Message Authentication Codes and Collision-Resistant Hash Functions

Yu Zhang

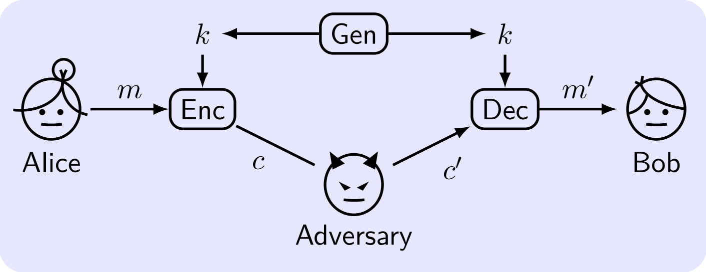
Harbin Institute of Technology

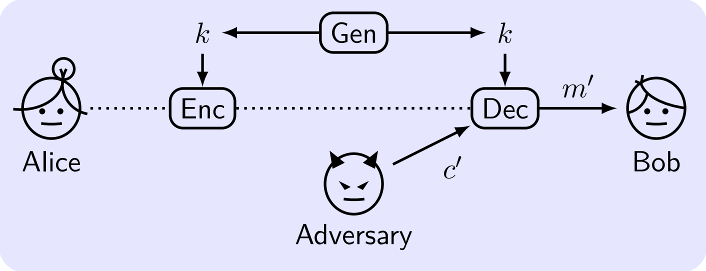
Cryptography, Autumn, 2022

Outline

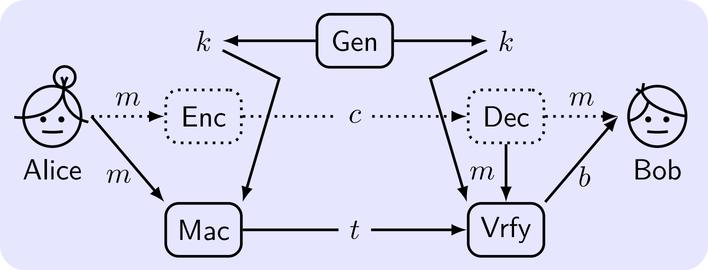
Content

Integrity and Authentication





The Syntax of MAC



- key k, tag t, a bit b means valid if $b = 1$; invalid if $b = 0$.
- **Key-generation** algorithm $k \leftarrow \mathsf{Gen}(1^n), |k| \geq n$.
- **Tag-generation** algorithm $t \leftarrow \mathsf{Mac}_k(m)$.
- **Verification** algorithm $b := \mathsf{Vrfy}_k(m, t)$.
- **Message authentication code**: $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$.
- **Basic correctness requirement**: $\mathsf{Vrfy}_k(m, \mathsf{Mac}_k(m)) = 1$.

Security of MAC

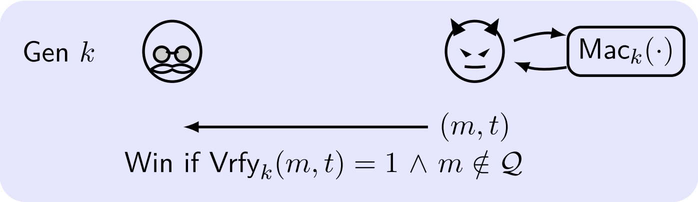
- **Intuition**: No adversary should be able to generate a **valid** tag on any "**new**" message[1] that was not previously sent.
- **Replay attack**: Copy a message and tag previously sent. (**excluded by only considering "new" message**)
  - Sequence numbers: receiver must store the previous ones.
  - Time-Stamps: sender/receiver maintain synchronized clocks.
- **Existential unforgeability**: **Not** be able to forge a valid tag on **any** message.
  - **Existential forgery**: *at least one* message.
  - **Selective forgery**: message chosen *prior* to the attack.
  - **Universal forgery**: *any* given message.
- **Adaptive chosen-message attack (CMA)**: be able to obtain tags on *any* message chosen adaptively *during* its attack.

[1] A stronger requirement is concerning *new message/tag pair*.

Definition of MAC Security

The message authentication experiment $\mathsf{Macforge}_{\mathcal{A},\Pi}(n)$:

1. $k \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given input 1^n and oracle access to $\mathsf{Mac}_k(\cdot)$, and outputs (m, t). $\mathcal{Q}$ is the set of queries to its oracle.
3. $\mathsf{Macforge}_{\mathcal{A},\Pi}(n) = 1 \iff \mathsf{Vrfy}_k(m, t) = 1 \wedge m \notin \mathcal{Q}$.



Definition 1

A MAC Π is **existentially unforgeable under an adaptive CMA** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that: $\Pr[\mathsf{Macforge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n)$.

Real World Case

The 802.11b Insecure MAC[2]

Consider a variant of WiFi encryption in 802.11b WEP (Wired Equivalent Privacy). Let F be a PRF with a 32-bit length output. Let CRC32 be an error-detecting code outputting a 32-bit string. Define the following MAC scheme:

$$S(k,m) := (r \leftarrow \{0,1\}^n, t \leftarrow F(k,r) \oplus \mathsf{CRC32}(m))$$

$$V(k,m,(r,t)) := 1 \quad \text{if} \quad t = F(k,r) \oplus \mathsf{CRC32}(m)$$

- Different messages may have the same CRC32 output.
- Attacker can learn $F(k,r)$ from a valid tag, and then output $(m', (r, F(k,r) \oplus \mathsf{CRC32}(m')))$.

[2]from BonehShoup v0.5 p.234

Questions

Suppose $\langle S, V\rangle$ are CMA-secure, are $\langle S', V'\rangle$ secure?

- $S'_k(m) = (S_k(m), m)$, $V'_k(m, (t_1, t_2)) = V_k(m, t_1) \wedge t_2 = m$
- $S'_{k_1,k_2}(m) = (S_{k1}(m), S_{k_2}(m))$
  $V'_{k_1,k_2}(m, (t_1, t_2)) = V_{k1}(m, t_1) \wedge V_{k_2}(m, t_2)$
- $S'_k(m) = (S_k(m), S_k(m))$
  $$V'_k(m, (t_1, t_2)) = \begin{cases} V_k(m, t_1) & \text{if } t_1 = t_2 \\ 0 & \text{otherwise} \end{cases}$$
- $S'_k(m) = (S_k(m), S_k(0^n))$
  $V'_k(m, (t_1, t_2)) = V_k(m, t_1) \wedge V_k(0^n, t_2)$
- $S'_k(m) = S_k(m)$, $V'_k(m, t) = \begin{cases} V_k(m, t) & \text{if } m \neq 0^n \\ 1 & \text{otherwise} \end{cases}$
- $S'_k(m) = S_k(m)$ without the LSB
  $V'_k(m, t) = V_k(m, t\|0) \;\vee\; V_k(m, t\|1)$

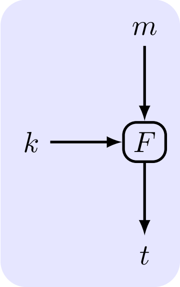
MAC Applications [3]

- **Browser cookie**: includes a MAC tag of the user's account generated by the web server, against the attacker forging others' cookies.
- **TCP SYN cookie**: The server's initial sequence number includes a MAC tag of the client's IP address, port number and some other values generated by the server, against "half-open" DDoS attack in TCP handshake.
- **Timed one-time passwords**: $p = \mathsf{Mac}_k(T)$, where k is the key shared between the user and the service provider, T is the current date $+$ time (usually rounded to the nearest 30 seconds.) The attacker who learns the current p can not gain access to your account in the future.

[3] from The Joy of Cryptography

Content

Constructing Secure MAC



Construction 2

- F *is PRF.* $|m| = n$.
- Gen(1^n): $k \leftarrow \{0,1\}^n$ *u.a.r.*
- Mac$_k(m)$: $t := F_k(m)$.
- Vrfy$_k(m,t)$: $1 \iff t \stackrel{?}{=} F_k(m)$.

Theorem 3

If F is a PRF, Construction is a secure fixed-length MAC.

Lemma 4

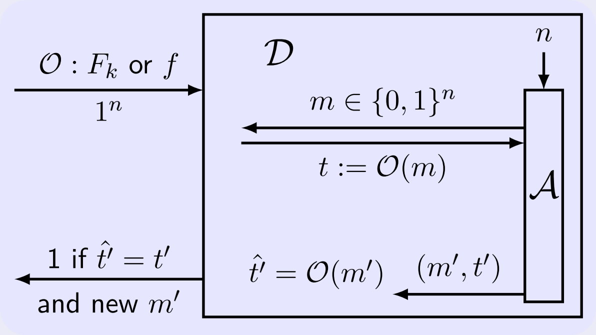
Truncating MACs based on PRFs*: If F is a PRF, so is* $F_k^t(m) = F_k(m)[1,\ldots,t]$.

Proof of Secure MAC from PRF

Idea: Show Π is secure unless F_k is not PRF by reduction.

Proof.

D distinguishes F_k; $\mathcal{A}$ attacks Π.



□

Proof of Secure MAC from PRF (Cont.)

Proof.

(1) If true random f is used, $t = f(m)$ is uniformly distributed.

$$\Pr[D^{f(\cdot)}(1^n) = 1] = \Pr[\mathsf{Macforge}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 2^{-n}.$$

(2) If F_k is used, conduct the experiment $\mathsf{Macforge}_{\mathcal{A},\Pi}(n)$.

$$\Pr[D^{F_k(\cdot)}(1^n) = 1] = \Pr[\mathsf{Macforge}_{\mathcal{A},\Pi}(n) = 1] = \varepsilon(n).$$

According to the definition of PRF,

$$\left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \geq \varepsilon(n) - 2^{-n}.$$

□

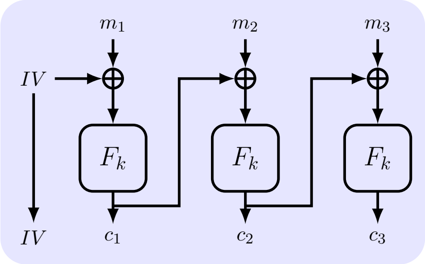
Extension to Variable-Length Messages

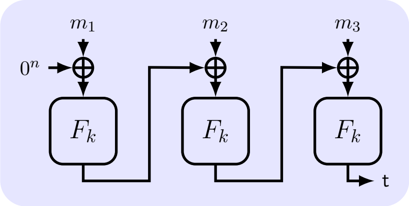
For variable-length messages, would the following suggestions be secure?

- **Suggestion 1**: XOR all the blocks together and authenticate the result. $t := \mathsf{Mac}'_k(\oplus_i m_i)$.
- **Suggestion 2**: Authenticate each block separately. $t_i := \mathsf{Mac}'_k(m_i)$.
- **Suggestion 3**: Authenticate each block along with a sequence number. $t_i := \mathsf{Mac}'_k(i \| m_i)$.

Content

Constructing Fixed-Length CBC-MAC





Modify CBC encryption into CBC-MAC:

- Change random IV to encrypted fixed 0^n,*otherwise*:
  Q: query m_1 and get (IV, t_1); output $m_1' = IV' \oplus IV \oplus m_1$ and $t' =$ ____.
- Tag only includes the output of the final block,*otherwise*:
  Q: query m_i and get t_i; output $m_i' = t_{i-1}' \oplus t_{i-1} \oplus m_i$ and $t_i' =$ ____.

Constructing Fixed-Length CBC-MAC (Cont.)

Construction 5

- *a PRF F and a length function ℓ. $|m| = \ell(n) \cdot n$. $\ell = \ell(n)$. $m = m_1, \ldots, m_\ell$.*
- $\mathsf{Gen}(1^n)$*: $k \leftarrow \{0,1\}^n$ u.a.r.*
- $\mathsf{Mac}_k(m)$*: $t_i := F_k(t_{i-1} \oplus m_i), t_0 = 0^n$. Output $t = t_\ell$.*
- $\mathsf{Vrfy}_k(m,t)$*: $1 \iff t \stackrel{?}{=} \mathsf{Mac}_k(m)$.*

Theorem 6

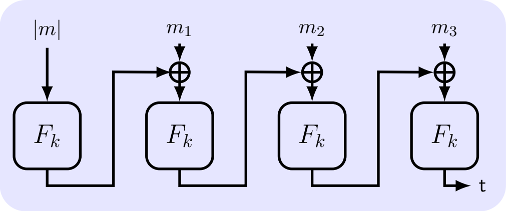
*If F is a PRF, Construction is a secure **fixed-length** MAC.*

Not for **variable-length** message:

Q: For one-block message m with tag t, adversary can append a block ____ and output tag t.

Secure Variable-Length MAC

- **Input-length key separation**: $k_\ell := F_k(\ell)$, use k_ℓ for CBC-MAC.
- **Length-prepending**: Prepend m with $|m|$, then use CBC-MAC.



- **Encrypt last block (ECBC-MAC)**: Use two keys k_1, k_2. Get t with k_1 by CBC-MAC, then output $\hat{t} := F_{k_2}(t)$.

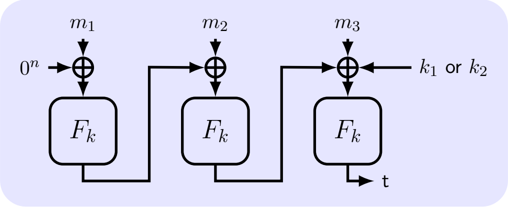
Q: To authenticate a voice stream, which approach do you prefer?

MAC Padding

Padding must be invertible!

$$m_0 \neq m_1 \Rightarrow \mathsf{pad}(m_0) \neq \mathsf{pad}(m_1).$$

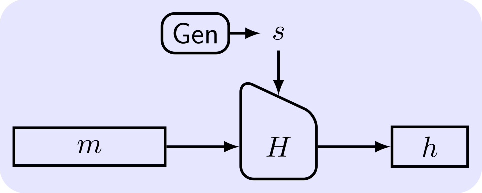
ISO: pad with "100…00". Add dummy block if needed.
Q: What if no dummy block?
CMAC (Cipher-based MAC from NIST): key$= (k, k_1, k_2)$.



- No final encryption: extension attack thwarted by keyed XOR.
- No dummy block: ambiguity resolved by use of k_1 or k_2.

Content

Defining Hash Function



Definition 7

A **hash function (compression function)** is a pair of PPT algorithms (Gen, H) satisfying:

- a key $s \leftarrow \mathsf{Gen}(1^n)$, s is **not kept secret**.
- $H^s(x) \in \{0,1\}^{\ell(n)}$, where $x \in \{0,1\}^*$ and ℓ is polynomial.

If H^s is defined only for $x \in \{0,1\}^{\ell'(n)}$ and $\ell'(n) > \ell(n)$, then (Gen, H) is a **fixed-length** hash function.

Defining Collision Resistance

- **Collision** in H: $x \neq x'$ and $H(x) = H(x')$.
- **Collision Resistance**: infeasible for any PPT alg. to find.

The collision-finding experiment $\mathsf{Hashcoll}_{\mathcal{A},\Pi}(n)$:

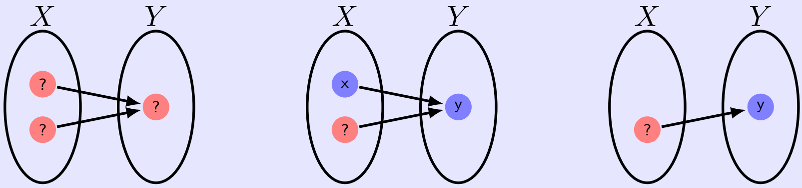
1. $s \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given s and outputs x, x'.
3. $\mathsf{Hashcoll}_{\mathcal{A},\Pi}(n) = 1 \iff x \neq x' \wedge H^s(x) = H^s(x')$.

Definition 8

Π (Gen, H^s) is **collision resistant** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

$$\Pr[\mathsf{Hashcoll}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Weaker Notions of Security for Hash Functions



Collision Resistance

2nd Pre-image Resistance

Pre-image Resistance

- **Collision resistance**: It is hard to find $(x, x'), x' \neq x$ such that $H(x) = H(x')$.
- **Second pre-image resistance**: Given s and x, it is hard to find $x' \neq x$ such that $H^s(x') = H^s(x)$.
- **Pre-image resistance**: Given s and $y = H^s(x)$, it is hard to find x' such that $H^s(x') = y$.

Questions

H is CRHF. Is H' CRHF?

- $H'(m) = H(m) \oplus H(m \oplus 1^{|m|})$
- $H'(m) = H(m) \| H(0)$
- $H'(m) = H(m) \| H(m)$
- $H'(m) = H(m) \oplus H(m)$
- $H'(m) = H(m[0, \ldots, |m| - 2])$
- $H'(m) = H(m \| 0)$
- $H'(m) = H(m)[0, \ldots, |H(m)| - 1]$

Applications of Hash Functions

- **Fingerprinting and Deduplication**: $H(alargefile)$ for virus fingerprinting, deduplication, P2P file sharing
- **Merkle Trees**:
  $H(H(H(file1), H(file2)), H(H(file3), H(file4)))$
  fingerprinting multiple files / parts of a file
- **Password Hashing**: $(salt, H(salt, pw))$ mitigating the risk of leaking password stored in the clear
- **Key Derivation**: $H(secret)$ deriving a key from a high-entropy (but not necessarily uniform) shared secret
- **Commitment Schemes**: $H(info)$ hiding the commited info; binding the commitment to a info

The "Birthday" Problem

The "Birthday" Problem

Q: "*What size group of people do we need to take such that with probability $1/2$ some pair of people share a birthday?*" **A**: 23.

Lemma 9

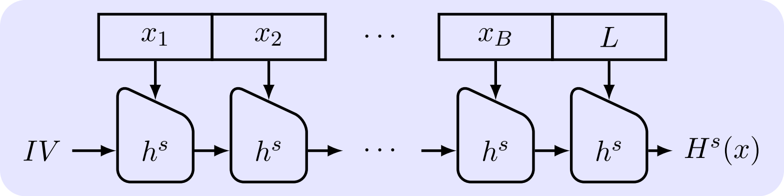
Choose q elements u.a.r from a set of size N, the probability that $\exists\, i \neq j$ with $y_i = y_j$ is $\mathsf{coll}(q, N)$, *then* $\mathsf{coll}(q, N) \leq \frac{q^2}{2N}$.

How many different meaningful sentences are below?

It is **hard/difficult/challenging/impossible** to **imagine/believe** that we will **find/locate/hire** another **employee/person** having similar **abilities/skills/character** as Alice. She has done a **great/super** job.

A principle: The length of hash value should be long enough.

The Merkle-Damgård Transform



Construction 10

Construct ***variable-length*** *CRHF* (Gen, H) *from fixed-length* (Gen, h) *(*2ℓ *bits* $\to \ell$ *bits,* $\ell = \ell(n)$*):*

- Gen*: remains unchanged*
- H*: key* s *and string* $x \in \{0,1\}^*$*,* $L = |x| < 2^\ell$*:*
  - $B := \lceil \frac{L}{\ell} \rceil$ *(# blocks).* ***Pad*** x ***with 0s****.* ℓ*-bit blocks* $x_1, \ldots, x_B$*.* $x_{B+1} := L$*,* L *is encoded using* ℓ *bits*
  - $z_0 := IV = 0^\ell$*. For* $i = 1, \ldots, B+1$*, compute* $z_i := h^s(z_{i-1} \| x_i)$

Security of the Merkle-Damgård Transform

Theorem 11

If (Gen, h) *is a fixed-length CRHF, then* (Gen, H) *is a CRHF.*

Proof.

Idea: a collision in H^s yields a collision in h^s.
Two messages $x \neq x'$ of respective lengths L and L' such that $H^s(x) = H^s(x')$. # blocks are B and B'.
$x_{B+1} := L$ is necessary since **Padding with 0s** will lead to the same input with different messages.

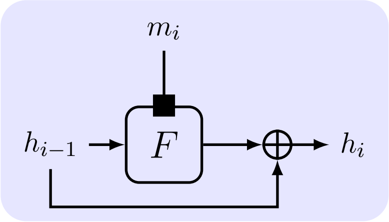
1. $L \neq L'$: $z_B \| L \neq z_{B'} \| L'$
2. $L = L'$: $z_{i^*-1} \| x_{i^*} \neq z'_{i^*-1} \| x'_{i^*}$

So there must be $x \neq x'$ such that $h^s(x) = h^s(x')$. □

Security on MD transform variations in Homework.

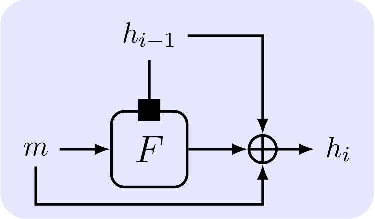
CRHF from Block Cipher

Davies-Meyer (SHA-1/2, MD5)



$$h_i = F_{m_i}(h_{i-1}) \oplus h_{i-1}$$

Miyaguchi-Preneel (Whirlpool)



$$h_i = F_{h_{i-1}}(m_i) \oplus h_{i-1} \oplus m$$

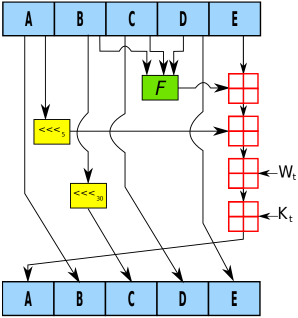
Theorem 12

If F is modeled as an ideal cipher, then Davies-Meyer construction yields a CRHF.

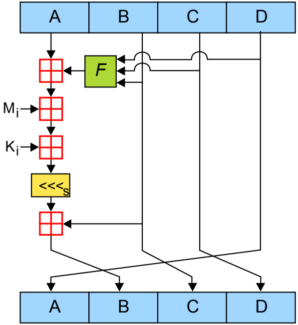
Q: what if $h_i = F_{m_i}(h_{i-1})$ without XOR with h_{i-1}?
Q: what if F is not ideal such that $\exists x, F_k(x) = x$?

Cryptographic Hash Functions: SHA-1 and MD5

SHA-1:



MD5:



A, B, C, D and E are 32-bit words of the state; F is a nonlinear function that varies; $\lll n$ denotes a left bit rotation by n places; W_t/M_t is the expanded message word of round t; K_t is the round constant of round t; $\boxplus$ denotes addition modulo 2^{32}.

- Finding a collision in 128-bit MD5 requires time $2^{20.96}$
- Finding a collision in 160-bit SHA-1 requires time 2^{51}

Content

Hash-and-MAC

Construction 13

$(\widetilde{\mathsf{Gen}}, H)$ *is a CRHF.* $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ *is a fixed-length MAC.*

- $\mathsf{Gen}'(1^n)$: (k, s). $s \leftarrow \widetilde{\mathsf{Gen}}, k \leftarrow \mathsf{Gen}$.
- $\mathsf{Mac}'_{s,k}(m)$: $t := \mathsf{Mac}_k(H^s(m))$.
- $\mathsf{Vrfy}'_{s,k}(m, t)$: $1 \iff \mathsf{Vrfy}_k(H^s(m), t) = 1$.

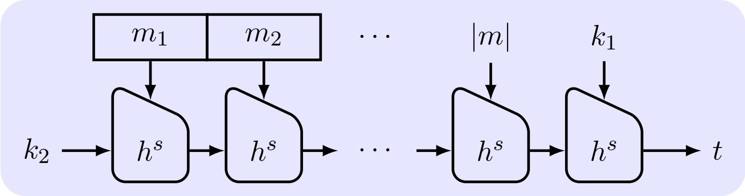
Theorem 14

The construction is a secure MAC for arbitrary-length messages.

Idea of proof: if the adversary has forged a tag on the "new message" m^*, then

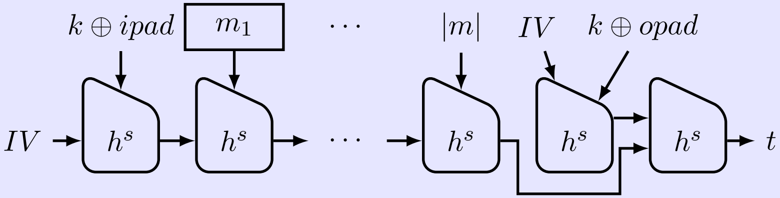
- Case 1: If there is a queried messages m such that $H^s(m) = H^s(m^*)$, then there is a collision in H^s.
- Case 2: If there is no queried messages m such that $H^s(m) = H^s(m^*)$, then the adversary has forged a valid tag on the "new message" $H^s(m^*)$ for MAC.

Nested MAC (NMAC)



- NMAC is a MAC using CRHF (MD transform) without using PRF.
- k_2 is not needed once h is CRHF, while it is needed if h is **weak collision resistance**: It is hard to find $(x, x'), x' \neq x$ such that $H^s_{k_2}(x) = H^s_{k_2}(x')$ **without knowing** k_2.
- **Disadvantage**: IV of H must be modified.

Hash-based MAC (HMAC)



Construction 15

$(\widetilde{\mathsf{Gen}}, h)$ *is a fixed-length CRHF.* $(\widetilde{\mathsf{Gen}}, H)$ *is the Merkle-Damgård transform.* IV, opad *(0x36),* ipad *(0x5C) are constants.*

- $\mathsf{Gen}(1^n)$*: Output* (s, k)*.* $s \leftarrow \widetilde{\mathsf{Gen}}, k \leftarrow \{0,1\}^n$ *u.a.r*
- $\mathsf{Mac}_{s,k}(m)$: $t := H^s_{IV}\Big((k \oplus \mathsf{opad}) \| H^s_{IV}((k \oplus \mathsf{ipad}) \| m)\Big)$
- $\mathsf{Vrfy}_{s,k}(m,t)$: $1 \iff t \stackrel{?}{=} \mathsf{Mac}_{s,k}(m)$

(In)Security of (Before-)HMAC [4]

We investigate HMAC's security by showing the insecurity of some before-HMAC designs.

- **Prepend the key** $H^s(k\|x)$: Vulnerable to length extension attack. Given $H^s(k\|x)$ and the length of x, get the valid tag $H^s(k\|x\|x')$ for a new message $x\|x'$.
- **Append the key** $H^s(x\|k)$: A collision in the **weak** CRHF has a collision in the MAC. Recall that there is an appended key for **weak** CRHF in NMAC.
- **Envelope** $H^s(k\|x\|k)$: Some known vulnerabilities with this approach, even when two different keys are used. This needs reasonable pseudorandomness assumptions on h.
- **Two-key nest** $H^s(k\|H^s(k\|x))$: NMAC with 2 keys, HMAC with 1 key

[4] from BonehShoup v0.5 p.303

Remarks on HMAC

- HMAC is based on NMAC which was first published in a paper "Keying Hash Functions for Message Authentication" by Mihir Bellare, Ran Canetti, and Hugo Krawczyk in 1996.
- HMAC became an industry standard (RFC2104) in 1997
- HMAC is faster than CBC-MAC

Verification timing attacks

Keyczar crypto library (Python):

```
def Verify(key, msg, sig_bytes):
    return HMAC(key, msg) == sig_bytes
```

The problem: implemented as a byte-by-byte comparison

In Xbox 360, a difference of 2.2 milliseconds between rejection times of i vs. i + 1 bytes.

Don't implement it yourself

Content

Definition of Information-Theoretic MAC Security

It is impossible to achieve "perfect" MAC, as the adversary can output a valid tag with probability $1/2^{|t|}$ at least.

The one-time MAC experiment $\text{Macforge}^{1-\text{time}}_{\mathcal{A},\Pi}$:

1. $k \leftarrow \text{Gen}$.
2. $\mathcal{A}$ outputs a message m', and is given a tag $t' \leftarrow \text{Mac}_k(m')$, and outputs (m, t).
3. $\text{Macforge}^{1-\text{time}}_{\mathcal{A},\Pi} = 1 \iff \text{Vrfy}_k(m, t) = 1 \wedge m \neq m'$.

Definition 16

A MAC Π is **one-time ε-secure** if $\forall$ PPT $\mathcal{A}$:

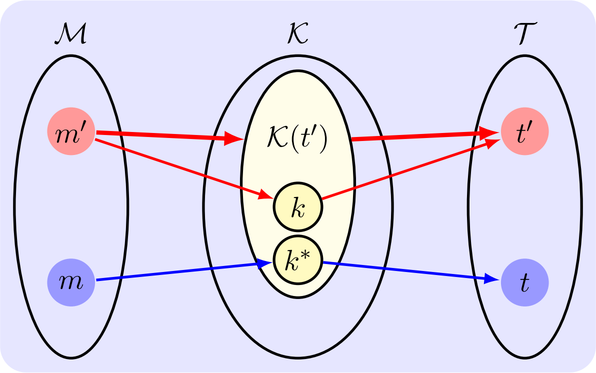
$$\Pr[\text{Macforge}^{1-\text{time}}_{\mathcal{A},\Pi} = 1] \leq \varepsilon.$$

Understanding Information-Theoretic MACs

An adversary will forge the tag in the following steps.

1. Obtain a tag t' from a MAC query for a fixed message m'
2. Obtain $\mathcal{K}(t') \stackrel{\text{def}}{=} \{k | \mathsf{Vrfy}_k(m', t') = 1\}$ by using his unlimited computing power
3. Output (m, t) using a key k^* from $\mathcal{K}(t')$

Question: What if $\mathcal{K}(t')$ is too large or too small?



Construction of Information-Theoretic MACs

Definition 17

A function $h\colon \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a **Strongly Universal (or pairwise-independent) Function (SUF)** if for all distinct $m, m' \in \mathcal{M}$ and all $t, t' \in \mathcal{T}$, it holds that:

$$\Pr[h_k(m) = t \wedge h_k(m') = t'] = 1/|\mathcal{T}|^2.$$

where the probability is taken over uniform choice of $k \in \mathcal{K}$.

Construction 18

- *Let $h\colon \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be an SUF.*
- Gen*: $k \leftarrow \{0,1\}^n$ u.a.r.*
- Mac$_k(m)$*: $t := h_k(m)$.*
- Vrfy$_k(m,t)$*: $1 \iff t \stackrel{?}{=} h_k(m)$. (If $m \notin \mathcal{M}$, then output 0.)*

Construction of An SUF

Theorem 19

For any prime P, the function h is an SUF:

$$h_{a,b}(m) \stackrel{\text{def}}{=} [a \cdot m + b \mod p]$$

Proof.

$h_{a,b}(m) = t$ and $h_{a,b}(m') = t'$, only if $a \cdot m + b = t \mod p$ and $a \cdot m' + b = t' \mod p$. We have $a = [(t - t') \cdot (m - m')^{-1} \mod p]$ and $b = [t - a \cdot m \mod p]$, which means there is a unique key (a, b). Since there are $|\mathcal{T}|^2$ keys,

$$\Pr[h_k(m) = t \wedge h_k(m') = t'] = \frac{1}{|\mathcal{T}|^2}.$$

□

Security of Construction from An SUF

Theorem 20

If h is an SUF, Construction is a $1/|\mathcal{T}|-$secure MAC.

Proof.

Assume that $\mathcal{A}$ is deterministic and receives tag t' for the message m', where m' is fixed. The pair (m,t) that $\mathcal{A}$ outputs is a deterministic function of (m',t').

$$\begin{aligned}
\Pr[\mathsf{Macforge}^{1-\mathsf{time}}_{\mathcal{A},\Pi}=1] &= \sum_{t'\in\mathcal{T}} \Pr[\mathsf{Macforge}^{1-\mathsf{time}}_{\mathcal{A},\Pi}=1 \wedge h_k(m')=t'] \\
&= \sum_{t'\in\mathcal{T}} \Pr[h_k(m)=t \wedge h_k(m')=t'] \\
&= \sum_{t'\in\mathcal{T}} \frac{1}{|\mathcal{T}^2|} = \frac{1}{|\mathcal{T}|}
\end{aligned}$$

□

Limitations on Information-Theoretic MACs

Limitations: Any ℓ-time 2^{-n}-secure MAC requires keys of length at least $(\ell + 1) \cdot n$.

Theorem 21

Let Π be a 1-time 2^{-n}-secure MAC where all keys are the same length. Then the keys must have length at least $2n$.

Proof.

Let $\mathcal{K}(t') \stackrel{\text{def}}{=} \{k | \mathsf{Vrfy}_k(m', t') = 1\}$. For any t', $|\mathcal{K}(t')| \leq 2^{-n} \cdot |\mathcal{K}|$. Otherwise, (m', t') would be a valid forgery with probability at least $|\mathcal{K}(t')|/|\mathcal{K}| > 2^{-n}$. The probability that $\mathcal{A}$ outputs a valid forgery by guessing k from $|\mathcal{K}(t')|$ is at least

$$\sum_{t'} \Pr[\mathsf{Mac}_k(m') = t'] \cdot \frac{1}{|\mathcal{K}(t')|} \geq \sum_{t'} \Pr[\mathsf{Mac}_k(m') = t'] \cdot \frac{2^n}{|\mathcal{K}|} = \frac{2^n}{|\mathcal{K}|}$$

As the probability is at most 2^{-n}, $|\mathcal{K}| \geq 2^{2n}$. Since all keys have the same length, each key must have length at least $2n$. $\square$

Summary

- Authentication means existential unforgeability.
- Secure MAC is constructed by using PRF.
- Secure MAC is constructed by using keyed CRHF.
- Information-theoretic MAC security requires very, very long key.