Indistinguishability

Roughly, distributions / r.v.'s D_1, D_2 are <u>indistinguishable</u> if for all adversaries A,

$$Adv(A) := \Pr[A(D_1) \Rightarrow 1] - \Pr[A(D_2) \Rightarrow 1] \leq \varepsilon(\epsilon)$$

for some small ε.

$D_1 \approx_s D_2$ when A is ~~too~~ <u>unbounded</u>.

$D_1 \approx_c D_2$ when A is <u>bounded</u>

(t, ε) - Indistinguishable

One-wayness

(X, Y) correlated r.v.'s

$(X, f(X))$

We say f is O.W. on X if $\Pr[A(f(X)) = x] \leq \varepsilon$ for every A and some small ε.

$\mathcal{A}(f(x)) \Rightarrow x$

if f is injective

$\mathcal{A}(f(x)) \Rightarrow x'$ s.t. $f(x) = f(x')$

in general

* One-wayness, indistinguishability are fundamental notions in cryptography.

Collision-resistance

$\mathcal{A} \Rightarrow (x, y)$ s.t. $f(x) = f(y)$, $x \neq y$.

and collision-resistance!

Hash function

$H : \{0,1\}^* \to \{0,1\}^n$

SHA family SHA-3

Merkle-Damgård paradigm

Start with a fixed-length compression function

$\Rightarrow h : \{0,1\}^{n+k} \to \{0,1\}^n$

want to construct H

this h is often built out of a blockcipher

Aside: blockciphers and PRF

$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$

AES

typical assumption about blockciphers:

PRF

$$Adv^{prf}_E(A) = Pr[A^{E_k(\cdot)} \Rightarrow 1] - Pr[A^{\$} \Rightarrow 1]$$

Davies-Meyer

$h : \{0,1\}^{k+n} \to \{0,1\}^n$

$h(k \| x) = E_k(x) \oplus k$ ← check this w/ book

$H: \{0,1\}^* \to \{0,1\}^n$ $\quad h: \{0,1\}^{n+k} \to \{0,1\}^n$

Merkle-Damgard

want to show if h is collision-resistant then H is collision resistant

need "strengthening": append length of message.

length-extension attack.

Given $H(x)$ for unknown x, can compute $H(x \| t)$ for any t.

Key derivation

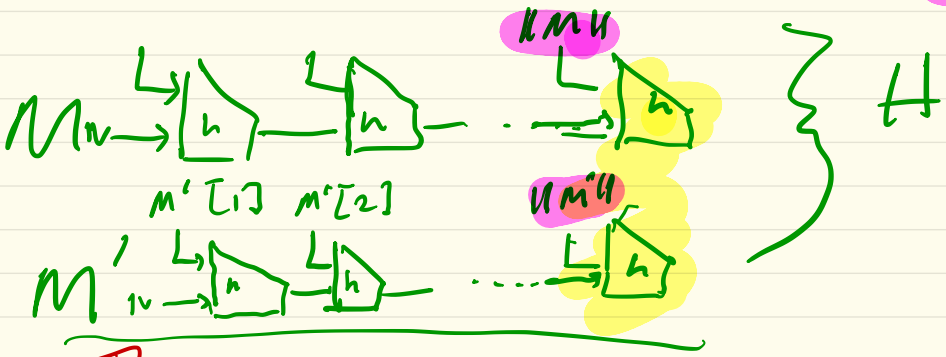
$H: \{0,1\}^* \to \{0,1\}^\kappa$

and a distribution X with "entropy"

$H(X) \approx_c \$$

Proof of MD

SHA1 256 bits output



Case 1: $||M|| \neq ||M'||$

Case 2: $||M|| = ||M'||$

idea:
line up the computations of little h and "work backwards"

- How to construct H out of h ✓
- How to construct h out of a blockcipher
  - Davies-Meyer

$$h(x||y) = E_x(y) \oplus y$$

PRF

blockcipher: family of permutations indexed by a key.

$E: Keys \times Dom \to Dom$

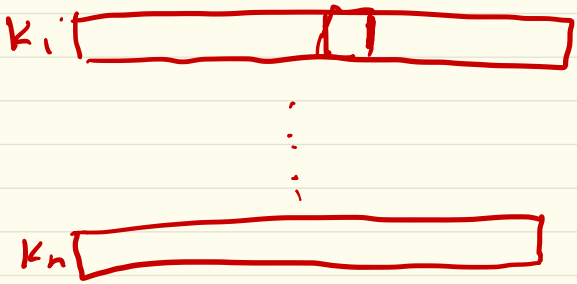
DES
AES

→ Conventional security assumption about blockciphers:
pseudorandom function

AES 128 key length
128 block length

→ stronger security "assumption"

ideal cipher model
- blockcipher is modeled as an independent uniform permutation on every key.



Constructing MACs from hash functions.

⇒ What are MACs?

$MAC = (Gen, \tilde{T}, V)$

- key-generation algorithm Gen

$k \xleftarrow{R} K$

f is O.W.

If $P_r[A(f(x)) \Rightarrow x' \text{ s.t. } f(x) = f(x')] \leq \varepsilon$.

(r.v. uniform on domain)

f is CR. if

$P_r[A \Rightarrow (x, y) \text{ s.t. } x \neq y, f(x) = f(y)] \leq \varepsilon$.

CR $\Rightarrow$ O.W.

If f is compressing

O.W. $\Rightarrow$ CR? No

How to prove a non-implication

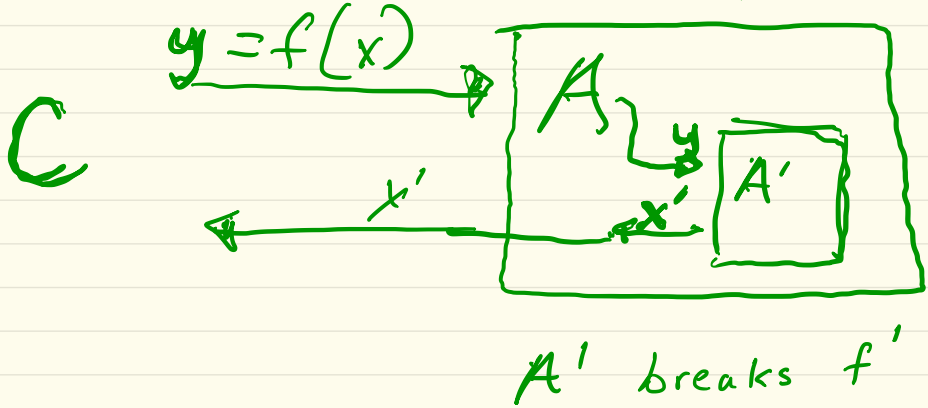
Suppose O.W. function f exists. Construct f' s.t:

* f' is O.W. * f' is not CR.

$$f'(x) = \begin{cases} 0 & \text{if } x \in \{1,2\} \\ f(x) & \text{o.w.} \end{cases}$$

→ collision on 1, 2.

→ W.T.S. if you can break one-wayness of f' then you can break one-wayness of f.



A' breaks f'

if A' succeeds: $f'(x') = y$

W.T.S.

$f(x') = y$

To construct f':

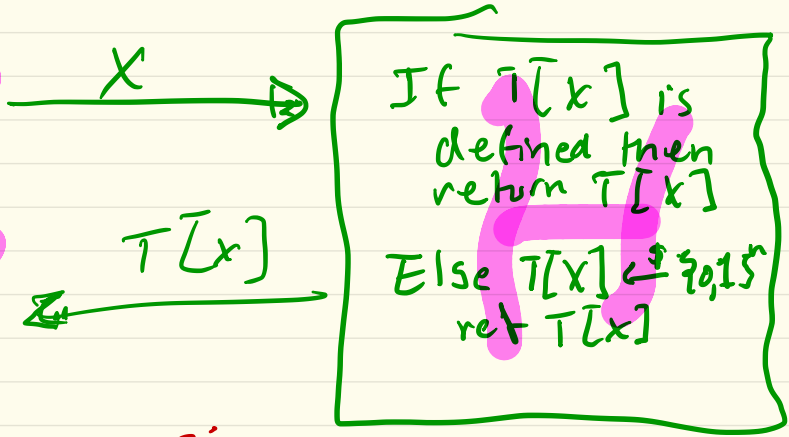
Find y st. $|f^{-1}(y)|$ is "small"

- If no such y exists then f is not O.W.

Define $f'(x) = \begin{cases} y & \text{if } x=1 \\ f(x) & \text{O.W.} \end{cases}$

Random Oracle Model

- Model of computation where every algorithm (scheme algs, adversaries...) have access to the same random function.



Warm-up:

How to design PRFs in the R.O. model?

Note: $F: Keys \times Dom \to Rng$ is a PRF if $\forall A$

$$Pr[A^{F_k(\cdot)} \Rightarrow 1] - Pr[A^{\$(\cdot)} \Rightarrow 1]$$

is small.

where the 1st probability is over K and $\$$ is a random function from Dom to Rng.

RO model guarantees no "black box" attacks

"prepend" construction

$$F_k^{pre}(x) = H(k \| x)$$

$$F_k^{post}(x) = H(x \| k)$$

very important to understand proof!

$$F_k^{envelope}(x) = (k \| x \| k)$$

Claim. F^{pre} is a PRF in the RO model.

Intuition:

two oracles

$F^{pre}, H \simeq \$, H$

$x \xleftarrow{\$} D$

pseudorandom function

PRFs in the RO model

Attack model

Recall PRF def in standard model: $F : \mathcal{K} \times D \to R$

Real Game:
$K \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} A^{F_K(\cdot)}$
ret b

Rand Game:
$f \xleftarrow{\$} \text{Func}[D, R]$
$b \xleftarrow{\$} A^{f(\cdot)}$
ret b

$$Adv^{prf}_{F}(A) = Pr[\text{Real} \Rightarrow 1] - Pr[\text{Rand} \Rightarrow 1]$$

Note: Rand Game can be implemented efficiently via "lazy sampling"

Attack Model in RO model

$Exp[b] =$
$O \xleftarrow{\$} \text{Func}[D, R]$
If $b = 1$ $K \xleftarrow{\$} \mathcal{K}$; $f \leftarrow F_K^O(\cdot)$
If $b = 0$ $f \xleftarrow{\$} \text{Func}[X, Y]$
$d \xleftarrow{\$} A^{O(\cdot), f(\cdot)}$
ret $(d = b)$

$Exp[1]$ is "real"

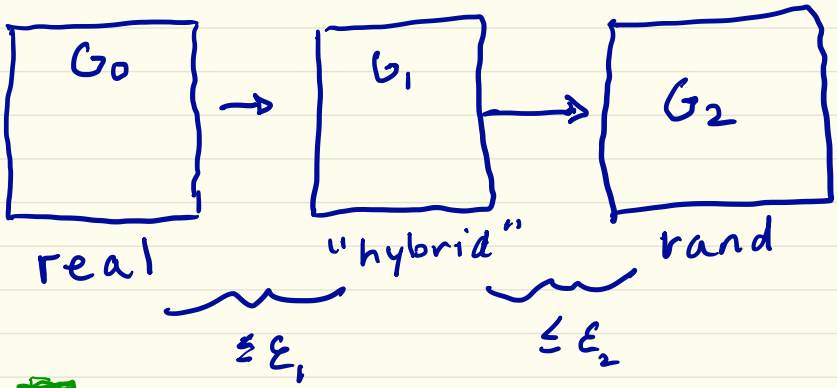
$Exp[0]$ is "rand"

A makes distinct queries

To prove: In the RO PRF wlog attack model

$Adv^{prf}_{F,Fpre}(A)$ is small.

We're going to use "game-playing" or "game hopping"



G_0/G_1

Initialize empty array Map: $D \to R$

$K \xleftarrow{\$} K$

On F_{pre} query x do:

$y \xleftarrow{\$} R$

If $K \| x \in Dom(Map)$ then bad ← true

$y \leftarrow Map(K \| x)$

$Map(K \| x) \leftarrow y$

ret y

G_1 / G_2

On $\mathcal{U}$-query m do:

$y \xleftarrow{\$} R$

if $m \in \text{Dom}(map)$ then {

bad $\leftarrow$ true

$y \leftarrow map(m)$ }

$map(m) \leftarrow y$

ret y

$H(K \| \cdot), \; H(\cdot)$

z

$K \| z$

"Fundamental Lemma of Game ..."

Lemma. Let G_0 ... until bad is set. Then

$$Pr[G_0 \Rightarrow 1] - Pr[G_1 \Rightarrow 1] \leq Pr[G_0 \text{ sets bad}]$$

$$\Pr[G_0 \Rightarrow 1] - \Pr[G_1 \Rightarrow 1]$$

$$\leq \Pr[G_0 \text{ sets bad}]$$

$$\leq \frac{q_V = pre}{|\mathcal{K}|} \quad \text{(union bound)}$$

$$\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]$$

$$\leq \Pr[G_1 \text{ sets bad}]$$

$$\leq \frac{q_D}{|\mathcal{K}|}$$

q_D

$MAC: K \times D \to R$

Suppose we have a MAC for short messages (eg a blockcipher)
Extend it to be a MAC for long messages via

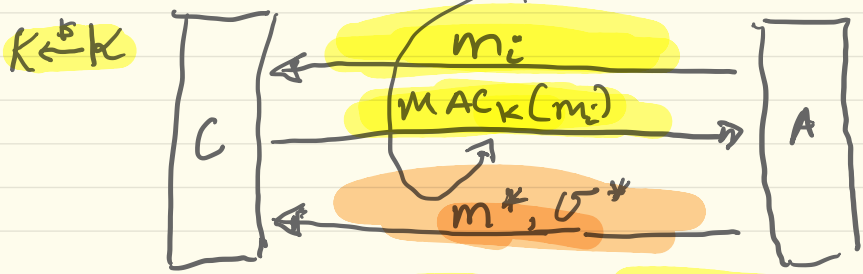
$$MAC'_K(x) = MAC_K(H(x))$$

where $H: \{0,1\}^* \to \{0,1\}^l$ and MAC works on messages of length l

Theorem. If MAC is a secure MAC (eg a PRF) and H is CR then MAC' is a secure MAC.

[Recall: Secure MAC

UF-CMA



Output 1 ("WIN") if
- $MAC_K(m^*) = \sigma^*$
- $m^* \neq m_i \ \forall i$

} called a forgery

Adv

WTS

$\mathrm{Adv}^{\text{uf-cma}}_{\text{MAC}}(A)$ is small

$\forall$ efficient A

Game G_1:

On forgery attempt (m^*, σ^*):

$y \leftarrow H(m^*)$

If $\exists i$ s.t. $H(m_i) = H(m^*)$

then $y \xleftarrow{*} \{0,1\}^l$

If $MAC_K(y) = \sigma^*$

Ret 1

Else ret 0.

$(m^*, \sigma^*) \xleftarrow{\$} A^{\mathcal{O}_{tag}(K, \cdot)}$

Oracle $\mathcal{O}_{tag}(x)$

$x_i \leftarrow x$
$y_i \leftarrow \text{Eval}(K, x)$
ret y_i

Proof: by sequence of games

Game G_0: UF-CMA game
Set BAD if $\exists i$ s.t. $H(m_i) = H(m^*)$

Game G_1: Return 0 when BAD is set!

$\Pr[G_0 \Rightarrow 1] \leq \Pr[G_1 \Rightarrow 1] + \Pr[G_0 \text{ sets BAD}]$

$\text{Adv}^{\text{uf-cma}}_{\text{MAC}}(A)$

We give a CR adversary A_H that finds a collision in H w/ P G_0 sets BAD.

Adversary A_{H}:

$K \xleftarrow{\$} \{0,1\}^\lambda$

Run A:

When A makes query m_i do:

$\sigma_i \leftarrow \text{Eval}(K, m_i)$

Until A outputs (m^*, σ^*)

If $\exists i$ s.t. $H(m^*) = H(m_i)$

then RETURN (m^*, m_i)

ret $\perp$

$\Pr[A_H \text{ wins}] = \Pr[\text{G. sets BAD}]$

QUESTION:
What is the goal in defining G_1?

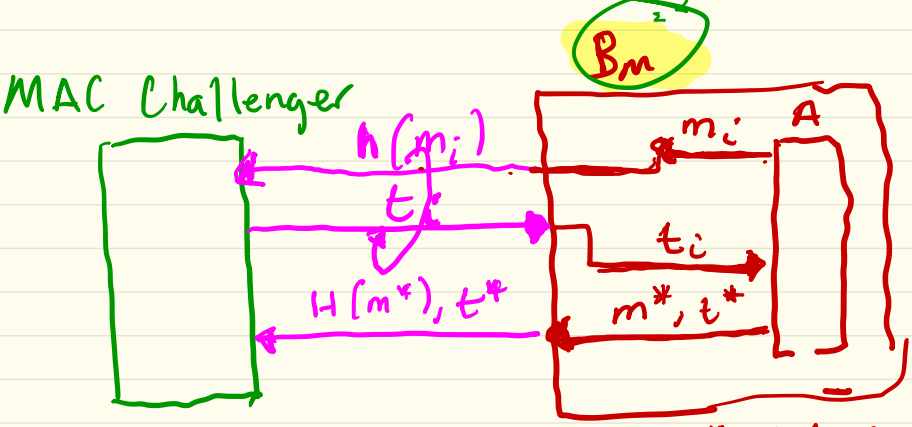
ANSWER:
We want to construct an adversary against the base (i.e. stacking) MAC scheme such that its advantage is at least as large as the probability G_1 outputs 1.

WTS: $\exists B_m$ s.t.

$$Adv^{uf\text{-}cma}_{MAC}(B_m) \geq Pr[G_1 \Rightarrow 1]$$

QUESTION:
Why?

QUESTION:
What is G_1 and what is B_m?

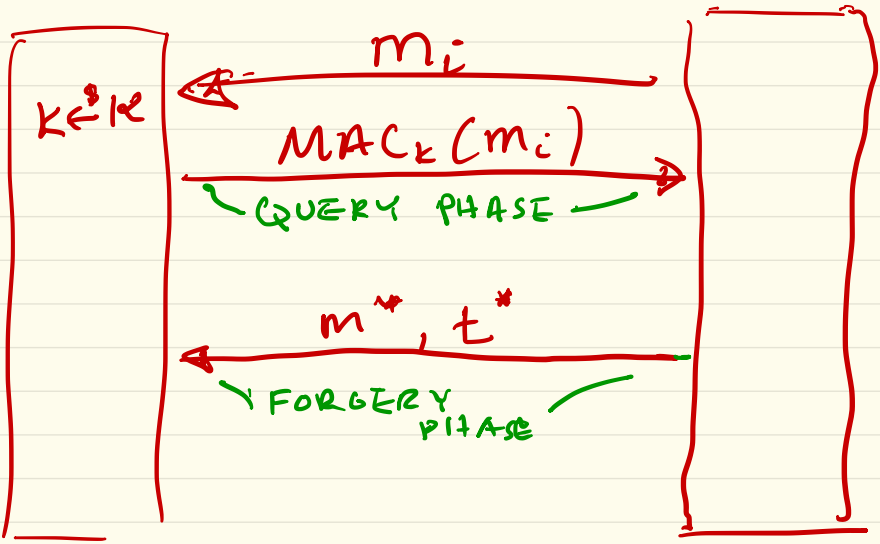


A: attacking BIG MAC
B_m: attacking little MAC

$t^* = MAC_k(H(m^*))$

UF-CMA GAME

$MAC : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$



{ return 1 if:
$m^* \neq m_i \ \forall i$ (*)
$t^* = MAC_k(m^*)$ }

m^*, t^* is said to be a forgery if this holds.

$$Adv^{uf\text{-}cma}_{MAC}(A) =$$

$$Pr[\text{UF-CMA GAME} \Rightarrow 1]$$

executed with A and MAC

MAC is UF-CMA secure if

$\forall$ efficient A,

$Adv^{uf\text{-}cma}_{MAC}(A)$ is small.

A message authentication code is a function family

$MAC = (Keygen, Tag)$

or

$MAC : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$

HMAC

- How to MAC using a hash function?

preliminary ideas:

* append the key

$H(m \| k)$

- offline attack!

* prepend the key

$H(k \| m)$

?

* envelope

$H(k \| m \| k)$

Offline-Online attack against append-the-key

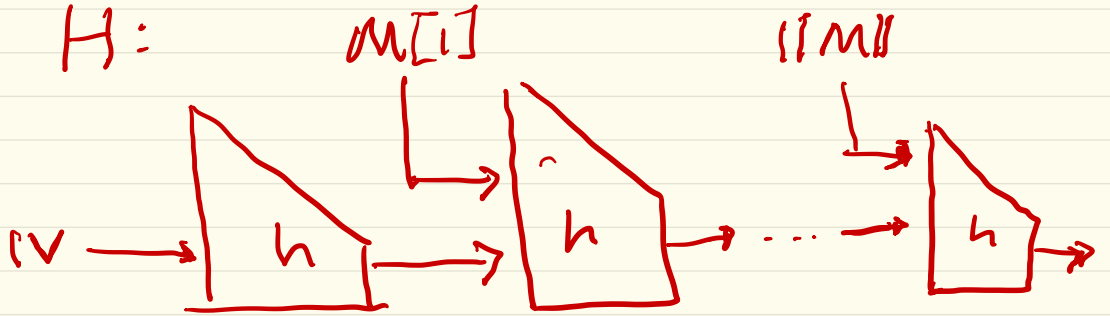
Offline. Find m_0, m_1 s.t. $H(m_0) = H(m_1)$

Online

$t \leftarrow TAG(m_0)$

$ret(m_1, t)$

Merkle - Damgard



Offline attack on H: Birthday Attack

For output of size N, finds a collision in expected

$O(\sqrt{N})$ time.

Alg Birthday Attack

- $s \leftarrow \lceil 2\sqrt{N} \rceil + 1$
- Generate s uniform random messages $m_0, \ldots, m_s$
- Compute $x_i \leftarrow H(m_i)\ \forall i$
- Look for $i \neq j$ st. $H(m_i) = H(m_j)$

Output m_i, m_j

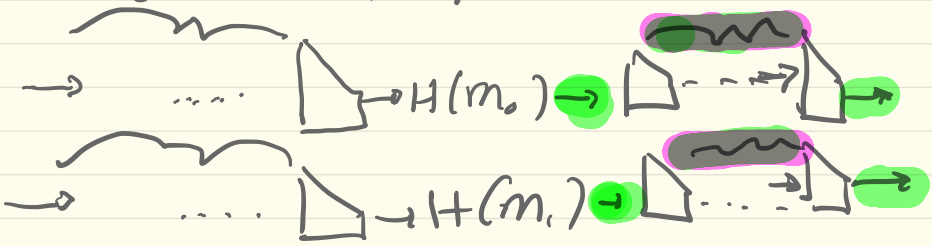
$H : [M] \to [N]$

Theorem. Let $m_1, \ldots, m_s$ be the uniformly random messages sampled. Assume $M \geq 100 \cdot N$. Then w.p. at least $1/2$ $\exists i, j$ s.t. $H(m_i) = H(m_j)$.

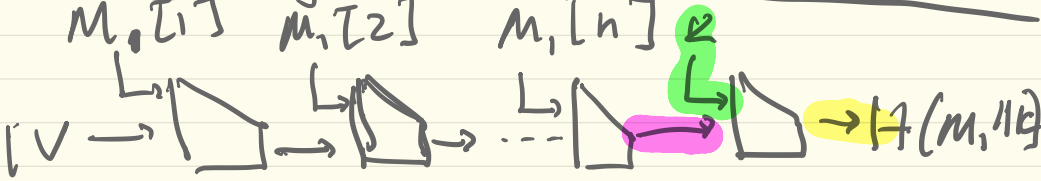
"proof": If x_i are random independent, a collision among the x_i will occur w.p. $1 - e^{-s(s-1)/2N} \geq 3/4$.

- $x_1, \ldots, x_m$ are distinct w.p. $4/5$ b/c $|M| \geq 100 \cdot |N|$

Claim. If $H(m_0) = H(m_1)$ then $H(m_0 \| k) = H(m_1 \| k)$. (assuming H is M.D.)



Hashing of $M_0 \| K$

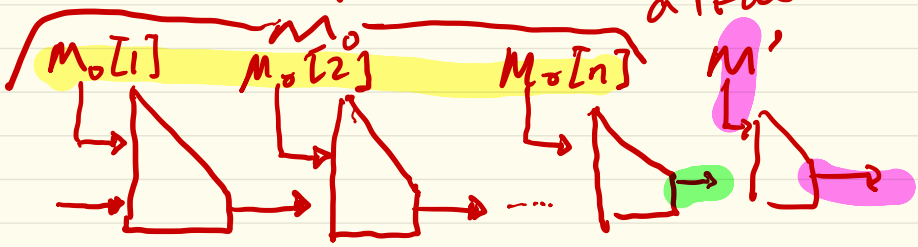


Hashing of $M_1 \| K$

Prepend-the-Key

$H(K \| M_0) \rightarrow H(K \| M_0 \| M')$

length extension attack



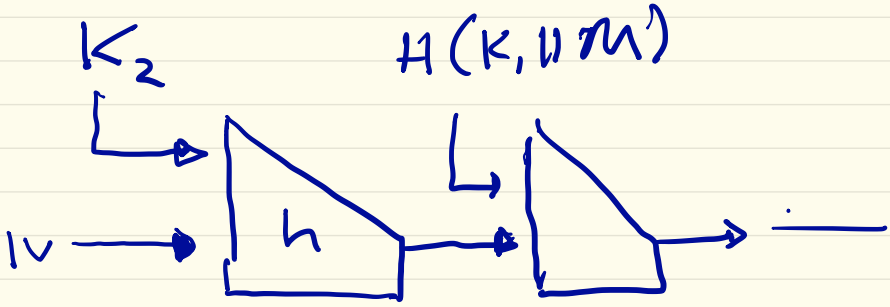
don't know M_0, just $H(M_0)$

Claim. Can compute $H(M_0 \| M')$ for any M'.

Prefix-Free MAC: no message allowed to be proper prefix of another message.

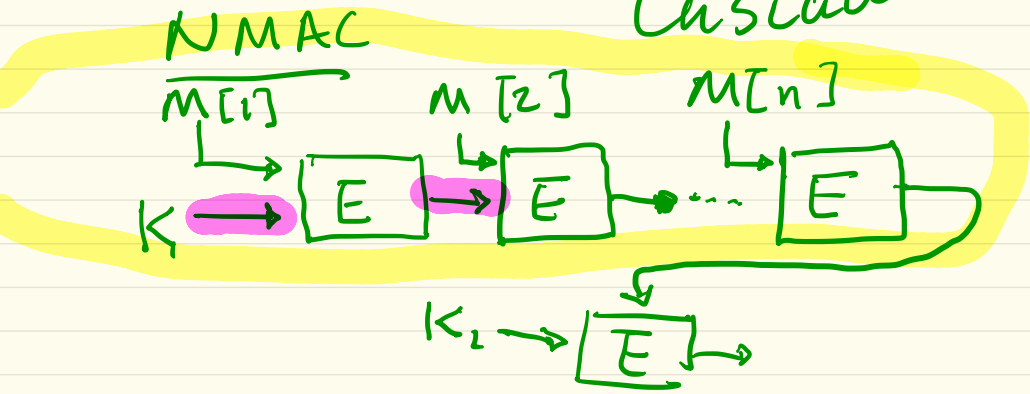
Two-key Nest:

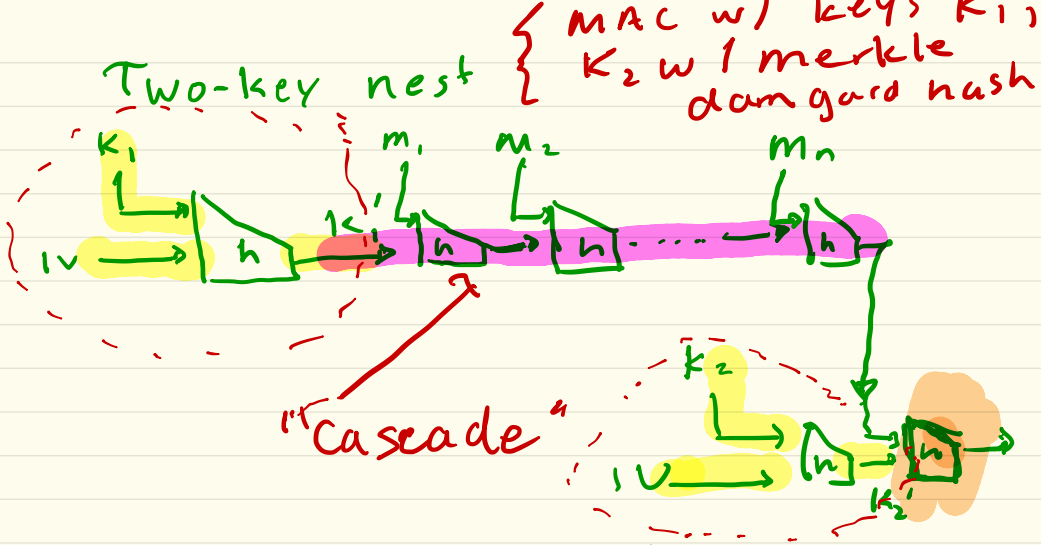
$$H(K_2 \| H(K_1 \| m))$$



related to HMAC.

Cascade.





How to analyze two-key nest?

① replace k_1', k_2' with $ by PRF sec of h_{top}

② this is "NMAC" applied to h_{bot}

$$h_{bot}(k, m) = h(k, m)$$

$$h_{top}(k, m) = h(m, k)$$

What is NMAC?

* Cascade-then-PRF

$EF((k_1, k_2), m) := F(k_2, PF(k_1, m))$

Theorem. If PF is extendable and prefix-free PRF, then EF is a secure PRF. and F is a PRF

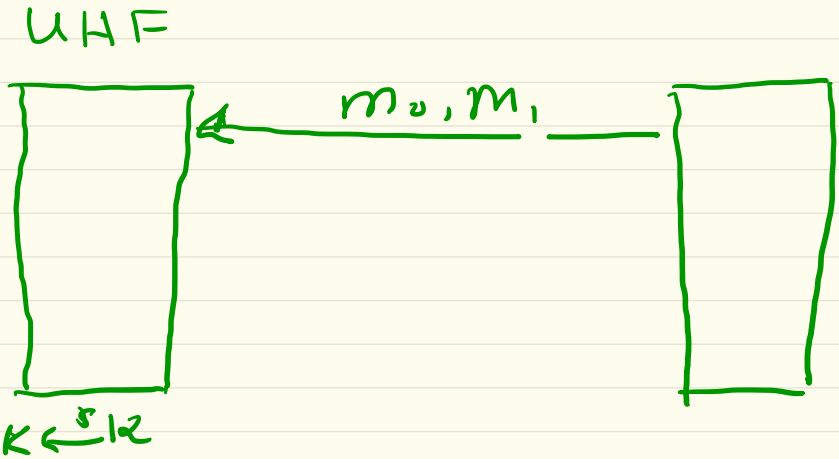
- Prefix-free MAC:
  restricted game in which adversary cannot query any m, m' s.t. m is a prefix of m'.

(block-wise prefixes.)

Extendable:

If $PF(k, x) = PF(k, y)$
then $\forall a$
$PF(k, x \| a) = PF(k, y \| a)$

Theorem: If PF is extendable and prefix-free PRF then PF is a computational UHF.



WIN if $H_K(m_0) = H_K(m_1)$

H is a computational UHF if $\forall$ efficient A the probability that A wins is small.

Theorem. If H is a computational UHF and E is a PRF then the "hash-then-PRF" MAC $w/$ H and E is UF-CMA.

Question: Why can't we just assume a MAC here??

Answer: We can't reveal info about the hashes.

Theorem. Cascade is extendable and prefix-free PRF.

$F_{bot}(k, m) = h(k, m)$

$F_{top}(k, m) = h(m, k)$

→ used to reduce HMAC to NMAC

Cascade is prefix-free & extendable

→ prefix-free & extendable PRF is a cUHF

$h(a) = h(b)$

⇓

$h(a \| y) = h(b \| x)$

↓

PRF (cUHF) is UF-CMA

GGM for variable length inputs & n-ary trees ← GGM for variable length inputs ← GGM

prefix free

prefix free

Cascade.

input: k, $m = (a_1, \ldots, a_v)$

output: a tag

$t \leftarrow k$
for $i = 1$ to v do:
  $t \leftarrow F(t, a_i)$
output t

If $a_1, \ldots, a_v$ are bits
call this
bitwise cascade

Goldreich-Goldwasser-Micali PRF.

Suppose we have a PRG

$$G: \{0,1\}^n \to \{0,1\}^{2n}$$

(length-doubling)

$$G(R) \approx_c U$$

seed

where R is uniform on $\{0,1\}^n$

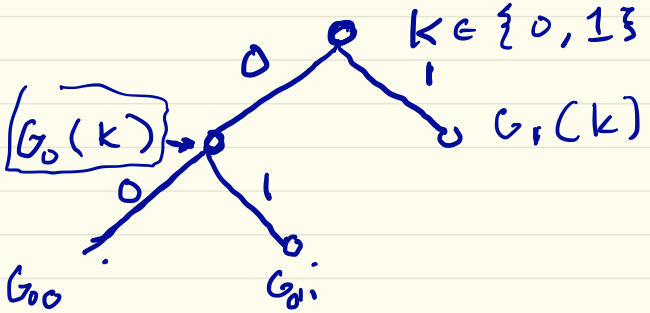
U is uniform on $\{0,1\}^{2n}$

How to construct a PRF

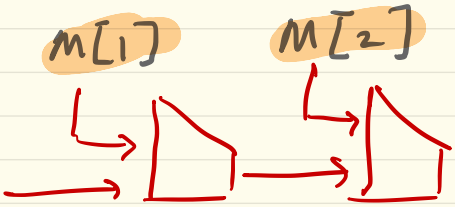
with domain $\{0,1\}^m$ $\quad G(R)_0$

range $\{0,1\}^n$ $\quad G(R)_1$

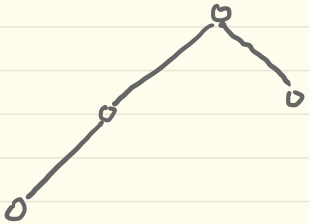
Idea: key is a SEED



$G^*(k) = F_k(0) \| F_k(1)$



Apply GGM to G^*



$F_{f_k(0)}(0)$

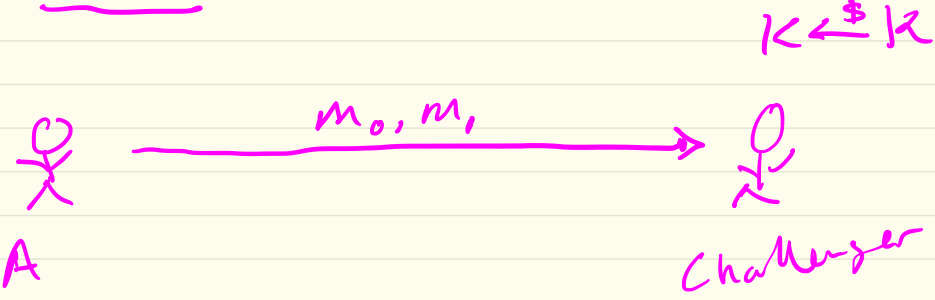
Claim. If GGM is prefix-free (fixed-length) PRF then cascade (bit-wise) is prefix-free (fixed length) PRF.

- This is how cascade is connected to GGM

- how does cascade connect to NMAC?

prefix-free PRF + extendable $\Longrightarrow$ cUHF

Recall



WIN if $H_K(m_0) = H_K(m_1)$

For all efficient A, $\Pr[\text{WIN}]$ is small.

0 01 00 010

Theorem. If F is an extendable prefix-free PRF then it is a computational UbtF.

proof: Suppose A is a cUbtF adversary attacking F. Consider PRF adversary B

Adversary $B^{\cdot}$:

$(m_0, m_1) \xleftarrow{\$} A$

$y_0 \leftarrow F_n(m_0)$

$y_1 \leftarrow F_n(m_1)$

If $y_0 = y_1$ ret 1

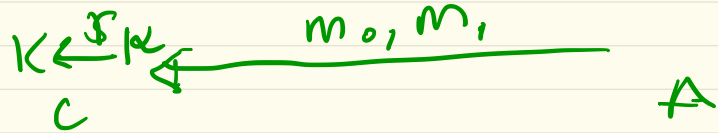
Else ret 0.

swap m_0, m_1 → $m_0 \| a$, $m_1 \| a$ st.

neither is a prefix of the other.

cUHF-then-PRF PRF(cUHF)

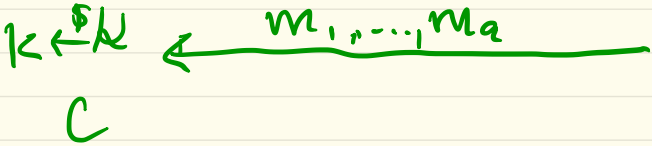
cUHF (single-query)



WIN if $H_K(m_0) = H_K(m_1)$

H is called a cUHF if
∀ efficient A, Pr[WIN] is small.

multi-query cUHF



WIN if $\exists i \neq j$ s.t. $H_K(m_i) = H_K(m_j)$

H if called a multi-query cUHF if ∀ efficient A,
Pr[WIN] is small.

Theorem. Single-query cUHF ⇒ multi-query cUHF

at most q queries.

proof. Let A be a multi-query cUHF adversary. Consider the single-query cUHF adversary

Adversary A'
$i, j \xleftarrow{R} [q]$
$(m_1, \ldots, m_q) \xleftarrow{R} A$
return (m_i, m_j)

Fix an arbitrary coin sequence on which A outputs a collision. If A' runs A on that coin sequence, then it has $\frac{1}{q^2}$ probability of producing a collision. Thus

says Joseph $\leftarrow$ $\mathrm{Adv}(A) \leq 2q^2 \mathrm{Adv}(A')$

$\Pr[A \text{ outputs collision} \wedge A' \text{ is right for } i,j]$

$= \Pr[A \text{ outputs collision}] \cdot \Pr[A' \text{ is right}]$

$\widetilde{F} = H \circ F$

Construction

$H : K_H \times D \to R$ ← cUHF

$F : K_F \times R \to R'$ ← PRF

Define $\widetilde{F} : (K_H \times K_F) \times D \to R'$

$$\widetilde{F}_{K_H, K_F}(x) = F_{K_F}(H_{K_H}(x))$$

← PRF

This is the

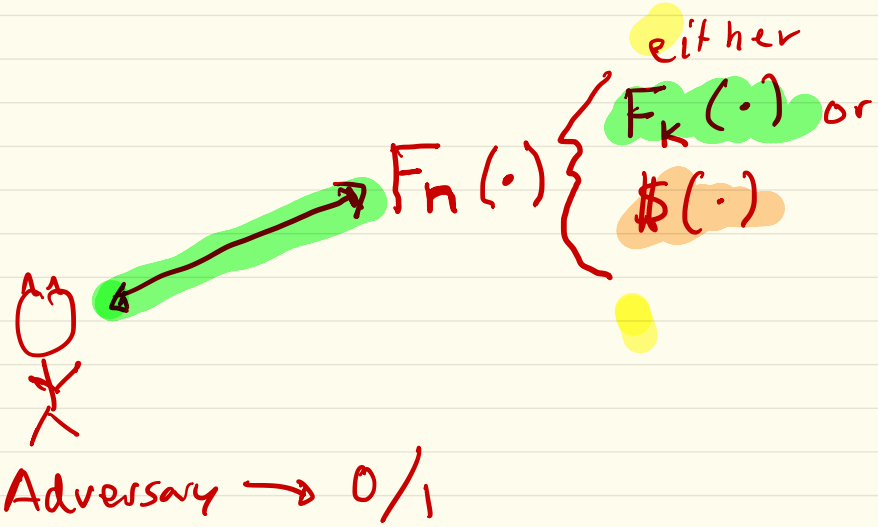
CMHF-then-PRF construction

Note: Compared to Nash-then-MAC, we

* weaken assumption on H
* strengthen assumption on F.
* The conclusion is stronger

Theorem. If H is cUHF and F is a PRF then $\widetilde{F}$ is a PRF.

$F : \mathcal{K} \times D \to R$



$$Adv(A) = Pr[REAL_F^A \Rightarrow 1] - Pr[RAND_F^A \Rightarrow 1]$$

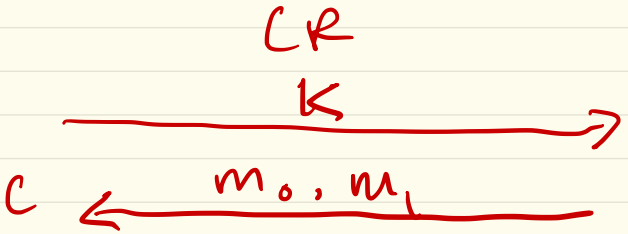
Computational universal hash function

collision resistant

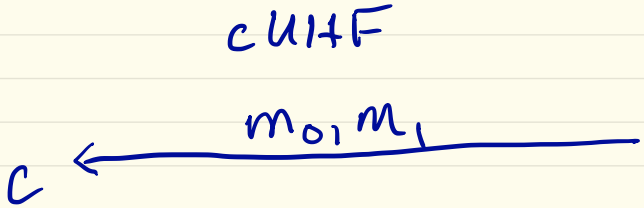
cUHF vs CR hash

Main difference:

CR adversary LETS THE KEY



WIN if $H_k(m_0) = H_k(m_1)$



WIN if $H_k(m_0) = H_k(m_1)$

CR much weaker!

practical CR hash fns are unkeyed

Assume A makes distinct queries

proof of PRF(cMAC)

Let A be a PRF adversary against $\tilde{F}$. Consider a sequence of games played by A:

G_0: REAL game

$K_1 \xleftarrow{\$} \mathcal{K}_H \quad K_2 \xleftarrow{\$} \mathcal{K}_F$

On i-th query m_i do:

$x_i \leftarrow H_{K_1}(m_i)$

$t_i \leftarrow F_{K_2}(x_i)$

return t_i

G_1: $K_1 \xleftarrow{\$} \mathcal{K}_H \quad t_1', \ldots, t_q' \xleftarrow{\$} R'$

On i-th query m_i do:

$x_i \leftarrow H_{K_H}(m_i)$

$t_i \leftarrow t_i'$

If $x_i = x_j$ for some i, j

BAD $\leftarrow$ true $\quad t_i \leftarrow t_j'$

return t_i

G_2

G_2 is RAND game

Claim.

$$\Pr[G_0^A \Rightarrow 1] - \Pr[G_1^A \Rightarrow 1] \leq Adv_F^{prf}(B_F)$$

actually equality (Joseph)

for some PRF adversary B_F.

proof.

Adversary $B_F^{Fn(\cdot)}$

$K_1 \xleftarrow{\$} \{0,1\}^k$

Run A

When A makes query m_i:

$x_i \leftarrow H_{K_1}(i \| m_i)$

$y \leftarrow Fn(x_i)$

ret y

Until A outputs b

Ret b

Claim.

$$\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1] \leq \Pr[G_1 \text{ sets BAD}]$$

proof: Fundamental Lemma

Claim. $\Pr[G_1 \text{ sets BAD}] \leq Adv_H^{muhf}(B_H)$

for some B_H muHF adversary.

Want to construct B_{1+} that breaks multi-query CUIF w.p. at least probability BAD is set.

Adversary B_{1+}

Run A

On query m_i,
$t_i \xleftarrow{\$} R$ [could also sample up front?]
ret t_i

until A outputs b
return $(m_1, \ldots, m_q)$

Observation: B_{1+} simulates G_1 for A up to the point BAD would be set.

Observation: When BAD would be set, B_{1+} wins the MCUIF game.

$Adv(A) =$

Hence, $P_1[G_0 \Rightarrow 1] - P_r[G_2 \Rightarrow 1]$
$= \leq Adv(B_F) + Adv(B_M)$
(telescoping sum)

GGM PRF

$00 \to 1011$

$G_0(00) = 10$

Given: length doubling PRG

$$G: \{0,1\}^n \to \{0,1\}^{2n}$$

$$G(R) \approx_c S$$

PRG security property

R uniform over $\{0,1\}^n$

S uniform over $\{0,1\}^{2n}$

Want: PRF

$$G^*: \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^n$$

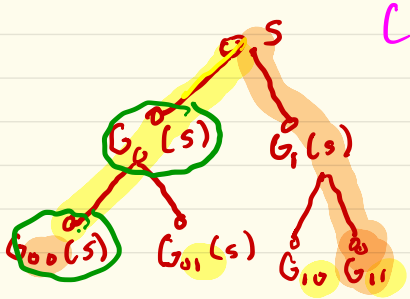
Algorithm $G^*(s, (a_1, \ldots, a_\ell))$: ($s \in \{0,1\}^n$)

$t \leftarrow s$

For $i = 1$ to ℓ do:

$t \leftarrow G_{a_i}(t)$

ret t



Can view GGM PRF as a walk down a tree.

Theorem. GGM is a secure PRF.

Think of the real and random games.

We want to create a sequence of hybrids that move us from the real game to random game.

hybrids: Change level i of the tree to random labels.

hybrid j:

$f \xleftarrow{\$} Func[\{0,1\}^{\leq \ell} \to \{0,1\}^n]$

On query $x = (a_1, \ldots, a_\ell)$

$u \leftarrow (a_1, \ldots, a_j)$

$v \leftarrow (a_{j+1}, \ldots, a_\ell)$

$y \leftarrow G^*(f(u), v)$

Ret y

$P_j = P_j[\text{hybrid } j' \Rightarrow 1]$

We will show a PRG adversary A_G s.t.

$$Adv(A_G) \geq \frac{1}{\ell} |P_\ell - P_0|$$

$$= \frac{1}{\ell} Adv(A_{G^*})$$

Note:

Key for PRF = seed for PRG

In particular we will show that

$$Adv(A_G) = \frac{1}{\ell} \sum_{j=1}^{\ell} p_j - \sum_{j=1}^{\ell} p_{j-1} = p_\ell - p_0$$

Adversary A_G ($\vec{r} = (r_{10}, r_{11}, \ldots, r_{q0}, r_{q1})$)

$w \xleftarrow{R} [\ell]$

Map : $\{0,1\}^* \to \mathbb{Z}_{>0}$

ctr $\leftarrow 0$

On query $x = (a_1, \ldots, a_\ell)$:

$u \leftarrow (a_1, \ldots, a_{w-1})$

$d \leftarrow a_w$

$v \leftarrow (a_{w+1}, \ldots, a_\ell)$

If $u \notin$ Domain (Map)

then

ctr $\leftarrow$ ctr+1

Map[u] $\leftarrow$ ctr

$p \leftarrow$ Map[u]

$y \leftarrow G^*(r_{pd}, v)$

return y

Run A_G*

$W_b :=$ " A_G outputs b

then

$$\begin{aligned}
Adv(A_G) &= P_r[W_1] - P_r[W_0] \\
&= \frac{1}{\ell} \sum_j P_r[W_1 \mid w=j] \\
&\quad - \sum P_r[W_0 \mid w=j] \\
&= \frac{1}{\ell} \sum P_j - \sum P_{j-1} \\
&= \frac{1}{\ell} (P_\ell - P_0) \\
&= \frac{1}{\ell} Adv(A_G^*)
\end{aligned}$$

A_G^* is the assumed adversary against the PRF