Message Authentication Codes and Collision-Resistant Hash Functions

Yu Zhang

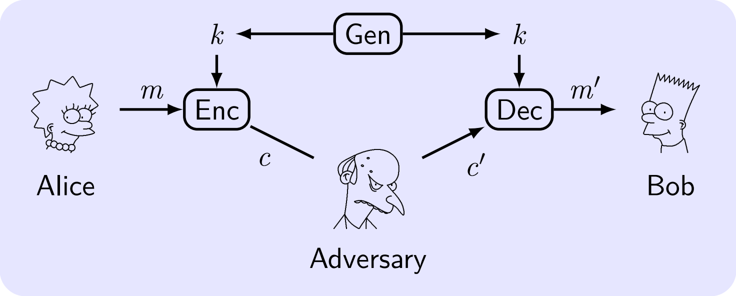
HIT/CST/NIS

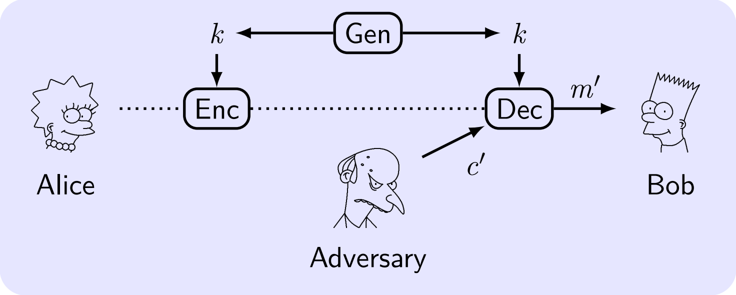
Cryptography, Spring, 2012

Outline

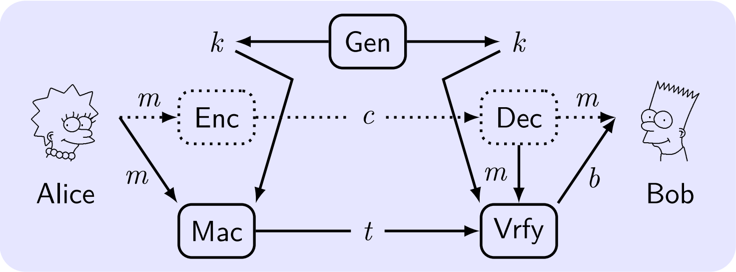
Content

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 $k \leftarrow \text{Gen}(1^n), |k| \geq n$.
- **Tag-generation** algorithm $t \leftarrow \text{Mac}_k(m)$.
- **Verification** algorithm $b := \text{Vrfy}_k(m, t)$.
- **Message authentication code**: $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$.
- **Basic correctness requirement**: $\text{Vrfy}_k(m, \text{Mac}_k(m)) = 1$.

Security of MAC

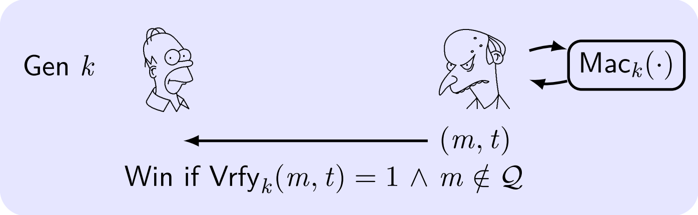
- **Intuition**: No adversary should be able to generate a **valid** tag on any "**new**" message[1] that was not previously sent.
- **Replay attack**: Copy a message and tag previously sent. (**excluded by only considering "new" message**)
  - Sequence numbers: receiver must store the previous ones.
  - 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. $\mathsf{Macforge}_{\mathcal{A},\Pi}(n) = 1 \iff \mathsf{Vrfy}_k(m, t) = 1 \wedge m \notin \mathcal{Q}$.



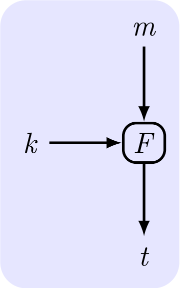
Definition 1

A MAC Π is **existentially unforgeable under an adaptive CMA** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that:

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

Content

Constructing Secure MACs



Construction 2

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

Theorem 3

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

Lemma 4

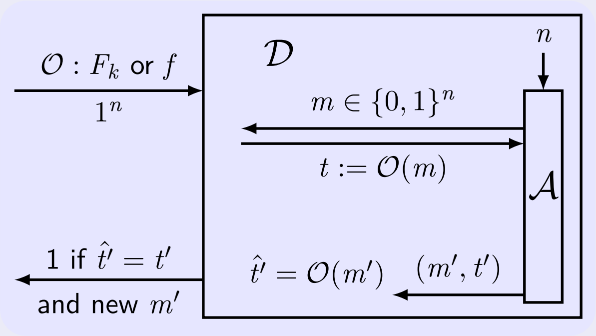
Truncating MACs based on PRFs: *If F is a PRF, so is $F_k^t(m) = F_k(m)[1,\ldots,t]$.*

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 PGF,

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

$\square$

Extension to Variable-Length Messages

- **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. $t_i := \mathsf{Mac}'_k(i \| m_i)$.
- **Weakness**: forgeable, changing the order, dropping blocks.
- **Countermeasure**: add information.
  - random "**message identifier**" provides randomness; prevents combination.
  - **sequence number** prevents reordering.
  - the **length** of message prevents dropping/appending.

Constructing Secure Variable-Length MACs

Construction 5

- $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ *be a fixed-length MAC.*
- Gen*: is identical to* Gen$'$.
- Mac*:* m *of length* $\ell < 2^{n/4}$ *and of* d *blocks* $m_1, \ldots, m_d$ *of length* $n/4$ *(padded with 0s);* $r \leftarrow \{0,1\}^{n/4}$.
  For $i = 1, \ldots, d$*,* $t_i \leftarrow \mathsf{Mac}'_k(r\|\ell\|i\|m_i)$*,* i *and* ℓ *are uniquely encoded as strings of length* $n/4$.
  Output $t := \langle r, t_1, \ldots, t_d \rangle$.
- Vrfy*: Input* m *of* d' *blocks and check* $d' = d$.
  Output $1 \iff \mathsf{Vrfy}'_k(r\|\ell\|i\|m_i, t_i) = 1$ *for* $1 \leq i \leq d$.

Theorem 6

If Π' *is a secure fixed-length MAC, Construction is a secure MAC.*

Proof of Secure Variable-Length MACs

Intuition: The extra information prevents all possible attacks.

Proof.

$$\begin{aligned}\text{Repeat} &: \text{ the same identifier } r \text{ is used twice by oracle } \mathcal{O}.\\ \text{Forge} &: \text{ at least one new block } r\|\ell\|i\|m_i \text{ is forged.}\\ \text{Break} &: \text{ Macforge}_{\mathcal{A},\Pi}(n)=1, \Pr[\text{Break}]=\varepsilon(n).\end{aligned}$$

$$\begin{aligned}\Pr[\text{Break}] =& \Pr[\text{Break} \wedge \text{Repeat}] + \Pr[\text{Break} \wedge \overline{\text{Repeat}} \wedge \overline{\text{Forge}}]\\ &+ \Pr[\text{Break} \wedge \overline{\text{Repeat}} \wedge \text{Forge}].\end{aligned}$$

To prove the below statements:

1. $\Pr[\text{Break} \wedge \text{Repeat}] \le \Pr[\text{Repeat}] \le \text{negl}(n)$.
2. $\Pr[\text{Break} \wedge \overline{\text{Repeat}} \wedge \overline{\text{Forge}}] = 0$.
3. For Π', $\Pr[\text{Break}'] = \Pr[\text{Break} \wedge \text{Forge}] \ge$ $\Pr[\text{Break} \wedge \overline{\text{Repeat}} \wedge \text{Forge}] \ge \varepsilon(n) - \text{negl}(n)$.

□

Proof of Secure Variable-Length MACs (Cont.)

Proof.

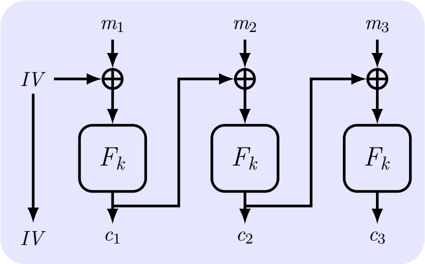
1. $r \leftarrow \{0,1\}^{\frac{n}{4}}$. By "brithday bound", $\Pr[\text{Repeat}] \leq q(n)^2/2^{\frac{n}{4}}$.
2. If Repeat does not occur, Break implies Forge.
   $\mathcal{A}$ finally outputs $(m,t), t := \langle r, t_1, \ldots, t_d \rangle$.
   - r is new, then $r\|\ell\|i\|m_i$ is new.
   - r is used exactly once, then the queried message $m' \neq m$.
     - $\ell' \neq \ell$, then $r\|\ell\|i\|m_i$ is new.
     - $\ell' = \ell$, then $\exists\ m_i' \neq m_i$, so $r\|\ell\|i\|m_i'$ is new.

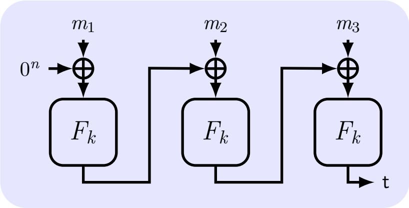
   So the block is new, Forge occurs.
3. Reduce $\mathcal{A}'$ to $\mathcal{A}$: $\mathcal{A}'$ attacks Π' with $\mathcal{A}$ as a sub-routine and answer the queries of $\mathcal{A}$ with $\mathcal{A}'$'s own oracle. $\mathcal{A}$ output (m,t); $\mathcal{A}'$ parses it and output a new block $(r\|\ell\|i\|m_i, t_i)$ if possible.

$\square$

Content

Constructing Fixed-Length CBC-MAC





Modify CBC encryption into CBC-MAC:

- Change random IV to encrypted fixed 0^n,*otherwise*: query m_1 and get (IV, t_1); output $m_1' = IV' \oplus IV \oplus m_1$ and (IV', t_1).
- Tag only includes the output of the final block,*otherwise*: 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 7

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

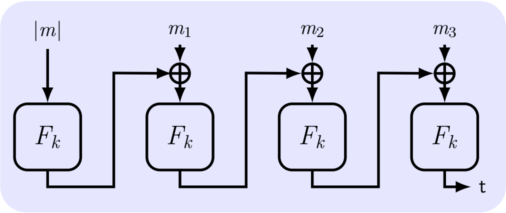
Theorem 8

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

Not for **variable-length** message:
For one-block message m with tag t, adversary can append a block $t \oplus m$ and output tag t.

Secure Variable-Length MAC

- **Option 1**: $k_\ell := F_k(\ell)$, use k_ℓ for CBC-MAC.
- **Option 2**: Prepend m with $|m|$, then use CBC-MAC.



- **Option 3 (ECBC-MAC)**: Use two keys k_1, k_2. Get t with k_1 by CBC-MAC, then output $\hat{t} := F_{k_2}(t)$.

Lessons learned

Wrap CBC-MAC with PRF(length/tag), and only output is tag!

Brute-force Attack against CBC-MAC

Query $2^{|t|/2}$ message to find $m \neq m'$ and $t = t'$.

Extension property of ECBC-MAC:

$$\forall x, y, z : F_k(x) = F_k(y) \Rightarrow F_k(x\|z) = F_k(y\|z).$$

So the tag of $m\|w$ is the same with that of $m'\|w$.

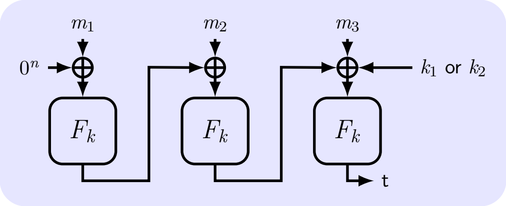
Lesson: the tag space should be enough large.
Improvement: Add a random string r, and output $(r, \mathsf{Mac}_{k'}(t\|r))$ instead of t.

MAC Padding

Padding must be invertible!

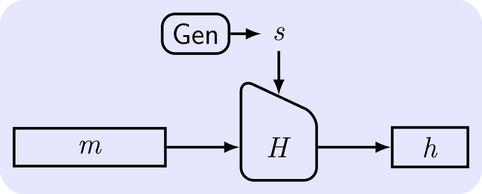
$$m_0 \neq m_1 \Rightarrow \mathsf{pad}(m_0) \neq \mathsf{pad}(m_1).$$

ISO: pad with "100...00". Add dummy block if needed.
CMAC (Cipher-based MAC from NIST): key$= (k, k_1, k_2)$.



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

Content



Definition 9

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

- a key $s \leftarrow \text{Gen}(1^n)$, 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 $\textsc{ppt}$ alg. to find.

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

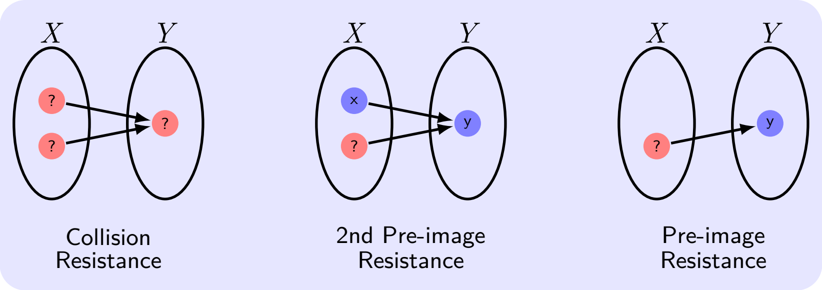
1. $s \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given s and outputs x, x'.
3. $\mathsf{Hashcoll}_{\mathcal{A},\Pi}(n) = 1 \iff x \neq x' \wedge H^s(x) = H^s(x')$.

Definition 10

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

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

Weaker Notions of Security for Hash Functions



- **Collision resistance**: 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 $H^s(x') = y$.

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 in the group share a birthday?*”
A: 23.

Lemma 11

Choose q elements $y_1, \ldots, y_q$ 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*

$$\mathsf{coll}(q, N) \leq \frac{q^2}{2N}.$$

$$\mathsf{coll}(q, N) \geq \frac{q(q-1)}{4N} \quad \textit{if } q \leq \sqrt{2N}.$$

$$\mathsf{coll}(q, N) = \Theta(q^2/N) \quad \textit{if } q < \sqrt{N}.$$

A Generic "Birthday" Attack

- **Birthday Attack**: $H : \{0,1\}^* \to \{0,1\}^\ell$. Choose q distinct inputs $x_1, \ldots, x_q \in \{0,1\}^{2\ell}$, check whether any of two $y_i := H(x_i)$ are equal.
- **Birthday problem**: Choose $y_1, \ldots, y_q \leftarrow \{0,1\}^\ell$ *u.a.r*, coll$(q, 2^\ell) =?$
- Collision occurs with a high probability when $\mathcal{O}(q) = \mathcal{O}(2^{\ell/2})$.
- To let time $T > 2^{\ell/2}$, then $\ell = 2\log T$ at least.
- Work only for collision resistance, no generic attacks for 2nd pre-image or pre-image resistance better than 2^ℓ.
- Require too much space $\mathcal{O}(2^{\ell/2})$.

Improved Birthday Attack

Algorithm 1: Improved birthday attack

input : A hash function $H : \{0,1\}^* \to \{0,1\}^\ell$

output: Distinct x, x' with $H(x) = H(x')$

$x_0 \leftarrow \{0,1\}^{\ell+1}$, $x' := x := x_0$

for $i = 1$ **to** $2^{\ell/2} + 1$ **do**

  $x := H(x)$, $x' := H(H(x'))$ // $x = H^i(x_0)$, $x' = H^{2i}(x_0)$

  if $x = x'$ **then break**

if $x \neq x'$ **then return** fail

$x' := x$, $x := x_0$

for $j = 1$ **to** i **do**

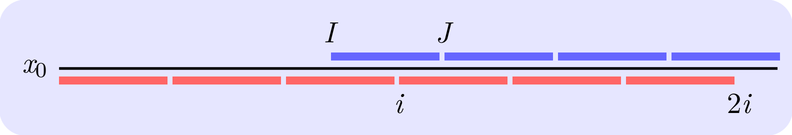
  if $H(x) = H(x')$ **then return** x, x' and **halt**

  else $x := H(x), x' := H(x')$ // $x = H^j(x_0)$, $x' = H^{j+i}(x_0)$

Proof of Improved Birthday Attack

Lemma 12

Let $x_1, \ldots, x_q$ be a sequence of values with $x_m = H(x_{m-1})$. If $x_I = x_J$ with $I < J$, then $\exists\, i < J$ such that $x_i = x_{2i}$.



Proof.

If $x_I = x_J$, then $x_I, x_{I+1}, \ldots$ repeats with period $J - I$.
Let i to be the smallest multiple of $J - I$ with $i \geq I$,

$$i \stackrel{\text{def}}{=} (J - I) \cdot \lceil I/(J - I) \rceil.$$

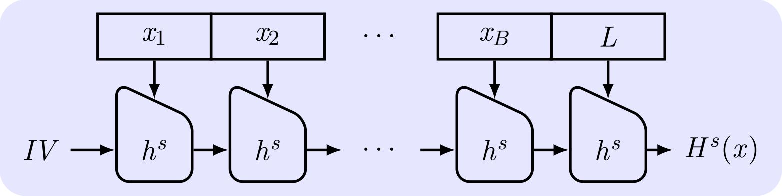
$i < J$ since $I, \ldots, J - 1$ contains a multiple of $J - I$.
Since $2i - i = i$ is a multiple of the period and $i \geq I$, $x_i = x_{2i}$. $\square$

Constructing “Meaningful” Collisions

An example with 288 different meaningful sentences

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.

The Merkle-Damgård Transform



Construction 13

*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 14

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

Proof.

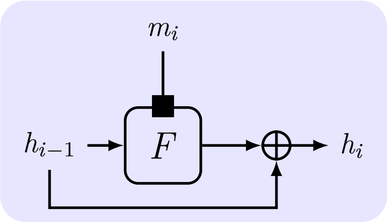
Idea: a collision in H^s yields a collision in h^s.
Two messages $x \neq x'$ of respective lengths L and L' such that $H^s(x) = H^s(x')$. # blocks are B and B'.
$x_{B+1} := L$ is necessary since **Padding with 0s** will lead to the same input with different messages.

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 $h^s(x) = h^s(x')$. □

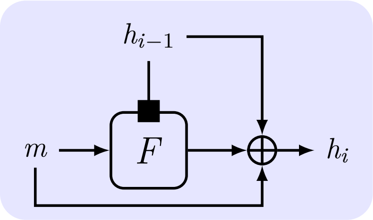
CRHF from Block Cipher

Davies-Meyer:



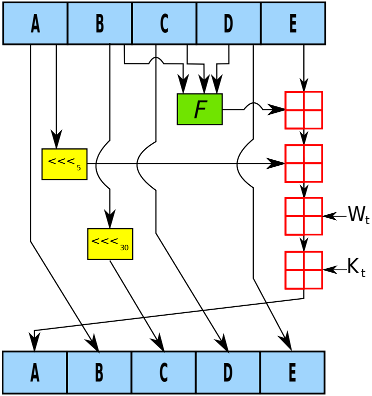
Used by SHA-1/2, MD5.

Miyaguchi-Preneel:

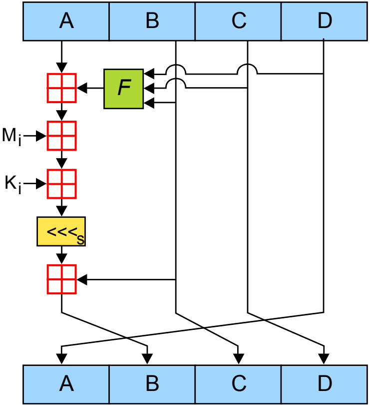


Used by Whirlpool (in ISO/IEC 10118-3).

SHA-1:



MD5:



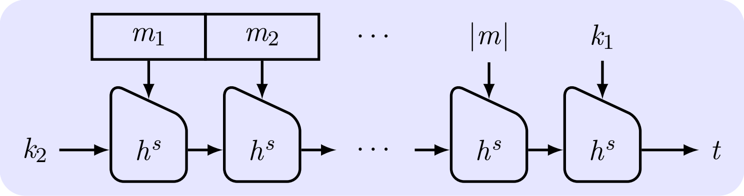
A, B, C, D and E are 32-bit words of the state; F is a nonlinear function that varies; $\lll n$ denotes a left bit rotation by n places; W_t/M_t is the expanded message word of round t; K_t is the round constant of round t; $\boxplus$ denotes addition modulo 2^{32}.

Collision-Resistant Hash Functions in Practice

- The hash functions used in practice are generally un-keyed.
- The constructions are more heuristic in nature.
- Finding a collision in MD5 (Message Digest 5) with 128-bit output requires time $2^{20.96}$.
- Finding a collision in SHA-1 (Secure Hash Algorithm) with a 160-bit output requires time 2^{51}.

Content

Nested MAC (NMAC)



Construction 15

$(\widetilde{\mathsf{Gen}}, h)$ *is a fixed-length CRHF.* $(\widetilde{\mathsf{Gen}}, H)$ *is Merkle-Damgård transform. NMAC:*

- $\mathsf{Gen}(1^n)$*: Output* (s, k_1, k_2). $s \leftarrow \widetilde{\mathsf{Gen}}, k_1, k_2 \leftarrow \{0,1\}^n$ *u.a.r.*
- $\mathsf{Mac}_{s,k_1,k_2}(m)$: $t_i := h^s_{k_1}(H^s_{k_2}(m))$. $h^s_k \stackrel{\text{def}}{=} h^s(k\|x)$.
  $H^s_{k_2}$ *is inner function;* $h^s_{k_1}$ *is outer function.*
- $\mathsf{Vrfy}_{s,k_1,k_2}(m,t)$: $1 \iff t \stackrel{?}{=} \mathsf{Mac}_{s,k_1,k_2}(m)$.

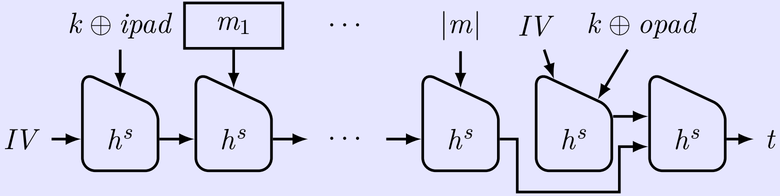
Security of NMAC

Theorem 16

If $(\widetilde{\mathsf{Gen}}, h)$ is CRHF and yields a secure MAC, then NMAC is secure. (existentially unforgeable under an adaptive CMA for arbitrary-length messages)

- k_2 is not needed once $(\widetilde{\mathsf{Gen}}, h)$ is CRHF.
  - **Weak collision resistance**: It is hard to find $(x, x'), x' \neq x$ such that $H^s_{k_2}(x) = H^s_{k_2}(x')$.
  - $H^{k_2}_s(x)$ is hidden by $h^{k_1}_s(H^{k_2}_s(x))$.
  - **Disadvantage**: IV of H must be modified.

Hash-based MAC (HMAC)



Construction 17

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

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

Security of HMAC

Theorem 18

$$G(k) \overset{\text{def}}{=} h^s(IV \| (k \oplus \mathsf{opad})) \| h^s(IV \| (k \oplus \mathsf{ipad})) = k_1 \| k_2.$$

$(\widetilde{\mathsf{Gen}}, h)$ *is CRHF. If* G *is a PRG, then HMAC is secure.*

- HMAC is an industry standard (RFC2104) and is widely used in practice.
- HMAC is faster than CBC-MAC.
- Before HMAC, a common mistake was to use $H^s(k \| x)$.
- *Don't implement it yourself.* Verification timing attacks.

Summary

- adaptive CMA, replay attack, birthday attack.
- existential unforgeability, collision resistance.
- CBC-MAC, CRHF, Merkle-Damgård transform, NMAC, HMAC.