Message Authentication Codes and Collision-Resistant Hash Functions

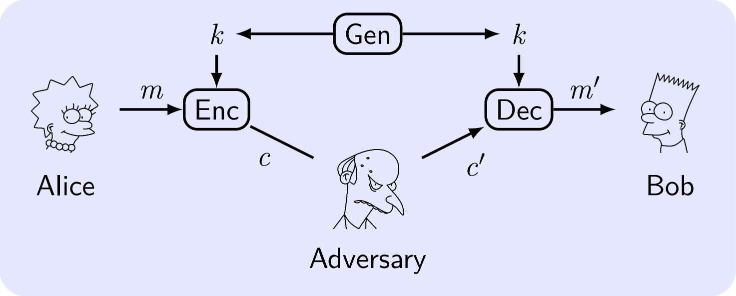
Yu Zhang

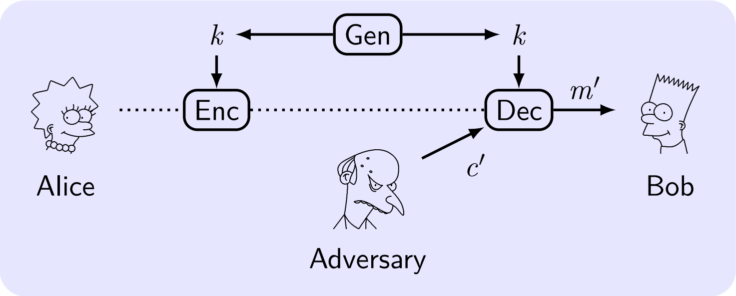
HIT/CST/NIS

Cryptography, Spring, 2012

Outline

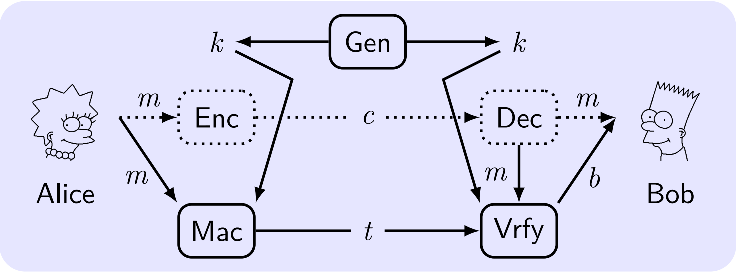
Content


Gen
k
k
m
Enc
Dec
m'
c
c'
Alice
Bob
Adversary


Gen
k
k
Enc
Dec
m'
c'
Alice
Bob
Adversary

Content

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 that was not previously sent (and authenticated) by one of the communicating parties.
- **Replay attack**: Copy a message and tag previously sent by the legitimate parties.
  - 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**: any one.
  - **Selective forgery**: message chosen *prior* to the attack.
  - **Universal forgery**: any *given* message.
- **Adaptive chosen-message attack (CMA)**: able to obtain tags on any message chosen adaptively *during* its attack.

Definition of MAC Security

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

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

Definition 1

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

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

Content

Constructing Secure MACs

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$_k(m,t)$: $1 \iff t \stackrel{?}{=} F_k(m)$.

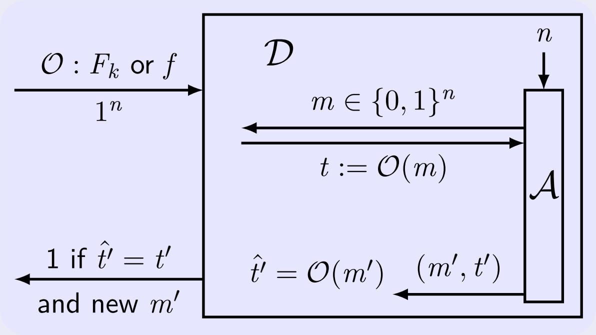
Theorem 3

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

Proof of Secure MAC from PRF

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

Proof.



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

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.
- **Idea**: Including a random "message identifier", a sequence number, and the length of the message.

Constructing Secure Variable-Length MACs

Construction 4

- $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ *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 5

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.

Repeat : the same identifier r is used twice by oracle $\mathcal{O}$.

Forge : at least one new block $r\|\ell\|i\|m_i$ is forged.

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

$$\Pr[\mathsf{Break}] = \Pr[\mathsf{Break} \wedge \mathsf{Repeat}] + \Pr[\mathsf{Break} \wedge \overline{\mathsf{Repeat}} \wedge \overline{\mathsf{Forge}}] + \Pr[\mathsf{Break} \wedge \overline{\mathsf{Repeat}} \wedge \mathsf{Forge}].$$

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

□

Proof of Secure Variable-Length MACs (Cont.)

Proof.

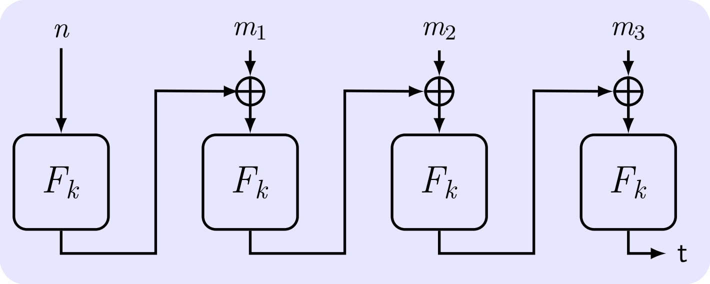
1. $r \leftarrow \{0,1\}^{\frac{n}{4}}$. By "brithday bound", $\Pr[\text{Repeat}] \leq q(n)^2/2^{\frac{n}{4}}$.
2. If Repeat does not occur, Break implies Forge.
   $\mathcal{A}$ finally outputs $(m,t), t := \langle r, t_1, \ldots, t_d \rangle$.
   - r is new, then $r\|\ell\|i\|m_i$ is new.
   - r is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

□

Content

Constructing CBC-MAC



Construction 6

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

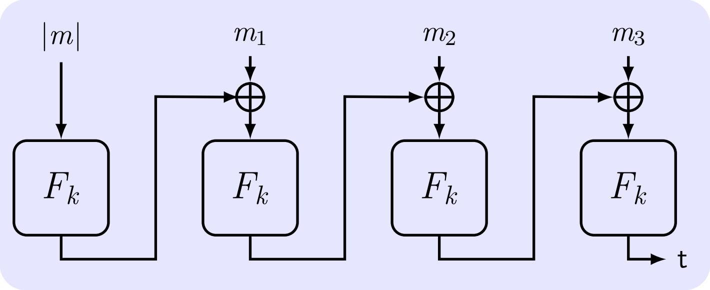
Secure Fixed/Variable-Length MAC

Theorem 7

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

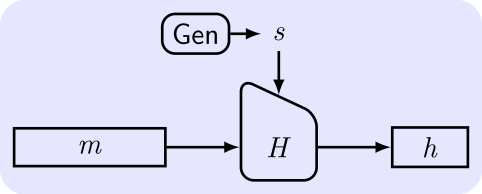
Secure CBC-MAC for variable-length messages:

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



Content

Defining Hash Function



Definition 8

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 is a pair of distinct input x and x' such that $H(x) = H(x')$.
- **Collision Resistance**: it is infeasible for any PPT algorithm to find a collision.

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

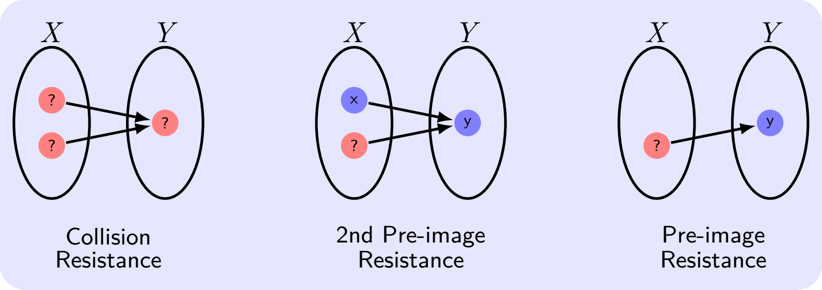
1. $s \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given s and outputs x, x'.
3. $\mathsf{Hashcoll}_{\mathcal{A},\Pi}(n) = 1 \iff x \neq x' \wedge H^s(x) = H^s(x')$.

Definition 9

Π (H, 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**: 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 $H^s(x') = y$.

Applications of Hash Functions

- **digital signatures**:CRHF
- **information authentication/integrity check**
- **protection of passwords**: pre-image resistant.
- **confirmation of knowledge/commitment**: CRHF
- **pseudo-random string generation/key derivation**
- **micropayments (e.g. micromint)**
- **construction of MACs, stream/block ciphers**

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 10

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 $\mathrm{coll}(q, N)$*, then*

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

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

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

A Generic "Birthday" Attack

- **Birthday Attack**: $H : \{0,1\}^* \to \{0,1\}^\ell$. Choose q distinct inputs $x_1, \dots, x_q \in \{0,1\}^{2\ell}$, check whether any of two $y_i := H(x_i)$ are equal.
- **Birthday problem**: Choose $y_1, \dots, y_q \leftarrow \{0,1\}^\ell$ *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**

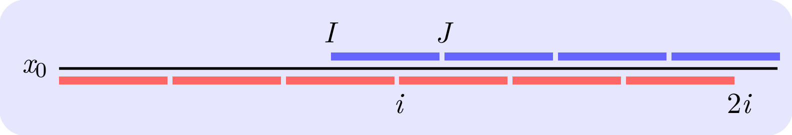
  if $H(x) = H(x')$ **then return** x, x' and **halt**

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

Proof of Improved Birthday Attack

Lemma 11

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 $i \geq I$,

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

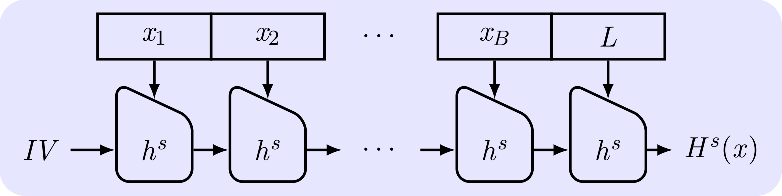
$i < J$ since $I, \ldots, J - 1$ contains a multiple of $J - I$.
Since $2i - i = i$ is a multiple of the period and $i \geq I$, $x_i = x_{2i}$. $\square$

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 12

(Gen, h) *is a fixed-length CRHF (input length* $2\ell(n)$ *and output length* $\ell(n)$*). Construct a* ***variable-length*** *CRHF* (Gen, H)*:*

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

Security of the Merkle-Damgård Transform

Theorem 13

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'.

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

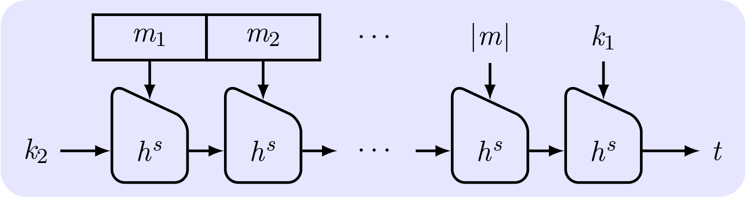
□

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 14

$(\widetilde{\mathsf{Gen}}, h)$ *is a fixed-length CRHF.* $(\widetilde{\mathsf{Gen}}, H)$ *is the 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.*
- $\mathsf{Mac}_{s,k_1,k_2}(m)$*:* $t_i := h^s_{k_1}(H^s_{k_2}(m))$.
- $\mathsf{Vrfy}_{s,k_1,k_2}(m,t)$*:* $1 \iff t \stackrel{?}{=} \mathsf{Mac}_{s,k_1,k_2}(m)$.

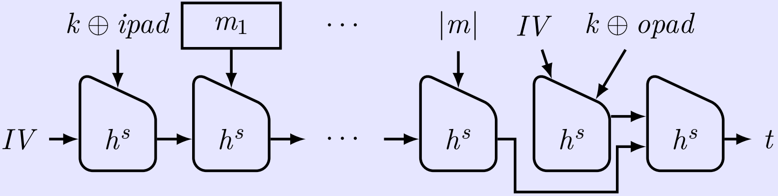
Security of NMAC

Theorem 15

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

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

Hash-based MAC (HMAC)



Construction 16

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

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

Security of HMAC

Theorem 17

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

$(\widetilde{\mathsf{Gen}}, h)$ *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 $H^s(k \| x)$ as a MAC.

Content

Recall Security Against CCA

The CCA indistinguishability experiment $\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi}(n)$:

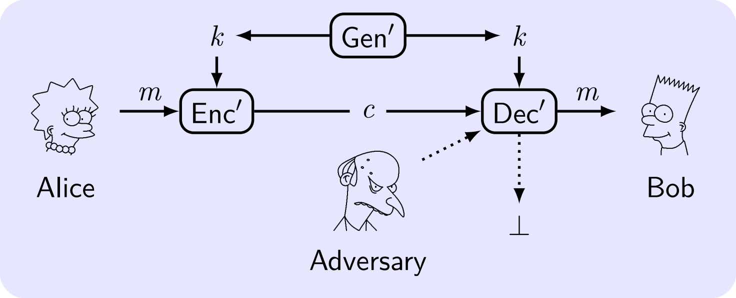
1. $k \leftarrow \mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given input 1^n and oracle access $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ and $\mathcal{A}^{\mathsf{Dec}_k(\cdot)}$, outputs m_0, m_1 of the same length.
3. a random bit $b \leftarrow \{0,1\}$ is chosen. Then $c \leftarrow \mathsf{Enc}_k(m_b)$ is given to $\mathcal{A}$.
4. $\mathcal{A}$ continues to have oracle access except for c, outputs b'.
5. If $b' = b$, $\mathcal{A}$ succeeded $\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi} = 1$, otherwise 0.

Definition 18

Π has **indistinguishable encryptions under a CCA (CCA-secure)** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n).$$

Constructing CCA-Secure Encryption Schemes



Construction 19

$\Pi_E = (\mathsf{Gen}_E, \mathsf{Enc}, \mathsf{Dec})$, $\Pi_M = (\mathsf{Gen}_M, \mathsf{Mac}, \mathsf{Vrfy})$. Π':

- $\mathsf{Gen}'(1^n)$: $k_1 \leftarrow \mathsf{Gen}_E(1^n)$ *and* $k_2 \leftarrow \mathsf{Gen}_M(1^n)$.
- $\mathsf{Enc}'_{k_1,k_2}(m)$: $c \leftarrow \mathsf{Enc}_{k_1}(m)$, $t \leftarrow \mathsf{Mac}_{k_2}(c)$ *and output* $\langle c, t\rangle$.
- $\mathsf{Dec}'_{k_1,k_2}(\langle c, t\rangle)$: *If* $\mathsf{Vrfy}_{k_2}(c, t) \stackrel{?}{=} 1$, *output* $\mathsf{Dec}_{k_1}(c)$; *otherwise output "failure"* $\perp$.

Proof of CCA-Secure Encryption Schemes

Theorem 20

If Π_E is a CPA-secure private-key encryption scheme and Π_M is a secure MAC with unique tags, then Construction Π' is CCA-secure.

Idea: The decryption oracle is useless. CCA $=$ CPA $+$ MAC.

Proof.

VQ: $\mathcal{A}$ submits a "new" query to oracle Dec' and $\mathsf{Vrfy} = 1$.

$$\Pr[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi'}(n) = 1] \leq \Pr[\mathsf{VQ}] + \Pr[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{\mathsf{VQ}}]$$

We need to prove the following claims.

1. $\Pr[\mathsf{VQ}]$ is negligible.
2. $\Pr[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{\mathsf{VQ}}] \leq \frac{1}{2} + \mathsf{negl}(n)$.

□

Proof of "$\Pr[\text{VQ}]$ is negligible"

Idea: Reduce $\mathcal{A}_M$ (attacking Π_M with an oracle $\mathsf{Mac}_{k_2}(\cdot)$) to $\mathcal{A}$.

Proof.

- $\mathcal{A}_M$ chooses $i \leftarrow \{1, \ldots, q(n)\}$ *u.a.r.*
- Run $\mathcal{A}$ with the encryption/decryption oracles.
- If the ith decryption oracle query from $\mathcal{A}$ uses a "new" c, output (c, t) and stop.
- $\mathsf{Macforge}_{\mathcal{A}_M, \Pi_M}(n) = 1$ only if VQ occurs.
- $\mathcal{A}_M$ correctly guesses i with probability $1/q(n)$.

$$\Pr[\mathsf{Macforge}_{\mathcal{A}_M, \Pi_M}(n) = 1] \geq \Pr[\mathsf{VQ}]/q(n).$$

$\square$

Proof of “$\Pr[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{\mathsf{VQ}}] \leq \frac{1}{2} + \mathsf{negl}(n)$”

Idea: Reduce $\mathcal{A}_E$ (attacking Π_E with an oracle $\mathsf{Enc}_{k_1}(\cdot)$) to $\mathcal{A}$.

Proof.

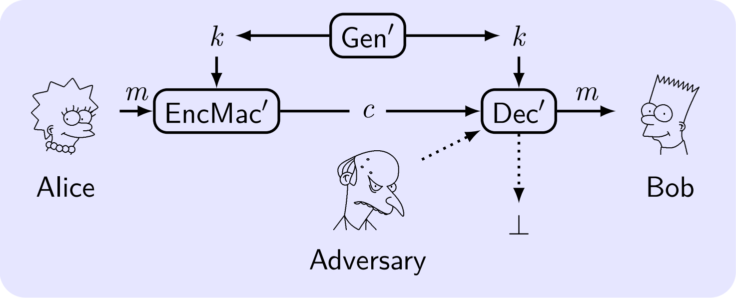
- Run $\mathcal{A}$ with the encryption/decryption oracles.
- Run $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A}_E,\Pi_E}$ as $\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi'}$.
- $\mathcal{A}_E$ outputs the same b' that is output by $\mathcal{A}$.
- $\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A}_E,\Pi_E}(n) = 1 \wedge \overline{\mathsf{VQ}}] = \Pr[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{\mathsf{VQ}}]$ unless VQ occurs.

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A}_E,\Pi_E}(n) = 1] \geq \Pr[\mathsf{PrivK}^{\mathsf{cca}}_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{\mathsf{VQ}}].$$

$\square$

Content

Message Transmission Scheme



- **Key-generation** algorithm outputs $k \leftarrow \mathsf{Gen}'(1^n)$.
  $k = (k_1, k_2)$. $k_1 \leftarrow \mathsf{Gen}_E(1^n)$, $k_2 \leftarrow \mathsf{Gen}_M(1^n)$.
- **Message transmission** algorithm is derived from $\mathsf{Enc}_{k_1}(\cdot)$ and $\mathsf{Mac}_{k_2}(\cdot)$, outputs $c \leftarrow \mathsf{EncMac}'_{k_1,k_2}(m)$.
- **Decryption** algorithm is derived from $\mathsf{Dec}_{k_1}(\cdot)$ and $\mathsf{Vrfy}_{k_2}(\cdot)$, outputs $m \leftarrow \mathsf{Dec}'_{k_1,k_2}(c)$ or $\perp$.
- **Correctness requirement**: $\mathsf{Dec}'_{k_1,k_2}(\mathsf{EncMac}'_{k_1,k_2}(m)) = m$.

Defining Secure Message Transmission

The secure message transmission experiment $\mathsf{Auth}_{\mathcal{A},\Pi'}(n)$:

1. $k = (k_1, k_2) \leftarrow \mathsf{Gen}'(1^n)$.
2. $\mathcal{A}$ is given input 1^n and oracle access to EncMac'_k, and outputs $c \leftarrow \mathsf{EncMac}'_k(m)$.
3. $m := \mathsf{Dec}'_k(c)$. $\mathsf{Auth}_{\mathcal{A},\Pi'}(n) = 1 \iff m \neq \perp \wedge\ m \notin \mathcal{Q}$.

Definition 21

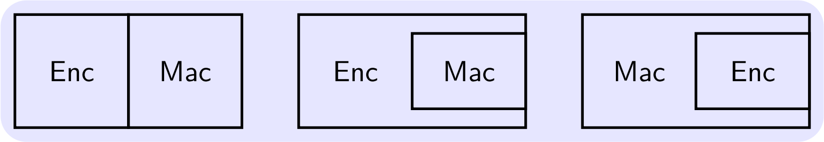
Π' achieves **authenticated communication** if $\forall$ PPT $\mathcal{A}$, $\exists$ negl such that

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

Definition 22

Π' is **secure** if it is both CCA-secure and also achieves authenticated communication.

Combining Encryption and Authentication



- **Encrypt-and-authenticate**:

$$c \leftarrow \mathsf{Enc}_{k_1}(m),\ t \leftarrow \mathsf{Mac}_{k_2}(m).$$

- **Authenticate-then-encrypt**:

$$t \leftarrow \mathsf{Mac}_{k_2}(m),\ c \leftarrow \mathsf{Enc}_{k_1}(m\|t).$$

- **Encrypt-then-authenticate**:

$$c \leftarrow \mathsf{Enc}_{k_1}(m),\ t \leftarrow \mathsf{Mac}_{k_2}(c).$$

Analyzing Security of Combinations

- **All-or-nothing**: Reject any combination for which there exists even a single counterexample is insecure.
- **Encrypt-and-authenticate**: $\mathsf{Mac}'_k(m) = (m, \mathsf{Mac}_k(m))$.
- **Authenticate-then-encrypt**:
  - Trans : $0 \to 00; 1 \to 10/01$, Enc′ uses CRT mode, $c = \mathsf{Enc}'(\mathsf{Trans}(m\|\mathsf{Mac}(m)))$.
  - Flip the first two bits of the second block of c and verify whether the ciphertext is valid.
  - If valid, the first bit of message is 1; otherwise 0.
  - For any MAC, this is not CCA-secure.
- **Encrypt-then-authenticate**:
  Decryption: If $\mathsf{Vrfy}(\cdot) = 1$, then $\mathsf{Dec}(\cdot)$; otherwise output $\perp$.

Theorem 23

Π_E is CPA-secure and Π_E is a secure MAC with unique tages, Π' deriving from encrypt-then-authenticate approach is secure.

Remarks on Secure Message Transmission

- Authentication may leak the message.
- Secure message transmission implies CCA-security. The opposite direction is not necessarily true.
- Different security goals should always use different keys.

Summary

- Integrity/authentication by MAC.
- adaptive CMA, replay attack, birthday attack.
- Existential unforgeability, collision resistance, CCA-secure, authenticated communication, secure message transmission.
- PRF, CBC-MAC, CRHF, Merkle-Damgård transform, NMAC, HMAC, encrypt-then-authenticate.