrm{Adv}_{PRF}(\mathbf{B}', \mathbf{F}') + \frac{q^2}{2|\mathcal{K}|}$$

- CBC MAC secure if $q \ll |\mathcal{X}|^{1/2}$
- NMAC secure if $q \ll |\mathcal{K}|^{1/2}$

Security Bounds are Tight: The Attack

- Consider a MAC $(\mathbf{S}, \mathbf{V})$ with the extension property:

$$\mathbf{S}(k, m_u) = \mathbf{S}(k, m_v) \qquad \Longrightarrow \mathbf{S}(k, m_u||w) = \mathbf{S}(k, m_v||w), \qquad \forall w$$

Any MAC $\mathbf{S}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ with this extension property can be forged with $q = |\mathcal{Y}|^{1/2} + 1$ queries:

- Query $q = |\mathcal{Y}|^{1/2}$ messages uniformly at random over $\mathcal{X}$;
- With very high probability at least two messages $m_u \neq m_v$ have same tags: $\mathbf{S}(k, m_u) = \mathbf{S}(k, m_v)$;
- Query $m_u||w$ for arbitrary w and obtain tag t;
- Forge the message-tag pair $(m_v||w, t)$

Birthday Paradox

Lemma

Let $X_1, \ldots, X_n$ be i.i.d. $\sim P_X$ over a space $\mathcal{X}$ and $n \geq 1.2|\mathcal{X}|^{1/2}$. Then

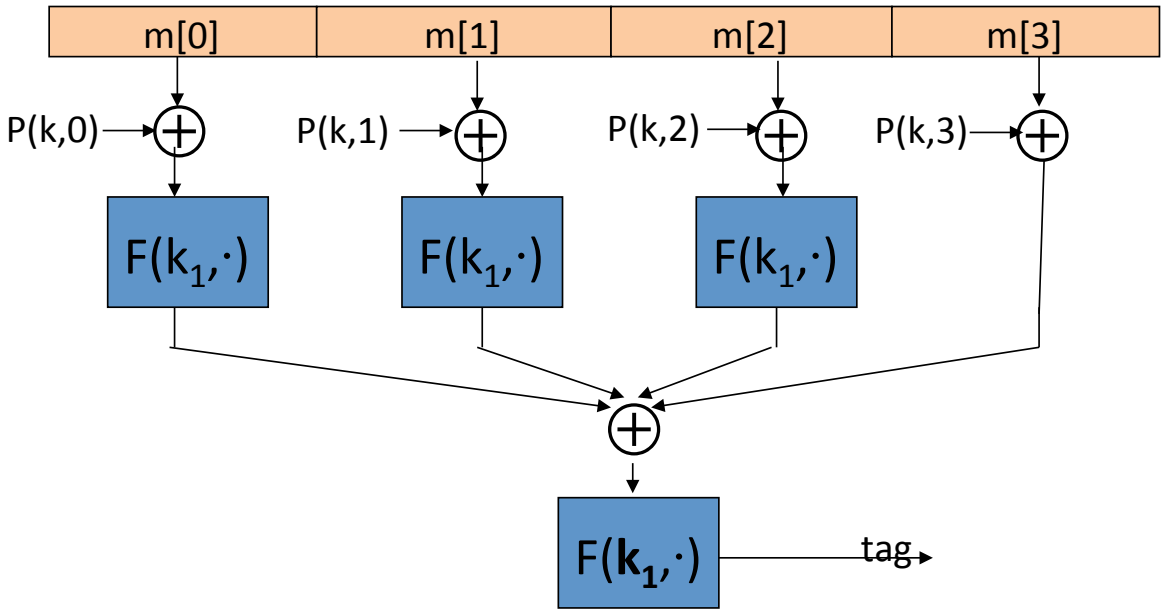
$$\Pr[\exists i \neq j \colon X_i = X_j] \geq 0.5.$$

Proof for P_X the uniform distribution over $\mathcal{X}$:

Hint: Use $(1 - x) \leq e^{-x}$

A (Long) Parallel MAC

- $P(k, i)$ easy to compute and PRF $\mathbf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{X}$
- key (k, k_1)



- Construct longer PRF $\mathbf{F}_{PMAC}\colon \mathcal{K}^2 \times \mathcal{X}^{\leq L} \to \mathcal{X}$ tagging at most L messages
- PMAC is incremental $\to$ can easily change $m[1]$ to $m'[1]$
- PMAC secure if constructed from a secure PRF $\mathbf{F}$ if $qL \ll |\mathcal{X}|^{1/2}$

A Secure One-Time MAC

- Consider a large prime number q and the random key $(a, b) \in \{1, \ldots, q\}^2$

$$\mathbf{S}(k = (a, b), m = [m(1), \ldots, m(L)]) = a^{L+1} + m[L]a^L + \cdots + m[1]a + b \mod q.$$

Theorem

This secure one-time MAC is information-theoretically secure because

$$\mathbb{P}[\mathbf{S}(k, m_1) = t_1 | \mathbf{S}(k, m_2) = t_2] \leq \mathbb{P}[\mathbf{S}(k, m_1) = t_1] \cdot L = \frac{L}{q}, \forall m_1, m_2, t_1, t_2.$$

Proof:

- This MAC is not many-times secure

Secure Many-Times MACs from Secure One-Time MACs

- A secure *one-time MAC* is secure against a single CMA ($q = 1$)
  $\rightarrow$ easier and faster to implement than MACs based on PRFs

- Let $(\mathbf{S}, \mathbf{V})$ be a secure one-time MAC with tag size $\{0,1\}^n$ and $\mathbf{F}$ a secure PRF onto $\{0,1\}^n$. Then, the following is a secure many-time MAC:

$$\mathbf{CW}((k_1, k_2), m) = (r, \mathbf{F}(k_1, r) \oplus \mathbf{S}(k_2, m)),$$

  where r is a randomness over $\{0,1\}^n$

Collision Resistant Hash Functions

- A hash function $h\colon \mathcal{M} \to \mathcal{T}$ maps a large domain $\mathcal{M}$ into a small tag space $\mathcal{T}$, i.e.,

$$|\mathcal{M}| \gg |\mathcal{T}|.$$

- A pair (m_0, m_1) with $m_0 \neq m_1$ and mapping to the same value under h, i.e., satisfying $h(m_0) = h(m_1)$, is called *a collision* for h.

Definition (Collision Resistance)

A hash function h is called *collision resistant* (C.R.), if the probability for any "efficient" algorithm **A** to produce a collision

$$\mathrm{Adv}_{CR}(\mathbf{A}, h) := \mathbb{P}[\mathbf{A} = (m_0, m_1) \quad \text{and} \quad (m_0, m_1) \text{ is a collision on } h]$$

is negligible

- Example: SHA-256 (256 output bits)

Obtaining a Large MAC from a C.R. Hash Function

- Let $h\colon \mathcal{M} \to \mathcal{T}'$ be a collision resistant hash function
- Let $(\mathbf{S}, \mathbf{V})$ be a MAC for short messages, $\mathbf{S}\colon \mathcal{K} \times \mathcal{T}' \to \mathcal{T}$

Theorem

The large MAC $(\mathbf{S}_{\text{big}}, \mathbf{V}_{\text{big}})$ *given by*

$$\begin{aligned} \mathbf{S}_{\text{big}}(k, m) &= \mathbf{S}(k, h(m)) \\ \mathbf{V}_{\text{big}}(k, m, t) &= \mathbf{V}(k, h(m), t) \end{aligned}$$

also is a secure MAC.

- Example: Combine AES in 2-block CBC mode with SHA-256.
- Certainly if h is not collision resistant the construction is not secure

Protecting Files with a C.R. Hash Function

- Simply publish a hash of all the files at their generation onto a public read-only space
- A user can detect changes if the hash function is collision resistant
- Don't need a key but a publicly available read-only space that cannot be tempered

Generic Attack on C.R. Hash Functions

Consider a hash function $h\colon \mathcal{M} \to \mathcal{T}$

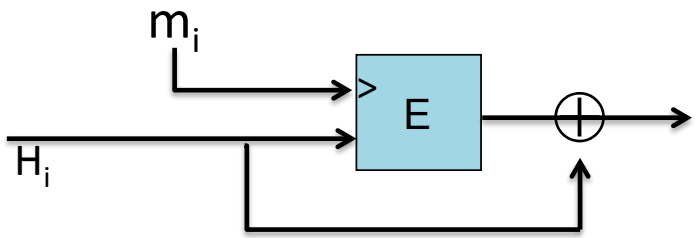
1. Evaluate the function h on $q = |\mathcal{T}|^{1/2}$ messages uniformly at random over $\mathcal{M}$;
2. If a collision has occurred, produce this collision, otherwise repeat step 1.

Expected number of iterations until success is only 2
$\rightarrow$ see the birthday paradox

Compression Functions from Block Ciphers: Davies-Meyer Construction

- Let $\mathbf{E}\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher.
- Compression function for the Merkle-Damgard construction:

$$h(H, m) = \mathbf{E}(m, H) \oplus H.$$



Theorem

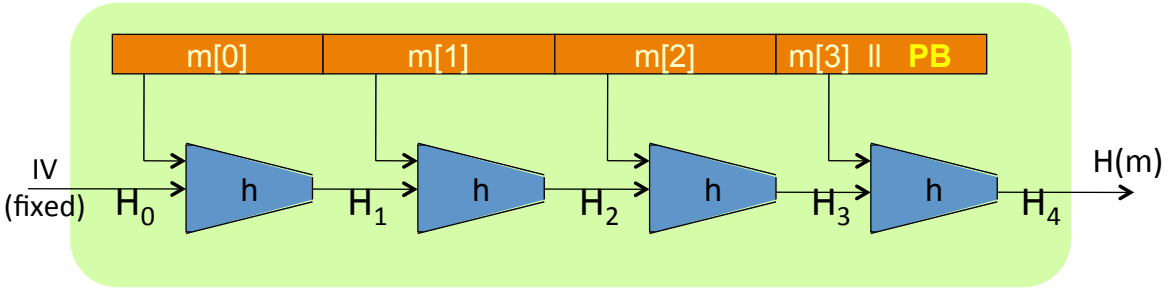
If $\mathbf{E}(m, \cdot)$ is a purely random permutation it takes $O(2^{n/2})$ evaluations of $\mathbf{E}$ to find a collision for the Davies-Meyer construction.

Merkle-Damgard Iterated Construction to Obtain a Hash Function

Goal: Construct a long hash function

$$H\colon \mathcal{X}^{\leq L} \to \mathcal{T}$$

from a small hash function $h\colon \mathcal{T} \times \mathcal{X} \to \mathcal{T}$



- Add padding block to the message (10...00||msg_len in 64 bits) $\rightarrow$ add an extra block if necessary

Theorem

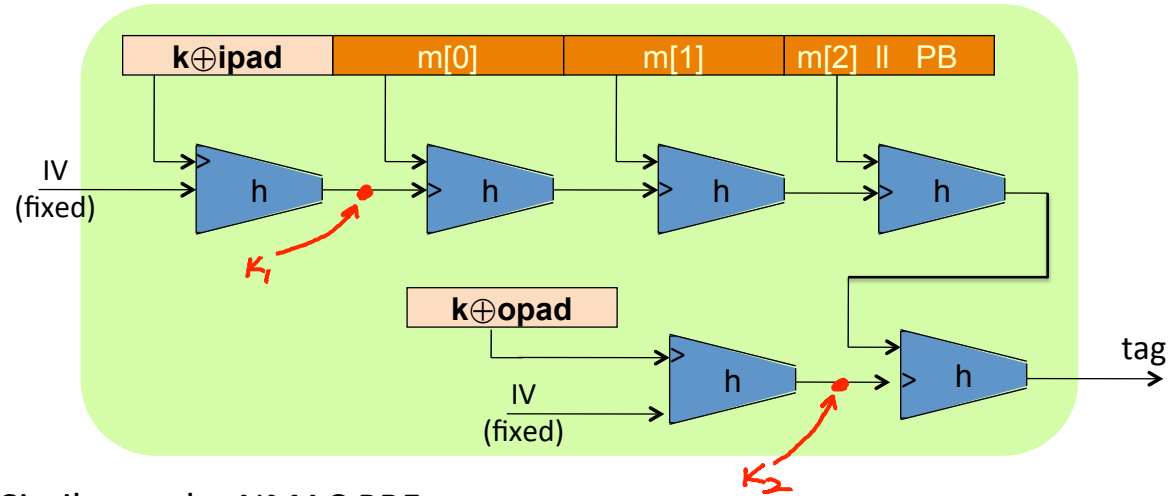
If h is collision resistant, so is the Merkle-Damgard construction H.

- However: MAC $S(k, m) = H(k\|m)$ is *not* a secure MAC

SHA-256

- Merkle-Damgard construction
- Davies-Meyer compression function but where XOR is replaced by addition modulo 2^{32}
- Used block cipher is SHACAL-2 with 512-bit key and 256-bit input/output

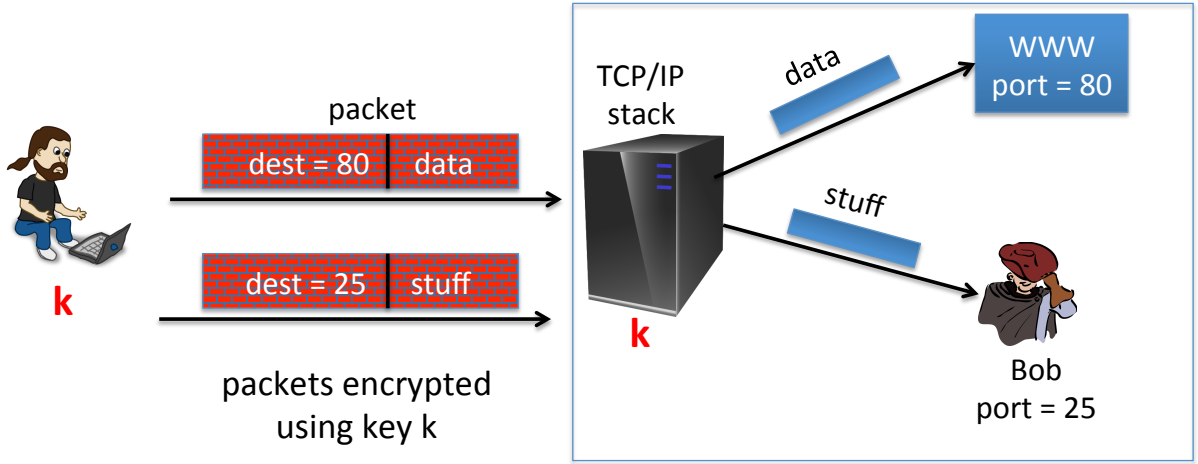
HMAC



- $\mathbf{S}_{\mathrm{HMAC}}(k, m) = H(k \oplus \mathrm{opad} || H(k \oplus \mathrm{ipad} || m))$
- opad and ipad are 512 bits long fixed values; IV is a fixed 256 bit value
- Similar to NMAC PRF with dependent keys k_1, k_2
  $\rightarrow$ HMAC is a secure PRF under some assumptions on h when $\frac{q^2}{|\mathcal{T}|^{1/2}}$ is negligible

Authenticated Encryption

An Active Attack on CBC Encryption



- Exchange CBC initial value IV with $IV' =$

Authenticated Encryption

$$\begin{aligned} \mathbf{E}: &\quad \mathcal{K} \times \mathcal{M} \times \mathcal{N} \rightarrow \mathcal{C} \\ \mathbf{D}: &\quad \mathcal{K} \times \mathcal{C} \times \mathcal{N} \rightarrow \mathcal{M} \cup \{\perp\} \end{aligned}$$

Definition (Authenticated Encryption (AE))

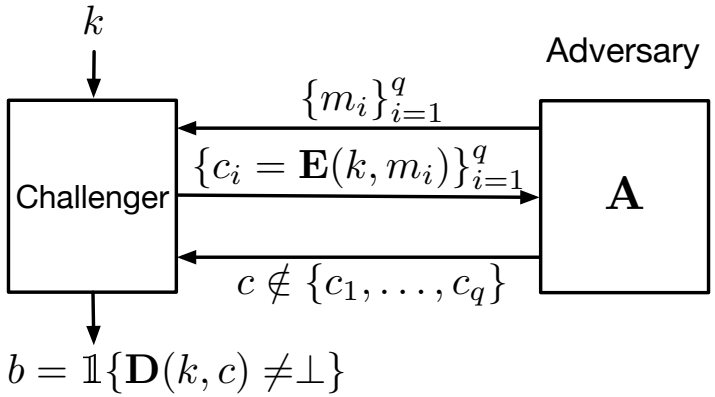
Cipher $(\mathbf{E}, \mathbf{D})$ provides *authenticated encryption (AE)* if it is

- semantically secure under multi-CPA; and
- has ciphertext integrity

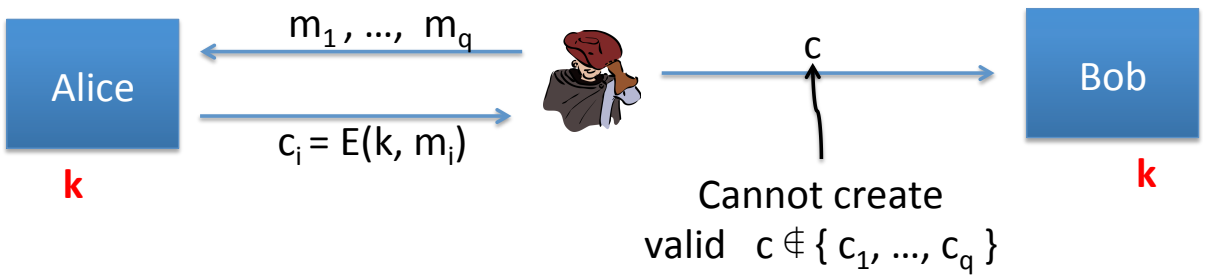
Definition (Ciphertext Integrity (CI))

Cipher $(\mathbf{E}, \mathbf{D})$ provides *ciphertext integrity*, if for all "efficient" $\mathbf{A}$:

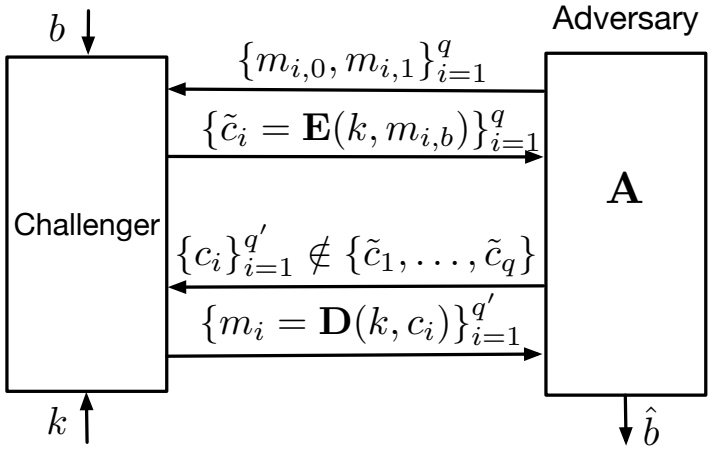
$$\mathsf{Adv}(\mathbf{A}, \mathbf{E}) = \mathbb{P}[b = 1] \qquad \text{is negligible}$$



Authenticated Encryption Implies Authenticity



Chosen Ciphertext Security (CCA)



Definition (Security against Chosen Ciphertext Attacks (CCA))

Cipher $(\mathbf{E}, \mathbf{D})$ provides *security under chosen ciphertext attacks*, if for all "efficient" $\mathbf{A}$:

$$\mathrm{Adv}(\mathbf{A}, \mathbf{E}) = |\mathbb{P}[\mathbf{A} = 1 | b = 1] - \mathbb{P}[\mathbf{A} = 1 | b = 0]| \qquad \text{is negligible}$$

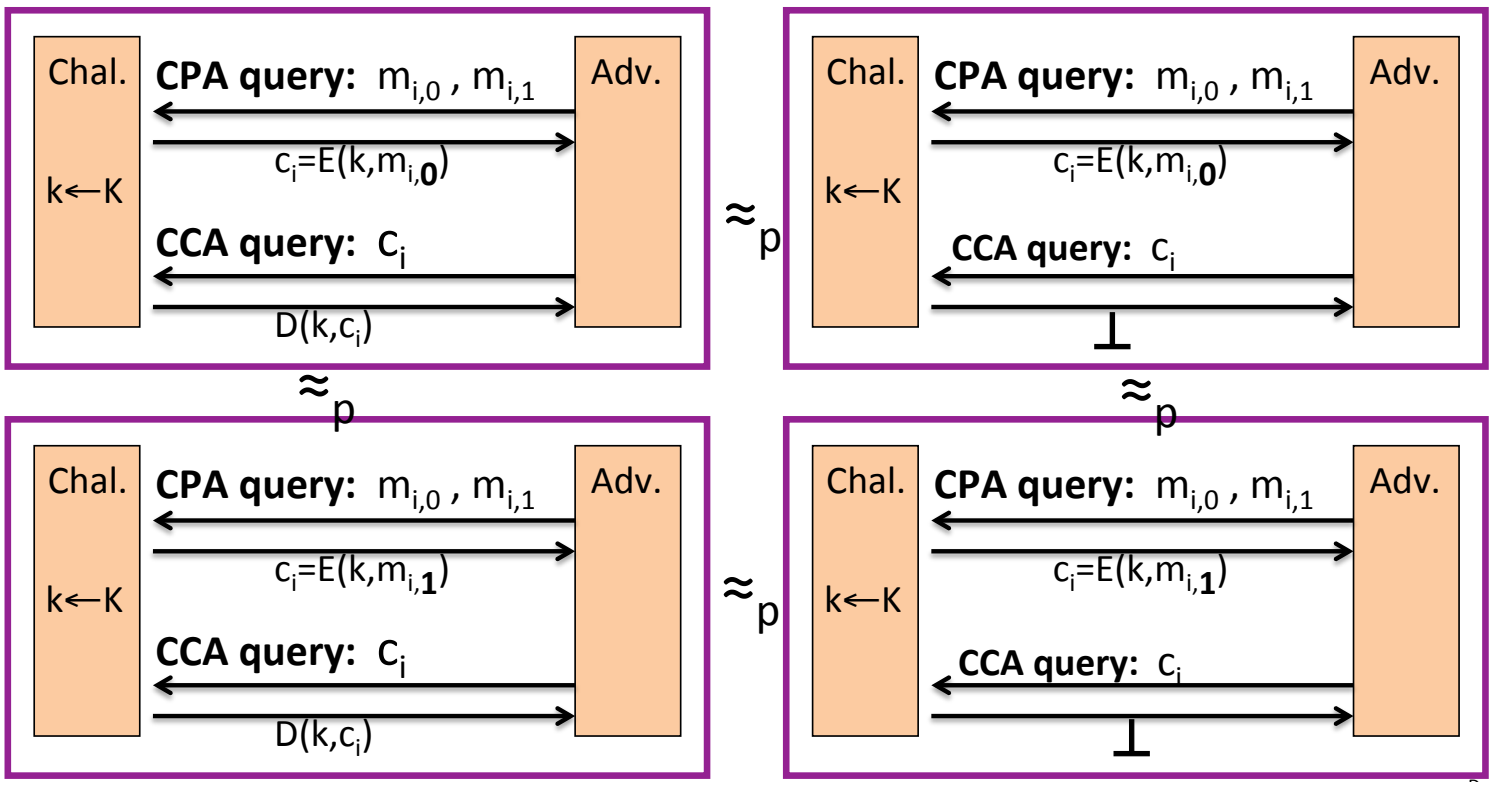
- CBC with random IV is not CCA secure

Authenticated Encryption Implies Chosen Ciphertext Security

Theorem

If a cipher $(\mathbf{E}, \mathbf{D})$ *provides authenticated encryption, then it also implies security against chosen ciphertext attacks (CCA).*

Proof: Based on CI-Adv and CPA-Adv



Encrypt-Then-MAC provides AE

- Let $(\mathbf{E}, \mathbf{D})$ be a CPA secure cipher and $(\mathbf{S}, \mathbf{V})$ a secure MAC
- Independent key pair (k_E, k_M)

Theorem

$\mathbf{S}(k_M, \mathbf{E}(k_E, m))$ *provides authenticated encryption.*

Standards for Authenticated Encryption

GCM: Apply first a CTR mode encryption then the CW-MAC

CCM: Apply the CBC-MAC then CTR mode encryption

EAX: CTR mode encryption then CMAC

OCB from a PRP