05 Hash, MAC, Signature

2020 Spring

Information Security

Teacher: Po-Wen Chi

neokent@gapps.ntnu.edu.tw

March 24, 2020

Department of Computer Science and Information Engineering,

National Taiwan Normal University

What We Have Learned?

- **Confidentiality**.
- **Integrity**.
- Availability.
- Authenticity.
- Accountability.

Now we will introduce some techniques about other factors.

What We Have Learned?

- **Confidentiality**.
- **Integrity**.
- Availability.
- Authenticity.
- Accountability.

Now we will introduce some techniques about other factors.

Wait a moment! I think public key system can be treated as an authentication way!

What We Have Learned?

- **Confidentiality**.
- **Integrity**.
- Availability.
- Authenticity.
- Accountability.

Now we will introduce some techniques about other factors.

Wait a moment! I think public key system can be treated as an authentication way!

Not good enough. Where do you get other's public key?

Message Integrity

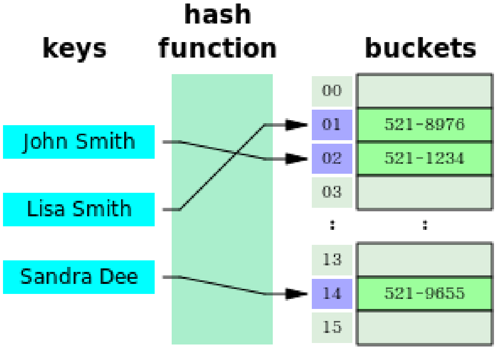
- Goal: **Integrity**, not confidentiality.

So in practice, all techniques you learned here should be used combined with what you learned before.

Cryptographic Hash

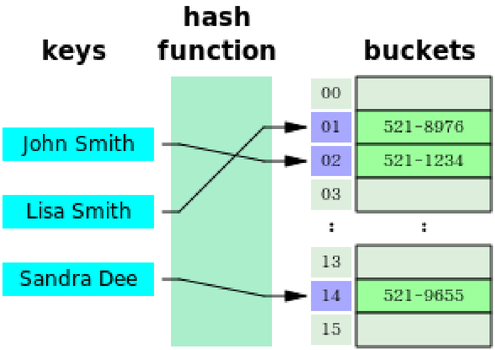
What is Hash?

- 雜湊函式、散列函式、哈希函式
- Undoubtedly, you have learned it from Data Structure.



What is Hash?

- 雜湊函式、散列函式、哈希函式
- Undoubtedly, you have learned it from Data Structure.
- Why do you need hash functions?



Cryptographic Hash Requirement

1. Variable input size.
2. Fixed output size.
3. Efficiency.
4. One-way property.
5. **Collision resistant**.
   - **Weak**: For any given x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
   - **Strong**: It is computationally infeasible to find any pair (x, y) that $y \neq x$ and $H(y) = H(x)$.
6. Pseudorandomness.

Quiz

Collision is inevitable.

Why?

Quiz

In this class, are there any two who have the same birthday?

Quiz

In this class, are there any two who have the same birthday?

Suppose there are n students here. The probability that every has an unique birthday is

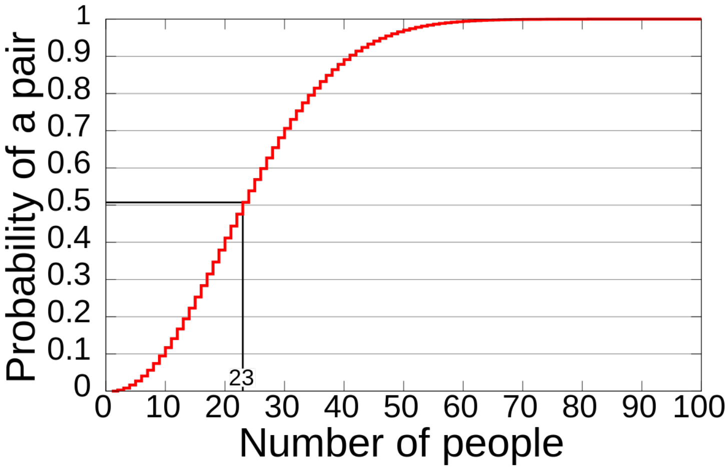
Quiz

In this class, are there any two who have the same birthday?

Suppose there are n students here. The probability that every has an unique birthday is

$$P = 1 - \bar{P} = 1 - \prod_{i=0}^{n-1} \frac{365 - i}{365}$$

Birthday Attack



Birthday Attack

- This implies to find a Hash collision, you do not need to do 2^n searches. For most cases, $2^{\frac{n}{2}}$ searches are enough for you to find a collision.

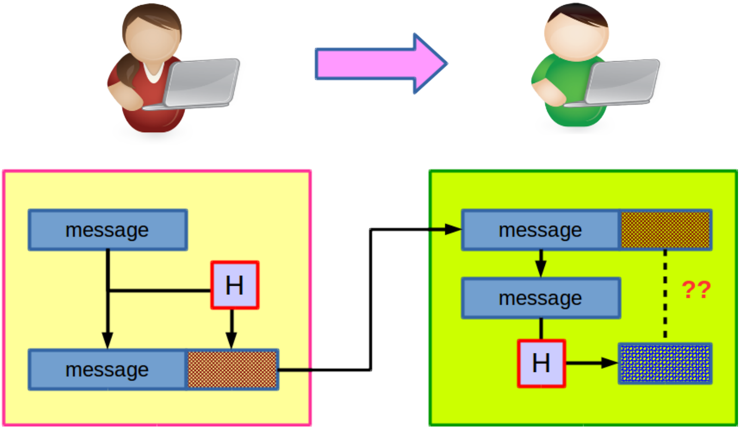
Birthday Attack

- This implies to find a Hash collision, you do not need to do 2^n searches. For most cases, $2^{\frac{n}{2}}$ searches are enough for you to find a collision.
- How to solve this problem?

Birthday Attack

- This implies to find a Hash collision, you do not need to do 2^n searches. For most cases, $2^{\frac{n}{2}}$ searches are enough for you to find a collision.
- How to solve this problem?
- Make n larger!

Hash Function Basic Usage



Integrity Check in the Real World

Name	Last modified	Size	Description
Parent Directory		-	
MD5SUMS	2019-02-14 22:53	138	
MD5SUMS-metalink	2019-02-14 22:53	148	
MD5SUMS-metalink.gpg	2019-02-14 22:53	916	
MD5SUMS.gpg	2019-02-14 22:53	916	
SHA1SUMS	2019-02-14 22:53	154	
SHA1SUMS.gpg	2019-02-14 22:53	916	
SHA256SUMS	2019-02-14 22:53	202	
SHA256SUMS.gpg	2019-02-14 22:53	916	
ubuntu-18.04.2-desktop-amd64.iso	2019-02-10 00:27	1.9G	Desktop image for 64-bit PC (AMD64) computers (standard download)
ubuntu-18.04.2-desktop-amd64.iso.torrent	2019-02-14 22:51	75K	Desktop image for 64-bit PC (AMD64) computers (BitTorrent download)
ubuntu-18.04.2-desktop-amd64.iso.zsync	2019-02-14 22:51	3.7M	Desktop image for 64-bit PC (AMD64) computers (zsync metafile)
ubuntu-18.04.2-desktop-amd64.list	2019-02-10 00:27	7.8K	Desktop image for 64-bit PC (AMD64) computers (file listing)
ubuntu-18.04.2-desktop-amd64.manifest	2019-02-10 00:25	57K	Desktop image for 64-bit PC (AMD64) computers (contents of live filesystem)
ubuntu-18.04.2-desktop-amd64.metalink	2019-02-14 22:53	47K	Ubuntu 18.04.2 LTS (Bionic Beaver)

Another Case: Linux Shadow

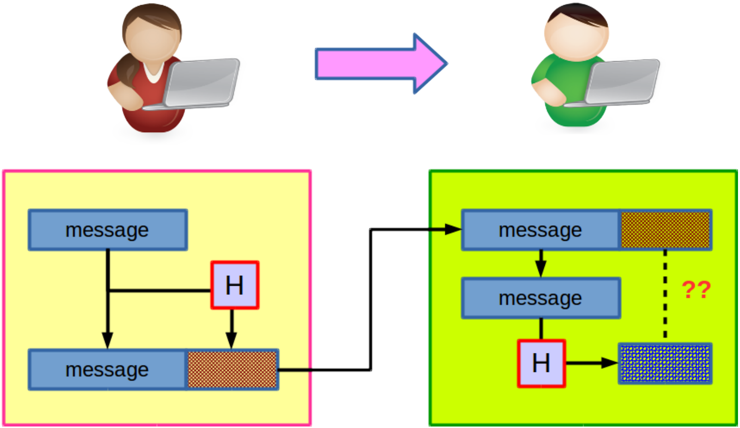
What is the difference between /etc/passwd and /etc/shadow in Linux?

Another Case: Linux Shadow

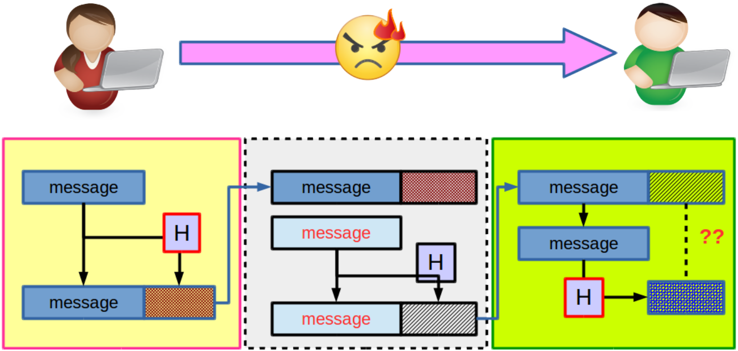
What is the difference between /etc/passwd and /etc/shadow in Linux?

Never store the passwords in the plaintext way.

Integrity Check is Not Enough



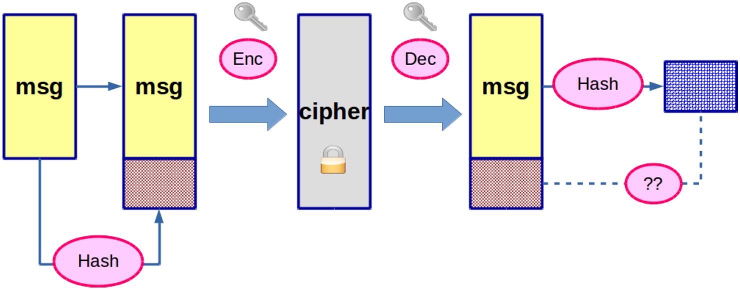
Integrity Check is Not Enough



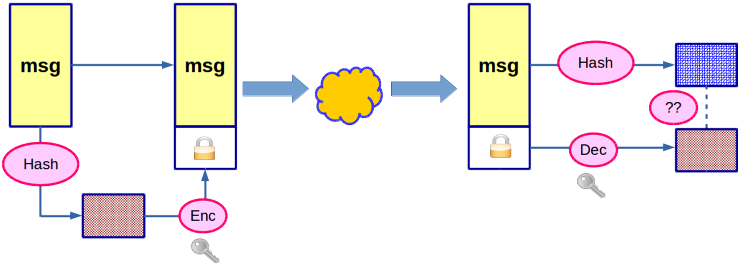
Note

- So in most cases, we will not use the hash function in that way.
- Wait! You just show some real cases, right?

Usage Example: 1/4

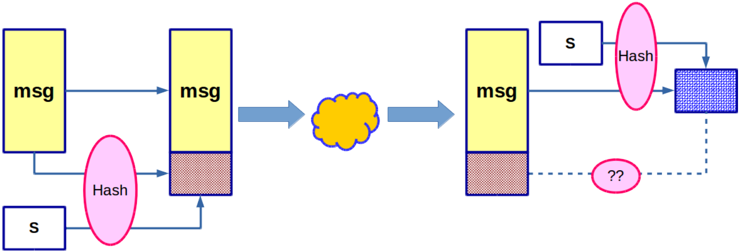


$$E(K, M||H(M))$$



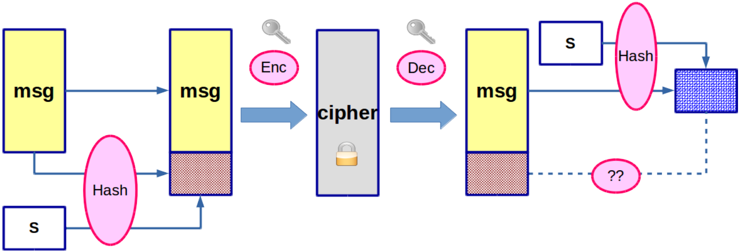
$$M||E(K, H(M))$$

Usage Example: 3/4



$$M || H(M || S)$$

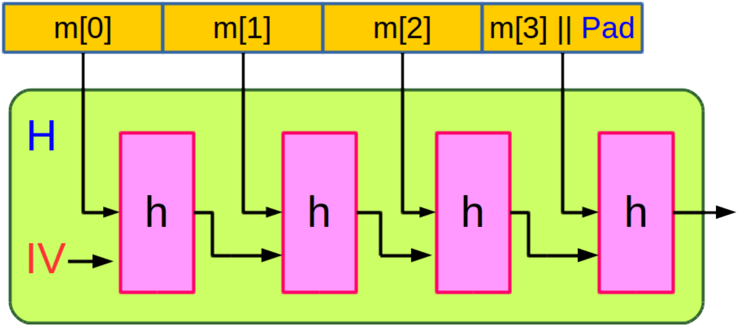
Usage Example: 4/4



$$E(K, M||H(M||S))$$

How to Make a Cryptographic HASH Function?

Merkle-Damgard Paradigm



Given $h : T \times X \to T$(compressible function), we obtain $H : X^L \to T$.

Theorem

If h is collision resistant, H is also collision resistant.

How to prove?

Collision Resistance

Theorem

If h is collision resistant, H is also collision resistant.

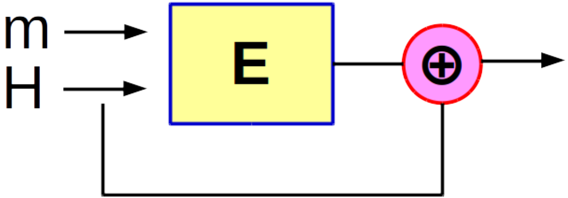
How to prove?

- Suppose $H(M) = H(M')$ and $M \neq M'$, we can find a collision pair for h.

Great! But How to Build h?

- We can build h by a block cipher.
- The Davies-Meyer compression function:

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



How to prove its collision resistance?

Quiz

If we define $h(m, H) = E(m, H)$, is h collision resistant?

Quiz

If we define $h(m, H) = E(m, H)$, is h collision resistant?

No! Why?

Quiz

If we define $h(m, H) = E(m, H)$, is h collision resistant?

No! Why?

Given H, m, m', $H' = D(m', E(m, H))$.

Other Example

- Miyaguchi-Preneel:

$$h(m, H) = E(m, H) \oplus H \oplus m$$

- Actually, there are 12 styles of this.

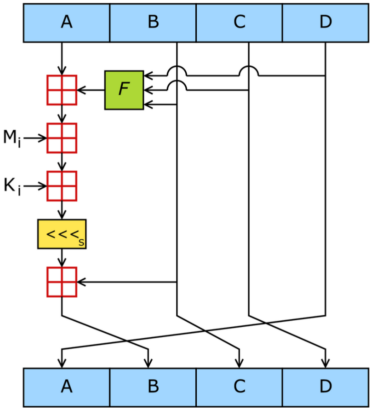
Common Hash Functions

Common Hash Functions

- MD5.
- SHA1.
- SHA256.
- SHA512.

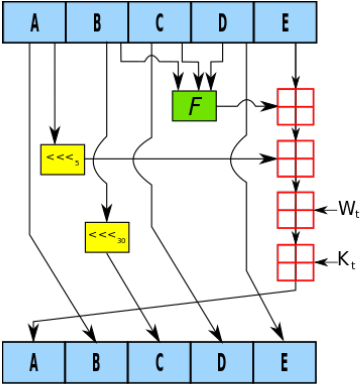
MD5

- Output: 128 bits.
- Widely used before.
- Collisions for the full MD5 were announced by Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu.
- It can still be used as a checksum to verify data integrity, but only against unintentional corruption.



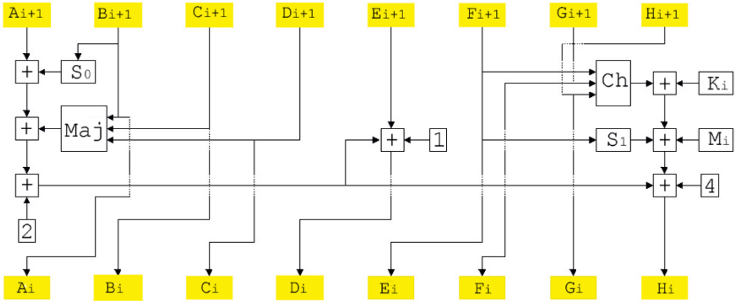
SHA-1

- Output 160 bits.
- On 23 February 2017, the CWI (Centrum Wiskunde & Informatica) and Google announced the SHAttered attack, in which they generated two different PDF files with the same SHA-1 hash in roughly 263.1 SHA-1 evaluations.
- `https://shattered.io/` .



SHA256

- Output: 256bits.
- Merkle-Damgard function.
- Davies-Meyer compression function.
- Block cipher: SHACAL-2.



Wait a Moment!

Have you noticed that I do not introduce the compression function in detail?

Provable Compression Function

- The compression function above can be replaced with some other provable compression functions.
- Example:

$$h(m, H) = u^H v^m \bmod p.$$

Provable Compression Function

- The compression function above can be replaced with some other provable compression functions.
- Example:

$$h(m, H) = u^H v^m \bmod p.$$

  - How to prove this is a secure compression function?

Provable Compression Function

- The compression function above can be replaced with some other provable compression functions.
- Example:

$$h(m, H) = u^H v^m \bmod p.$$

  - How to prove this is a secure compression function?
  - In fact, we do not use this. Why?

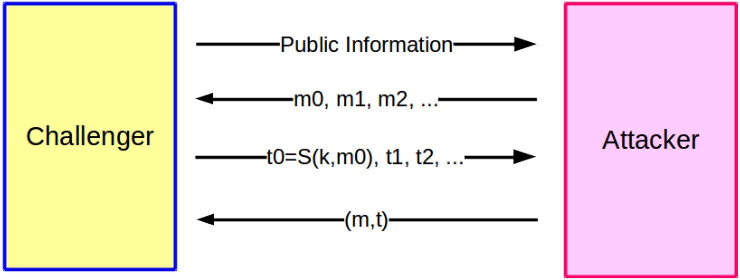
MAC: Message Authentication Code

MAC Requirements

- Message authentication is a procedure to verify that received messages come from the **alleged source** and have not been modified.
- We also called MAC is a **key-ed hash**.

Secure MAC

For a MAC $I = (S, V)$ where S is **Sign** algorithm and V is **Verify** algorithm, the security model is



The attacker wins the game if (m, t) is valid and $m \neq m_0, m_1, m_2, \ldots$.

A MAC $I = (S, V)$ is called secure if for all efficient A, $\mathrm{Adv}_{\mathrm{MAC}}[A, I]$ is negligible.

Quiz

Let $I = (S, V)$ be a MAC.

Suppose an attacker can find $m_0 \neq m_1$ such that $S(k, m_0) = S(k, m_1)$ for $\frac{1}{2}$ of the keys.

Is $I = (S, V)$ a secure MAC?

Quiz

Let $I = (S, V)$ be a MAC.

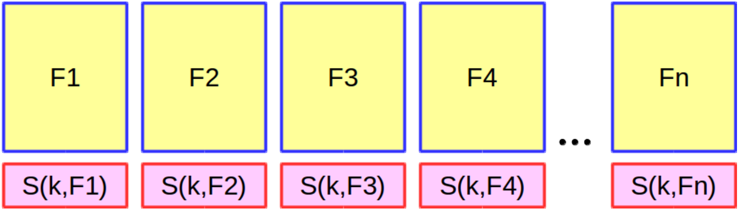
Suppose $S(k, m)$ is always 5-bits long.

Is $I = (S, V)$ a secure MAC?

Example: System File Protection

How can they catch the cheating event?

Example: System File Protection



How to Build a MAC

How to Build a MAC?

- Based on PRF.
- CBC-MAC.
- NMAC.
- PMAC.
- HMAC (Hash-based MAC).

For a PRF $F: K \times X \to Y$, define a MAC $I = (S, V)$ as follows:

$$S(k, m) = F(k, m),$$

$$V(k, m, t) = \begin{cases} 1, & \text{if } t = F(k, m) \\ 0, & \text{otherwise} \end{cases}.$$

Is this a secure MAC?

Bad Example

Suppose $F: K \times X \to Y$ is a secure PRF with $Y = \{0,1\}^{10}$.

Is the derived MAC a secure MAC?

Security MAC based on PRF

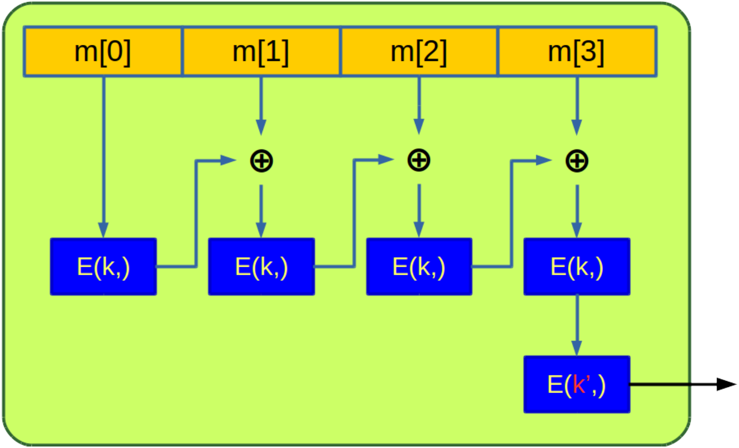
Theorem

If $F : K \times X \to Y$ is a secure PRF and $\frac{1}{|Y|}$ is negligible, then I_F is a secure MAC.

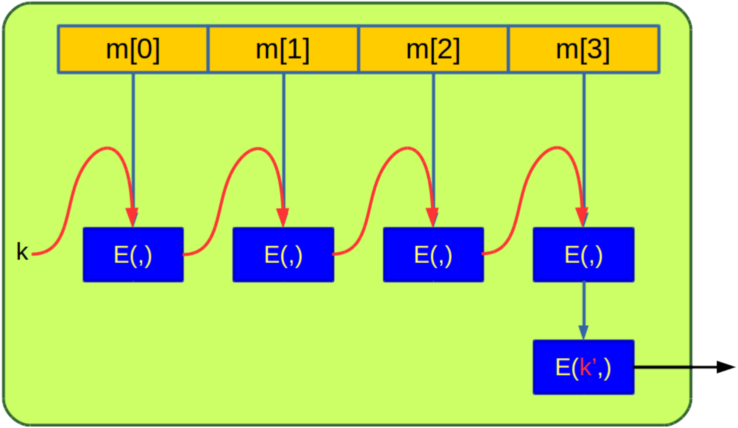
$$\mathrm{Adv}_{\mathrm{MAC}}[A, I_F] \leq \mathrm{Adv}_{\mathrm{PRF}}[B, F] + \frac{1}{|Y|}.$$

CMAC

$$F_{\text{CMAC}} : K^2 \times X^L \to X.$$



$$F_{\mathrm{NMAC}} : K^2 \times X^L \to K.$$



Quiz

Why do we need the last stage encryption?

Quiz

Why do we need the last stage encryption?

Cascade problem.

Quiz

Why do we need the last stage encryption?

Cascade problem.

Wait a moment! I know that NMAC has the cascade problem.
How about CMAC?? The attacker do not have k, right?

Quiz

Why do we need the last stage encryption?

Cascade problem.

Wait a moment! I know that NMAC has the cascade problem.
How about CMAC?? The attacker do not have k, right?

Good question! But that will be your homework.

Note

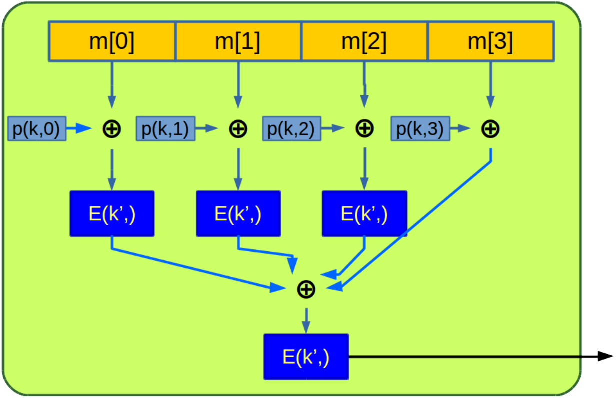
- **ECBC-MAC** is commonly used as an AES-based MAC.
  - CCM encryption mode (used in 802.11i).
  - NIST standard called CMAC.
- **NMAC** is not usually used with AES or 3DES.
  - Why?
  - But NMAC is the basis for a popular MAC called **HMAC**.
- Both of them are sequential, is there any parallel one?

Note

- **ECBC-MAC** is commonly used as an AES-based MAC.
  - CCM encryption mode (used in 802.11i).
  - NIST standard called CMAC.
- **NMAC** is not usually used with AES or 3DES.
  - Why? Need to change AES key on every block.
  - But NMAC is the basis for a popular MAC called **HMAC**.
- Both of them are sequential, is there any parallel one?

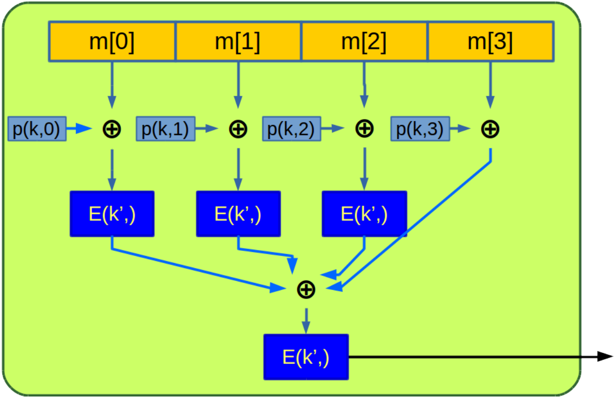
PMAC: Parallelizable MAC

$$F_{\text{CMAC}} : K^2 \times X^L \to X.$$



PMAC: Parallelizable MAC

$$F_{\text{CMAC}} : K^2 \times X^L \to X.$$



If we modify m[1] to m[1]', how can we update the MAC value?

HMAC

- We can also use hash functions to build MAC.
- Of course, you should not use MD5 or SHA-1.

HMAC

- We can also use hash functions to build MAC.
- Of course, you should not use MD5 or SHA-1.
- **Quiz**:
  - How about the following construction?

$$S(k, m) = H(k||m).$$

  - Is this secure or not?

HMAC

- We can also use hash functions to build MAC.
- Of course, you should not use MD5 or SHA-1.
- **Quiz**:
  - How about the following construction?

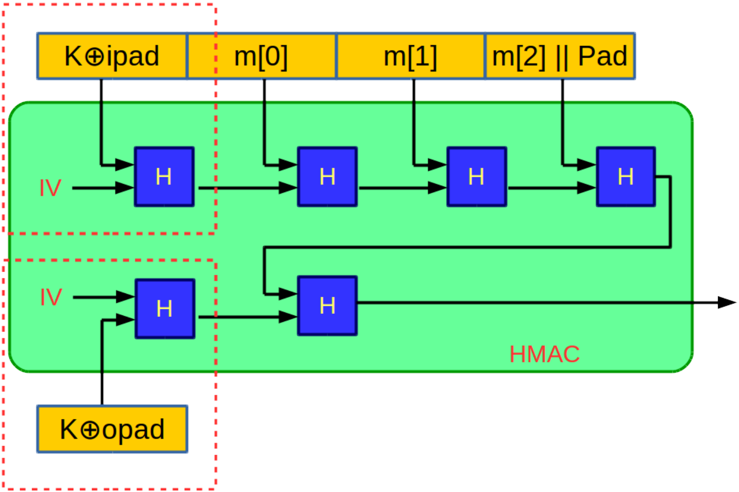
$$S(k, m) = H(k||m).$$

  - Is this secure or not?
  - No. Because of Merkle-Damgard iterated construction.

HMAC

- Build from a black-box Hash function.
  - This implies that you can replace the black-box with what you want.
- HMAC(K, M) : $H(H(K \oplus \text{opad})||H((K \oplus \text{ipad})||M))$.
  - opad: 0x5c5c5c...5c5c
  - ipad: 0x363636...3636
- TLS must supports HMAC-SHA1-96.

HMAC



The red blocks can be pre-computed.

MAC Padding Issue

- How about padding all zeros?
    - Bad idea, why?

MAC Padding Issue

- How about padding all zeros?
  - Bad idea, why?
- ISO: Pad with "100...0", add a dummy block if necessary.

Digital Signature

Digital Signature

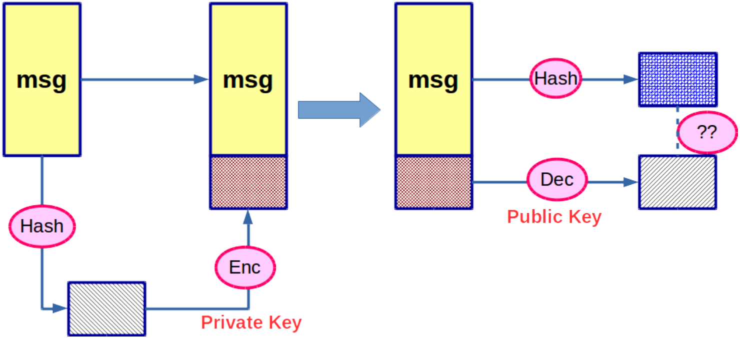
Digital Signature $=$ HASH $+$ Authentication

- **Hash**: Msg content is correct, which means it is the same with the one when the sender send it.
- **Authentication**: The message is really from the one who claims that he/she sends it.

Note

- Digital Signature has nothing to do with confidentiality.
- So in most case, we will use it with encryption.

RSA Signature



Blind Signature

- Can you sign a document when you do not know its content?
- Why do I need this feature?
  - Think about **eVoting**.
- Blind RSA Signature:
  - Given a message m and you want to ask others to sign this message without knowing its content, how to do this?

For DSA

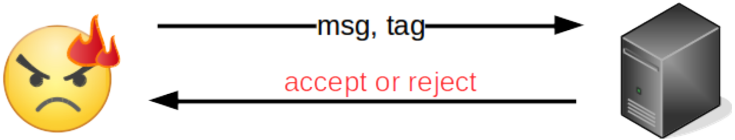
- I will not cover this topic here.
- To introduce DSA, I will introduce another technique first $\cdots$
  So please wait.

Appendix

Note

Generally speaking, Hash is faster than Symmetric Key Encryption and Symmetric Key Encryption is faster than Asymmetric Key Encryption.

Timing Attack



To compute a tag for a target message:

1. Query the server with random tag.
2. Loop over all possible first bytes and query the server. Stop when verification takes a little longer.
3. Repeat until all bytes are found.

Defense

Make the string comparator always takes the same time.

Make the string comparator always takes the same time.

Unfortunately, sometimes compiler may optimize this part.

Do Not Implement Crypto Yourself!