Lecture 2

Def (Cryptographic hash function)
A (cryptographic) hash function $H = (Gen_H, Eval_H)$ consists of 2 PPT algo's
- $Gen_H(1^\kappa)$ outputs t that defines $H_t: \{0,1\}^* \to M_t$
- $Eval_H(1^\kappa, t, m)$ computes $H_t(m)$

Def (Collision resistance)
A hash function H is collision-res iff for $\forall$ PPT A $Pr[A(1^\kappa, t) = (x, x') \mid H_t(x) = H_t(x')]$ is negligible.

Signatures w/ unbounded msg space
Given $\Sigma' = (Gen', Sign', Vfy')$ and collision resistant $H: \{0,1\}^* \to M$
- $Gen(1^\kappa)$ outputs $((pk', t), (sk', t)) = (pk, sk)$
- $Sign(sk, m)$: computes $Sign'(sk', H(m))$
- $Vfy((pk', t), m, \sigma) = Vfy'(pk', H(m), \sigma)$

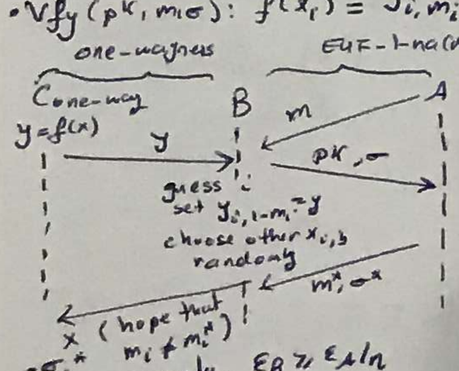
Thm: for every EUF-CMA PPT A runtime t_A and succ. ε_A there exist
- PPT B on Σ''s EUF-CMA sec. in time $t_B \approx t_A$ and succ. ε_B
- a PPT C on collision-res. of H that runs in $t_C \approx t_A$ and succ. ε_C s.t. $\varepsilon_B + \varepsilon_C \geq \varepsilon_A$

Proof
whenever A successfully forges a message sign. σ for m then σ is a Σ'-sign. for $H(m)$.
Either:
① $H(m)$ has been signed before (i.e. $H(m) = H(m_i)$, for some sig. query m_i) or
② $H(m)$ has never been signed before in a query

Def A function f is one-way iff f is computable in polynomial time and $\forall$ PPT A
$Pr[x \leftarrow \{0,1\}^\kappa; x' \leftarrow A(1^\kappa, f(x)) : f(x') = f(x)]$
is negligible.

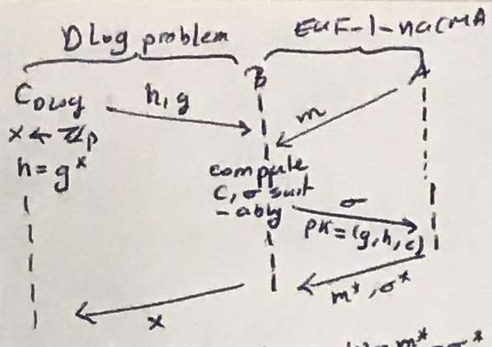
Thm if one-way fcts exist then $P \neq NP$. **Proof**: Let f be a one-way fct. $L_f := \{(y, \bar{x}, \ell) \mid \exists x \in \{0,1\}^\ell$ s.t. $\bar{x}$ is prefix of x & $f(x) = y\} \Rightarrow L_f \in NP$ (x is a witness for $(y, \bar{x}, \ell)$) but if $L_f \in P$ then $\exists$ inverter that runs in poly-time for f. Thus $L_f \in NP \setminus P$ and $P \neq NP$.

Lamport (EUF-1-CMA secure)
$sk = \begin{pmatrix} x_{1,0} & \dots & x_{n,0} \\ x_{1,1} & \dots & x_{n,1} \end{pmatrix}$ $pk = \begin{pmatrix} y_{1,0} & \dots \\ y_{1,1} & \dots \\ y_{n,0} & \\ y_{n,1} & \end{pmatrix}$
- $Sign(sk, m)$: $\sigma = (x_{1,m_1}, \dots, x_{n,m_n})$
- $Vfy(pk, m, \sigma)$: $f(x_i) \overset{?}{=} y_{i,m_i}$ $\forall i$



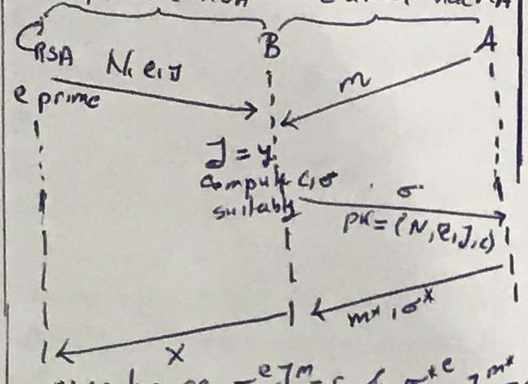
Lecture 3

- DLog problem: Given generator g and $y \leftarrow G_q$, find $x \in \mathbb{Z}_p$ with $g^x = y$
- DLog assumption: $\forall$ PPT A
$Pr[x \leftarrow \mathbb{Z}_p; x' \leftarrow A(1^\kappa, g, g^x) : x' = x]$
is negligible.
- **DLog OTS**
$Gen(1^\kappa)$: $x \leftarrow \mathbb{Z}_p^*$, $pk = (g, h, c)$, $sk = (x, \omega)$; $\omega \leftarrow \mathbb{Z}_p$; $h = g^x$; $c = g^\omega$
$Sign(sk, m)$: $(\omega - m)/x = \sigma$
Vfy: $c \overset{?}{=} g^m h^\sigma$



$x \Rightarrow$ found by solving $\frac{\omega - m^*}{x} = \sigma^*$ and $(\omega - m)/x = \sigma$

OTS RSA
RSA assumption: $Pr[N, e \text{ valid}; y \leftarrow \mathbb{Z}_N; x \leftarrow A(1^\kappa, N, e, y) : x^e = y \bmod N]$ negligible.
$\Sigma = (Gen, Sign, Vfy)$ msg space $= \{0, \dots, 2^{n-1}\}$
- $Gen(1^\kappa)$: choose primes P, Q $N = P \cdot Q$; e prime $2^n < e < \varphi(N)$; $d = e^{-1} \bmod \varphi(N)$; $J, c \leftarrow \mathbb{Z}_N$; $pk = (N, e, J, c)$; $sk = d$
- $Sign(sk, m)$: $\sigma = \left(\frac{c}{J^m}\right)^d \bmod N$
- Vfy: $\sigma^e J^m \overset{?}{=} c \bmod N$



given two eq. $\sigma^e J^m = c$ & $\sigma^{*e} J^{m^*} = c$
obtain $(\sigma/\sigma^*)^e = J^{m^* - m}$
$gcd(e, f) = 1 = gcd(e, m^* - m)$
use Shamir's trick to get $x^e = J = y$

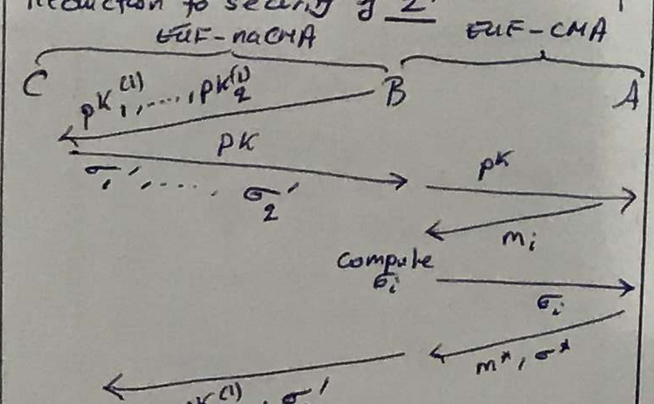
$J, s \in \mathbb{Z}_N^*$, $e, f \in \mathbb{Z}$
- $gcd(e, f) = 1$
- $Jf \equiv s^e \bmod N$

$e\alpha + \beta f = 1 \Rightarrow x = J^\alpha s^\beta$

Lecture 4

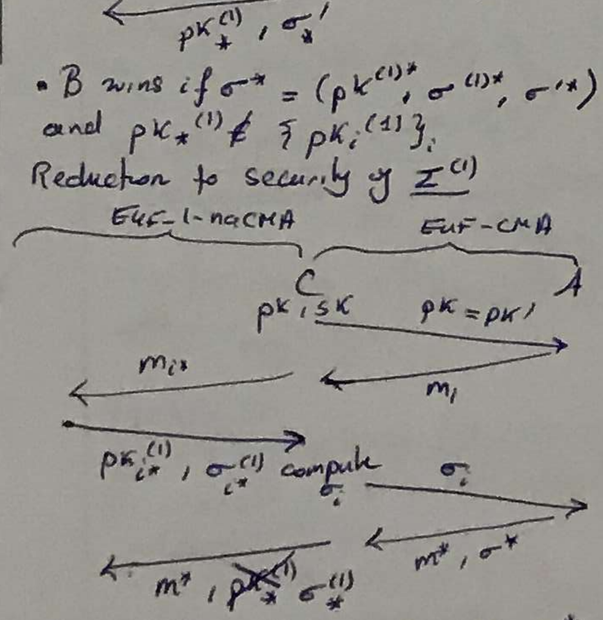
Given EUF-naCMA Σ' and EUF-1-naCMA $\Sigma^{(1)}$ construct EUF-CMA scheme Σ
- $Gen(1^\kappa)$: $(pk, sk) = (pk', sk') \leftarrow Gen'(1^\kappa)$
- $Sign(sk, m)$: $(pk^{(1)}, sk^{(1)}) \leftarrow Gen^{(1)}(1^\kappa)$; $\sigma' = Sign'(sk', pk^{(1)})$; $\sigma^{(1)} = Sign^{(1)}(sk^{(1)}, m)$; $\sigma = (pk^{(1)}, \sigma^{(1)}, \sigma')$
- $Vfy(pk, m, \sigma)$: $Vfy'(pk', pk^{(1)}, \sigma') = 1 \wedge Vfy^{(1)}(pk^{(1)}, m, \sigma^{(1)}) = 1$

Reduction to security of Σ'



- B wins if $\sigma^* = (pk^{(1)*}, \sigma^{(1)*}, \sigma'^*)$ and $pk_*^{(1)} \notin \{pk_i^{(1)}\}_i$

Reduction to security of $\Sigma^{(1)}$



wins iff $\sigma_*^{(1)}$ valid for $m^* \neq m_i^*$ and $pk_*^{(1)} = pk_{i^*}^{(1)}$

B wins w/ prob $\geq \varepsilon_A \cdot (1-p)$ where $p = Pr(pk_{i^*}^{(1)} \in \{pk_i^{(1)}\}_i)$ and C wins w/ prob $\geq \varepsilon_A \cdot p/q$
$\Rightarrow q\varepsilon_C + \varepsilon_B \geq \varepsilon_A$

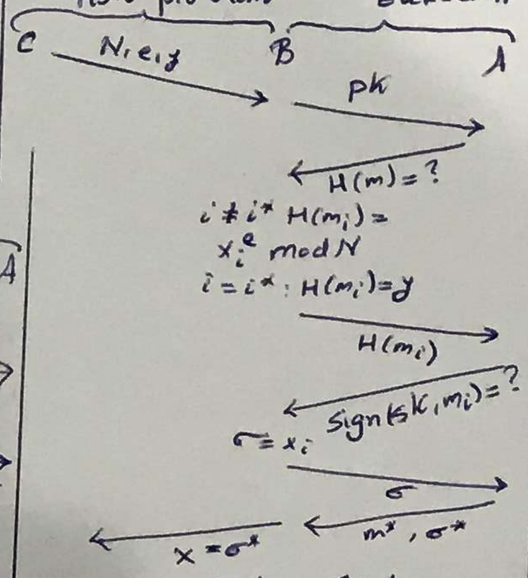
Construction of EUF-q-(na)CMA using Merkle trees
- $Gen(1^\kappa)$: $(pk_i, sk_i) \leftarrow Gen^{(1)}(1^\kappa)$; $pk = \text{tree-hash}(pk_1, \dots, pk_q)$; $sk = (sk_1, \dots, sk_q, pk_1, \dots, pk_q, st = 1)$
- $Sign(sk, m)$: $i = st$; $\sigma_i \leftarrow Sign^{(1)}(sk_i, m)$; $\sigma = (\sigma_i, pk_i, \text{co-path}_i, i)$; $st = st + 1$
- $Vfy(pk, m, \sigma)$: recompute root h. $h \overset{?}{=} pk \wedge Vfy^{(1)}(pk_i, m, \sigma_i)$

Complexity: $|pk| \in O(1)$, $|sk| \in O(q)$, $|\sigma| = O(\log q)$.
Optimization: use PRF to gen. key-pairs (no need to store keys) $|pk| \in O(1)$, $|sk| \in O(1)$, $|\sigma| \in O(\log q) \Rightarrow$ runtime of Gen & Sign in $O(q)$.
Optimization: build entire tree w/ OTS scheme. Every vertex is an OTS keypair. Every key signs the public keys of the 2 child vertices.

Lecture 5

RSA PKCS (security unclear)
- $Gen(1^\kappa)$: as w/ textbook RSA
- $Sign(sk, m)$: let H be a collision resistant hash fct. Encode m as $m' = 0x00 \| 0x01 \| 0xFF \| \dots \| 0xFF \| 0x00 \| \text{spec.} \; H \| H(m)$
$\sigma := m'^d \bmod N$
- Vfy: $m' = \sigma^e \bmod N$
Check that m' valid encoding of m.

RSA-FDH (EUF-CMA)
Let $H: \{0,1\}^* \to \mathbb{Z}_N$ be a coll.-res. hash fct. that fully spans $\mathbb{Z}_N$
- $Gen(1^\kappa)$: as with textbook RSA
- $Sign(sk, m)$: $\sigma := H(m)^d \bmod N$
- $Vfy(pk, m, \sigma)$: $\sigma^e \overset{?}{=} H(m) \bmod N$



$\Rightarrow \varepsilon_B \geq \varepsilon_A / q_H = \varepsilon_A / q_H$

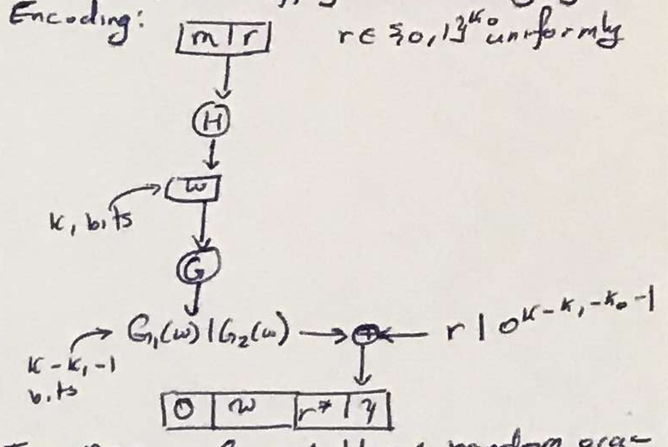
how to choose params:
- Goal: for all FOH adv. A, we want their quality $\varepsilon_A / t_A \leq 1/2^{100}$
- Allow $q_H = 2^{60}$ hash queries
- Reduction says $\frac{1}{2^{160}} \geq \varepsilon_B / t_B \geq \frac{\varepsilon_A}{t_A \cdot q_H}$

t_B / ε_B can be obtained from runtime of GNFS (current fastest Las Vegas Algorithm that solves RSA problem)
Reduction loss can be brought down to $O(q_S)$ instead of $O(q_H)$.

Lecture 6

RSA-PSS (EUF-CMA) under ROM model
- $Gen(1^\kappa)$: as with textbook RSA
- $Sign(sk, m)$:
$r \leftarrow \{0,1\}^\kappa$ uniformly
$\sigma := PSS\text{-}Encode(m)^d \bmod N$
- $Vfy(pk, m, \sigma)$:
  - compute $y = \sigma^e \bmod N$
  - output 1 iff y valid encoding of m

Encoding:



Thm Assume G and H are random oracles. Then for every adversary A that breaks the EUF-CMA security of RSA-PSS
- in time t_A
- w/ at most q_H hash queries to G and H
- at most q signature queries
- and success probability ε_A there is B that solves RSA problem in time t_B s.t. $\varepsilon_B \geq \varepsilon_A - (2(q+q_h)^2+1)(2^{-\kappa_0} + 2^{-\kappa_1})$
$t_B \leq t_A + (q+q_h+1) \cdot \kappa_0 \cdot O(n^3)$ $(\kappa_0 = \kappa_1)$

strong RSA assumption: $\forall$ PPT A
$Pr[N = P \cdot Q, J \leftarrow \mathbb{Z}_N; (x, e) \leftarrow A(1^\kappa, N, J) : x^e = J \bmod N \wedge e \neq 1]$
negligible.

GHR signature (EUF-naCMA)
Let $h: \{0,1\}^\kappa \to \mathbb{P}$ be a hash fct.
- $Gen(1^\kappa)$:
  - $N = P \cdot Q$
  - $s \leftarrow \mathbb{Z}_N$
  - $pk = (N, s, h)$
  - $sk = (pk, \varphi(N)) = (pk, (P-1)(Q-1))$
- $Sign(sk, m)$: $\sigma = s^{1/h(m) \bmod \varphi(N)} \bmod N$
- $Vfy(pk, m, \sigma)$ $\sigma^{h(m)} \overset{?}{=} s \bmod N$

Thm $\forall$PPT A that breaks EUF-naCMA sec. of Σ in time t_A w/ success ε_A there are PPT B and C ($t_B \approx t_C \approx t_A$) s.t.:
- B breaks the collision res. of h w/ succ. $\varepsilon_B \geq \varepsilon_A / 2$

OR
- C breaks strong RSA w/ succ. $\varepsilon_C \geq \varepsilon_A / 2$.

reduction for C: C gets as input (N, y) and needs to find (x, e) with $e > 1$, $x^e = y \bmod N$.
C sets up $s = y^{\prod_i h(m_i)} \bmod N$ $\forall i \in [q]$
- signature for m_j: $\sigma_j := y^{\prod_{i \neq j} h(m_i)} \bmod N$

A outputs valid forgery (m^*, σ^*) with $h(m^*) \neq h(m_i)$ $\forall i \in [q]$ and $(\sigma^*)^{h(m^*)} = s = y^{\prod_i h(m_i)} \bmod N$
$gcd(h(m^*), \prod_i h(m_i)) = 1$ since h outputs primes $\Rightarrow$ use Shamir.

Lecture 7

Def (Chameleon hash fct.): A CH consists of two PPT algo's $Gen_{CH}(1^\kappa)$ which outputs (τ, ch) and $TrapColl_{CH}(\tau, m, r, m')$ outputs r' s.t. $ch(m, r) = ch(m', r')$

Properties of CH
- well distributed $\forall m, m'$, r random $\Rightarrow$ r' random
- Collision resistant: hard to find $(m, r) \neq (m', r')$ s.t. $ch(m, r) = ch(m', r')$ only knowing ch.

DLog-based CH
- G group, $|G| = p$ prime

$Gen(1^\kappa)$:
- $x \leftarrow \mathbb{Z}_p^*$
- $h = g^x$
- $\tau = x$
- $ch = (g, h)$

$ch(m, r) = g^m h^r$

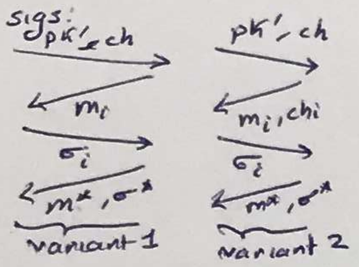
RSA-based CH
- $Gen(1^\kappa)$
  - $N = P \cdot Q$, msg space $\mathbb{Z}_{2^\ell}$
  - e prime, $e > 2^\ell$
  - $d = e^{-1} \bmod \varphi(N)$
  - $J \leftarrow \mathbb{Z}_N$
  - $pk = (N, e, J)$
  - $\tau = d$

$ch(m, r) = J^m r^e \bmod N$

Lecture 7

- Given $CH = (Gen_{CH}, TrapColl_{CH})$ and sig. scheme $\Sigma' = (Gen', Sig', Vfy')$
- $Gen(1^\kappa)$: $(pk', sk') \leftarrow Gen'(1^\kappa)$
  - $(ch, \tau) \leftarrow Gen_{CH}(1^\kappa)$
  - $pk = (pk', ch)$, $sk = (sk', \tau)$
- $Sign(sk, m, ch)$ ch is CHF of receiver
  $r \leftarrow R$, $ch(m,r) = y$
  $\sigma' = Sign'(sk', y)$
  $\sigma = (\sigma', r)$
- $Vfy(pk, m, \sigma, ch)$:
  - $Vfy'(pk, ch(m,r), \sigma') \stackrel{?}{=} 1$

There are two EUF-CMA variants for ch sigs:



we focus on variant 1 ONLY. (in the course).

Lecture 8

Thm $\forall$ PPT adv. $A(pk, ch)$ that breaks EUF-CMA security of Σ in time t_A w/ succ. ε_A there are PPT adv. B and C that run in time $t_C \approx t_B \approx t_A$ s.t.

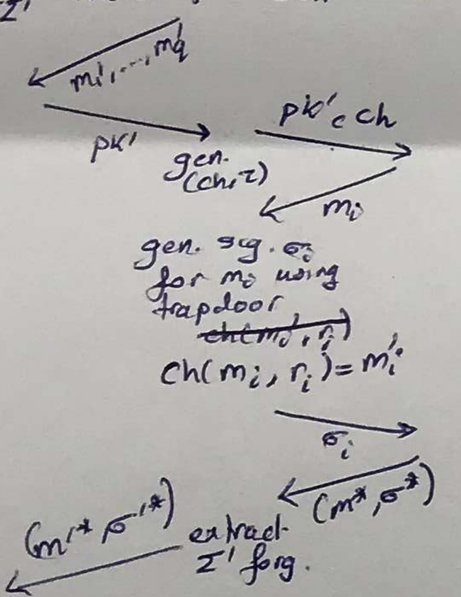
- C breaks the collision resistance of ch w/ success $\varepsilon_{ch} \geq \varepsilon_A/2$

OR

- B breaks the EUF-naCMA sec. of Σ' with prob. $\varepsilon' \geq \varepsilon_A/2$

Reduction to EUF-naCMA sec. of Σ'

$C_{\Sigma'}$ (naCMA) B (CMA) A



OTS from CH (EUF-1-naCMA sec. if CH coll. res.)

- $Gen(1^\kappa)$:
  $(ch, \tau) \leftarrow Gen_{CH}(1^\kappa)$
  $(\tilde{m}, \tilde{r}) \leftarrow M \times R$
  $c := ch(\tilde{m}, \tilde{r})$
  $pk = (ch, c)$
  $sk = (\tau, \tilde{m}, \tilde{r})$
- $Sign(sk, m)$:
  - $r = TrapColl_{CH}(\tau, \tilde{m}, \tilde{r}, m)$
  - $\sigma = r$
- $Vfy(pk, m, \sigma)$:
  $c \stackrel{?}{=} ch(m, \sigma)$

Def (sEUF-CMA)

$\forall$ PPT A $Pr[A(pk) = (m^*, \sigma^*)$: $Vfy(pk, m^*, \sigma^*) = 1 \wedge (m^*, \sigma^*) \notin \{(m_1, \sigma_1), \ldots, (m_q, \sigma_q)\}] \approx$ negl.

CH + EUF-CMA → sEUF-CMA

- $Gen(1^\kappa)$:
  - $(pk', sk') \leftarrow Gen'(1^\kappa)$
  - $(ch_F, \tau_F) \leftarrow Gen_{CH}(1^\kappa)$
  - $(ch_H, \tau_H) \leftarrow Gen_{CH}(1^\kappa)$
  - $pk = (pk', ch_F, ch_H)$
  - $sk = (sk', \tau_H)$
- $Sign(sk, m)$ $sk = (sk', \tau_H)$
  - $r_F \leftarrow R$, $r_H' \leftarrow R$
  - $h = ch_H(m' \| \sigma', r_H')$
  - $\tilde{m} = ch_F(h, r_F)$
  - $\tilde{\sigma} = Sign'(sk', \tilde{m})$
  - $r_H = TrapColl_{CH}(\tau_H, m' \| b', r_H', m \| \tilde{\sigma})$
  - $\sigma = (\tilde{\sigma}, r_F, r_H)$
- $Vfy(pk, m, \sigma)$
  $h = ch_H(m \| \tilde{\sigma}, r_H)$
  $\tilde{m} = ch_F(h, r_F)$
  $Vfy'(pk, \tilde{m}, \tilde{\sigma}) \stackrel{?}{=} 1$

Why sEUF-CMA?
"same $\tilde{m}$" $\Rightarrow$ "same h, r_F" $\Rightarrow$ "same $m, \tilde{\sigma}, r_H$" $\Rightarrow$ "same (m, σ)" if we assume both CHFs being collision-resistant.

skipped pairing alg:

Lecture 9

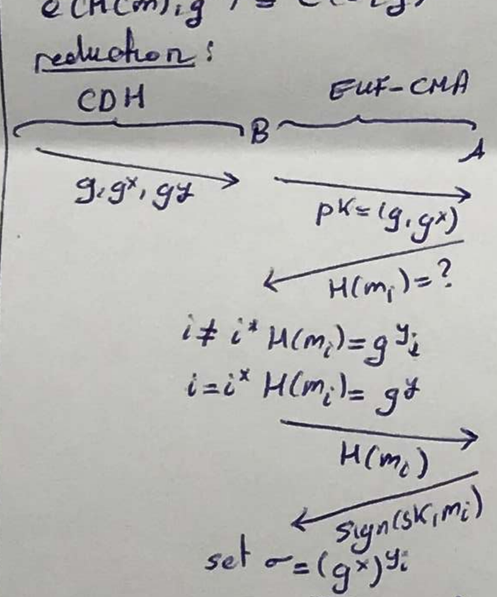
Joux's 3-party key exchg:
Parties A, B, C. A broadcasts g^A and gets g^B, g^C & then computes the shared key $e(g^B, g^C)^A = e(g,g)^{ABC}$
$(A, B, C \leftarrow \mathbb{Z}_p)$

BLS (EUF-CMA under ROM)

Hash fct: $\{0,1\}^* \to G \setminus \{1\}$
G, G_T all have same order p

- $Gen(1^\kappa)$: $x \leftarrow \mathbb{Z}_p^*$
- $pk = (g, g^x)$, $sk = x$
- $Sign(sk, m)$: $\sigma = H(m)^x \in G$
- $Vfy(pk, m, \sigma)$:
  $e(H(m), g^x) \stackrel{?}{=} e(\sigma, g)$

reduction:



$\Rightarrow \varepsilon_B \geq \varepsilon_A / q_H$

Can be optimized to loss of $O(q_S)$:
- upon hash query $H(m_i) = g^{y+y_i}$ (for known y_i) with probability $O(1/q_S)$. Set $H(m_i) = g^{y_i}$ w/ probability $1 - O(1/q_S)$

Extra property:

U_1 with $(pk_1, sk_1) \to m_1, \sigma_1$
$\vdots$
U_n with $(pk_n, sk_n) \to m_n, \sigma_n$

$\hookrightarrow$ aggregator $\xrightarrow{m_1, \ldots, m_n, \sigma_{Agg}}$ Verifier

Verifier checks $e(\sigma_{Agg}, g) \stackrel{?}{=} \prod_{i=1}^{n} e(H(m_i), g^{x_i})$
where $\sigma_{Agg} = \prod_i \sigma_i$

Lecture 10

Def: A group hash fct over a group G consists of two PPT algorithms

- $Gen(1^\kappa) \to \kappa$
- $Eval(\kappa, m) \to H_\kappa(m) \in G$

Def A group hash fct. $(Gen, Eval)$ is (ν, w, γ)-programmable if there are two PPT algorithms

- $TrapGen(g, h) \to (\kappa, \tau)$
- $TrapEval(\tau, m) \to (a_m, b_m)$

with $h^{a_m} g^{b_m} = H_\kappa(m)$ that fulfill two req:

- κ from Gen statistically close to κ from TrapGen
- TrapEval has (ν, w, γ)-well distributed outputs which means
  $\forall g, h, \kappa, (m_i^*)_{i=1}^{\nu}, (m_i)_{i=1}^{w}$:
  $Pr\left[\begin{array}{l} a_{m_i^*} = 0 \text{ for } i = 1 \ldots \nu \wedge \\ a_{m_j} \neq 0 \text{ for } j = 1 \ldots w \end{array}\right] \geq \gamma$

Lecture 11

Waters' group hash fct:

- $Gen(1^\kappa)$: choose $u_0, \ldots, u_\kappa \leftarrow G$
  $\kappa = (u_0, \ldots, u_\kappa)$
- $Eval(\kappa, m)$
  $H_\kappa(m) = u_0 \prod_{i \in M} u_i$

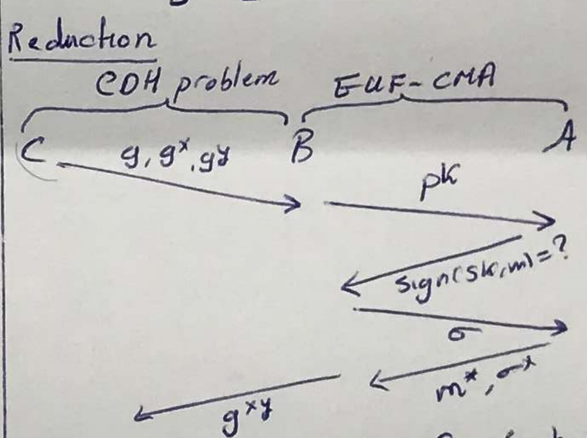
Let $q = q(\kappa)$ be a polynomial then Waters' group hash fct is $(1, q, \gamma)$ programmable for $\gamma = 1/\Theta(q\sqrt{\kappa})$

(note $u_i = h^{\hat{a}_i} g^{\hat{b}_i}$ and $a_m = \hat{a}_0 + \sum_{i=1}^{\kappa} m_i \hat{a}_i$ similarly for b_m) $\tau = (\hat{a}_0, \ldots, \hat{a}_\kappa, \hat{b}_0, \ldots, \hat{b}_\kappa)$

Waters signatures (EUF-CMA)

- $Gen(1^\kappa)$
  $g^\alpha \leftarrow G$, $\kappa \leftarrow Gen_{PHF}(1^\kappa)$
  $pk = (g, \kappa, e(g, g^\alpha))$
  $sk = g^\alpha$
- $Sign(sk, m)$: choose $r \leftarrow \mathbb{Z}_p$
  $\sigma_1 = g^r$ $\sigma_2 = g^\alpha H(m)^r$
  $\sigma = (\sigma_1, \sigma_2)$
- Vfy $e(g, \sigma_2) \stackrel{?}{=} e(g, g^\alpha) \cdot e(\sigma_1, H(m))$

Reduction



B computes $(\kappa, \tau) \leftarrow TrapGen(g, h = g^x)$
sets $pk = (g, \kappa, e(g^x, g^y)) = (g, \kappa, e(g,g)^{xy})$
$\Rightarrow sk = g^\alpha = g^{xy}$
B wants event E: 1) $a_{m_i} \neq 0$ $\forall$ queries m_i
2) $a_{m^*} = 0$ for forgery m^*
since H is $(1, q, \gamma_q)$-programmable, we have
$Pr[E] \geq \gamma_q \Rightarrow \varepsilon_B \geq \gamma_q \cdot \varepsilon_A$
signing query: $(a_m, b_m) \leftarrow TrapEval(\tau, m)$
$\Rightarrow H(m) = (g^x)^{a_m} g^{b_m}$. Choose fresh $s \leftarrow \mathbb{Z}_p$ and use $r = -y/a_m + s$
For forgery: $\sigma_2^* (\sigma_1^*)^{-b^*} = g^{xy}$