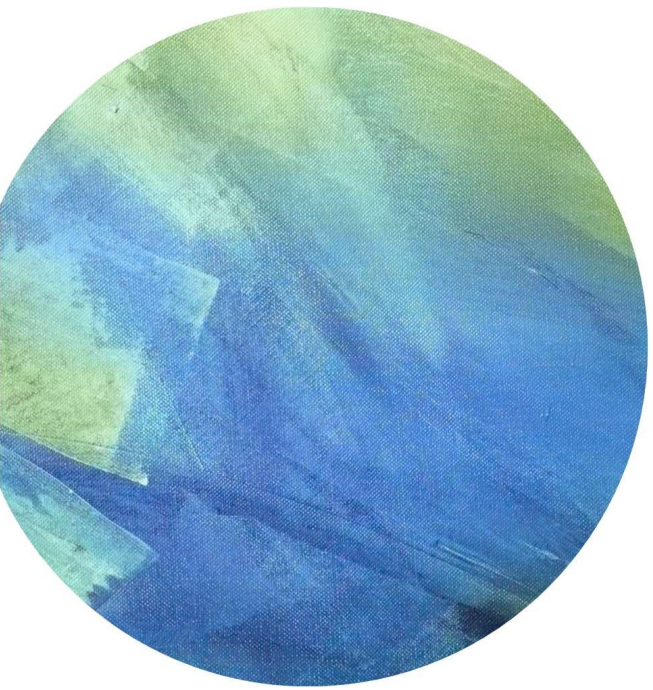
Lecture 18



Outline

Pairing-based cryptography

NIZKs from pairings

Recall: Pairings

Pairing-based cryptography

- So far, we've looked at hard problems like discrete log, CDH, HDH, DDH in groups
- Certain groups have an additional structure

- Let G_0, G_1, G_T be 3 cyclic groups of prime order

  where $g_0 \in G_0$ and $g_1 \in G_1$ are generators

- A pairing is an efficiently computable function $e: G_0 \times G_1 \rightarrow G_T$ such that:

  1. $g_T = e(g_0, g_1)$ is a generator of G_T
  2. For all $(u, u') \in G_0$ and $(v, v') \in G_1$,

     $e(u.u', v) = e(u,v).e(u',v)$ and $e(u, v.v') = e(u,v).e(u,v')$

Pairing-based cryptography

- A pairing is an efficiently computable function e: $G_0 \times G_1 \rightarrow G_T$ such that:
  1. $g_T = e(g_0, g_1)$ is a generator of G_T
  2. For all $(u, u') \in G_0$ and $(v, v') \in G_1$,

     $e(u.u', v) = e(u,v).e(u',v)$ and $e(u, v.v') = e(u,v).e(u,v')$

- Consequences: $e(g_0^a, g_1^b) = e(g_0,g_1)^{ab} = e(g_0^b, g_1^a)$

$g^a \; g^b \not\rightarrow g^{ab}$ (in regular groups where CDH is hard)

$e(g^a, g^b) \rightarrow g_T^{ab}$ (in pairing groups, makes DDH easy)

A Useful Hardness Assumption

- Co-CDH assumption:
  - Sample random (a, b) in Z_q
  - $u_0 = g_0^a$, $u_1 = g_1^a$, $v_0 = g_0^b$, $z_0 = g_0^{ab}$
  - Send (u_0, u_1, v_0) to $\mathcal{A}$
  - $\mathcal{A}$ outputs $z' \in G_0$
  - $\mathcal{A}$ wins if $z' = z_0$.

cannot output g_0^{ab}.

Warmup: BLS Signatures

G_0 G_1

pk g_1^a

σ $(H(m))^a$

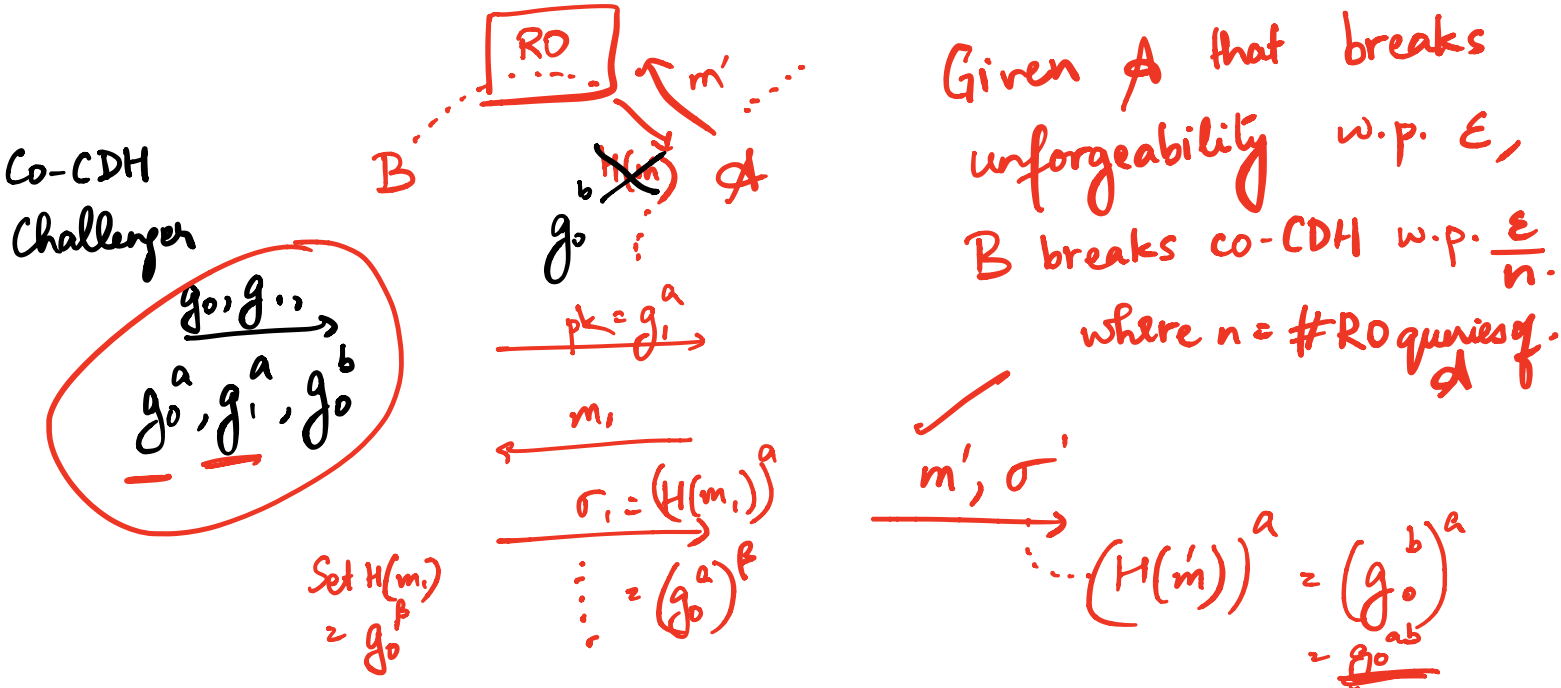
- Constructed as:
  1. KeyGen (1^k): Sample random a, output (sk = a), (pk = g_1^a).
  2. Sign (sk = a, m): $\sigma = (H(m))^a \in G_0$
  3. Verify (pk, m, σ): Output 1 iff e (H(m), pk) = e (σ, g_1).

check : $e(H(m), g_1^a) = e(H(m)^a, g_1)$

Suppose $\mathcal{A}$, given pk and signatures σ_1 on m_1, σ_2 on m_2 ..... outputs (m', σ') s.t. Verify $(pk, m', \sigma') = 1$.

BLS Signatures Proof of Unforgeability

1. KeyGen (1^k): Sample random a, output (sk = a), (pk = g_1^a).
2. Sign (sk = a, m): $\sigma = (H(m))^a \in G_0$
3. Verify (pk, m, σ): Output 1 iff e (H(m), pk) = e (σ, g_1).



BLS Signatures Proof of Unforgeability

1. KeyGen (1^k): Sample random a, output (sk = a), (pk = g_1^a).
2. Sign (sk = a, m): $\sigma = (H(m))^a \in G_0$
3. Verify (pk, m, σ): Output 1 iff e (H(m), pk) = e (σ, g_1).

Zero-Knowledge from Pairings

[GOS 06]

SHA ⇒ Factoring

hardness ⇐ hardness

G, G_T

Subgroup of G : $G_{(q)}$

Another Assumption

- Subgroup Hiding Assumption

N is a product of 2 large primes. q is secret

- Let G, G_T be cyclic groups of *composite* order $N = pq$
  Let (g, g_T) denote generators of (G, G_T), and define $e: G \times G \rightarrow G_T$
- Define $G_{(q)}$ to be the unique order-q subgroup of G (fact: such a subgroup exists!)
- Pick a generator g of G
- Set $h_0 \leftarrow$ random generator of G, $h_1 \leftarrow$ random generator of $G_{(q)}$
- Send (g, h_b) to Adv. Adv wins if it guesses b.

(N, g, h_b)

G and $G_{(q)}$ are diff.

order N order q.

(g, h_1, N)
$\downarrow$
$\in Gen(G_{(q)})$

A B

Commitments

DUAL-MODE COMMITMENTS.

- Commitments

If (g, h, N) s.t. $h \in Gen(G)$
* comp. binding
* perfectly hiding

- Public parameters: (g, h) where g ← Gen(G) and h ← $Gen(G_{(q)})$
- Commit (m;r) = $g^m h^r$

- This commitment is perfectly binding: $\{g^{m_1} h^r\}_r$ $\{g^{m_2} h^r\}_r$
  - For all $m_1 \neq m_2$ in Z_p, Supp(Commit(m_1;r)) ∩ Supp(Commit(m_2;r)) = {}
- This commitment is computationally hiding:
  - By subgroup hiding, h ← $Gen(G_{(q)})$ is indistinguishable from h ← Gen(G)
  - Perfectly hiding when h ← Gen(G)

↳ $h = g^\alpha$ for some α.

$g^{m+r\alpha}$

BINDING: Suppose $\exists\, r_1, r_2$ s.t. $g^{m_1} h^{r_1} = g^{m_2} h^{r_2}$

$g^{m \in Z_p} = g^{m_1 - m_2} \cdot h^{r_2 - r_1} \in G_{(q)} = g^{p.1}, g^{p.2} \ldots$

These Commitments are Homomorphic

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = $g^m h^r$

- These are homomorphic:

- Commit(m_1;r_1) . Commit(m_2;r_2) = g^{m1+m2} h^{r1+r2} = Commit(m_1+m_2;r_1+r_2)

$g^{m_1} h^{r_1} \cdot g^{m_2} h^{r_2}$

P V

c →

$c \in G_{(q)}$ or $cg^{-1} \in G_{(q)}$

Proof that a commitment is to 0 or 1

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = $g^m h^r$

- $c = \{g^0 h^r\}_r \cup \{g^1 h^r\}_r \quad \Leftrightarrow \quad c \in G_{(q)}$ or $cg^{-1} \in G_{(q)}$

$c = g^0 h^r$ for some r

$\Leftrightarrow c \in G_{(q)}$

$c = g^1 h^r$ for some r

$\Leftrightarrow \frac{c}{g} \in G_{(q)}$

Proof that a commitment is to 0 or 1

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = $g^m h^r$

- $c = \{g^0 h^r\}_r \cup \{g^1 h^r\}_r \quad \Leftrightarrow \quad c \in G_{(q)}$ or $cg^{-1} \in G_{(q)}$

  $\Leftrightarrow \quad e(c, cg^{-1})^q = \mathbf{1}_T$

$e(c, cg^{-1})^q = e(c^q, cg^{-1}) = e(c, (cg^{-1})^q)$

$= 1_T$.

$c \in G_{(q)} \iff c^q = 1_{G(q)} = 1_G$ $\qquad 1_G = g^N$

$cg^{-1} \in G_{(q)} \iff (cg^{-1})^q = 1_G$. $\qquad 1_G = 1_{G(q)}$

$= e(g^N, A) = e(g, A)^N = 1_T$.

$\forall A, A', \quad e(1_G, A) = 1_T = e(A', 1_G)$

Proof that a commitment is to 0 or 1

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = $g^m h^r$

- $c = \{g^0 h^r\}_r \cup \{g^1 h^r\}_r \quad \Leftrightarrow \quad c \in G_{(q)}$ or $cg^{-1} \in G_{(q)}$

  $\Leftrightarrow \quad e(c, cg^{-1})^q = \mathbf{1}_T$

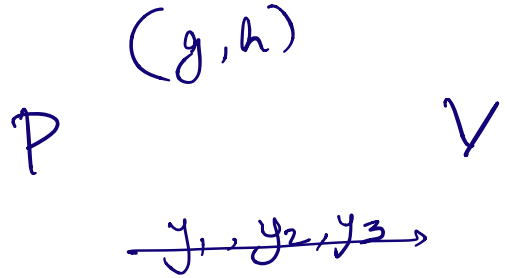
Tempting: V(c) computes $e(c, cg^{-1})^q$ and checks if the result is 1_T.

But V doesn't know q!

Proof that a commitment is to 0 or 1

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = $g^m h^r$ $\in \{0,1\}$
- Proof = (y_1, y_2, y_3) where
  - s ← Z_N^* , t = s^{-1} mod N, u = $g^{2m-1}h^r$
  - $y_1 = h^s$
  - $y_2 = u^{rt}$
  - $y_3 = g^s$

$st = 1 \bmod N$



$u^{rt} = (g^{-1}h^r)^{rt}$ or $(gh^r)^{rt}$

$(m=0)$ $(m=1)$

Proof that a commitment is to 0 or 1

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = c = $g^m h^r$

- Proof = (y_1, y_2, y_3) where
  - s ← Z_n^* , t = s^{-1} mod n , u = $g^{2m-1} h^r$
  - $y_1 = h^s$
  - $y_2 = u^{rt}$
  - $y_3 = g^s$

Verify (c, y_1, y_2, y_3) →

- First, check if y_1 is of the form h^s for some s
- e (y_1, g) = e (h, y_3)

Suppose mal. P^* set $y_1 = g^{100}$. Suppose $g^{100} \notin G_{(q)}$

$e(y_1, g) = e(h, g^{\tilde{s}}) = e(h, g)^{\tilde{s}}$

$= e(y_1, g)$ if & only if $y_1 = h^{\tilde{s}}$.

Proof that a commitment is to 0 or 1

$e(g^a, g^b) = e(g^b, g^a)$ when $G_0 = G_1$

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← $Gen(G_{(q)})$
- Commit (m;r) = c = $g^m h^r$

- Proof = (y_1, y_2, y_3) where
  - s ← Z_n^* , t = s^{-1} mod n , u = $g^{2m-1}h^r$
  - $y_1 = h^s$
  - $y_2 = u^{rt}$
  - $y_3 = g^s$

Verify (c, y_1, y_2, y_3) →

- Next, check if c ∈ $G_{(q)}$ or $cg^{-1} \in G_{(q)}$
- $e(c, cg^{-1}) = e(y_1, y_2)$

$e(y_1, y_2) = e(h^s, u^{rt}) = e(h^r, u^{st}) = e(h^r, u)$

$= e(h^r, g^{2m-1} \cdot h^r)$

$m=0 \Rightarrow e(y_1, y_2) = e(h^r, g^{-1} \cdot h^r) = e(c, cg^{-1})$

$m=1 \Rightarrow e(y_1, y_2) = e(h^r, g \cdot h^r) = e(cg^{-1}, c) = e(c, cg^{-1})$

Proof that a commitment is to 0 or 1

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = c = $g^m h^r$

- Proof = (y_1, y_2, y_3) where
  - s ← Z_n^* , t = s^{-1} mod n , u = $g^{2m-1} h^r$
  - y_1 = h^s
  - y_2 = u^{rt}
  - y_3 = g^s

Verify (c, y_1, y_2, y_3) → check

- e (c, cg^{-1}) = e (y_1, y_2) and
- e (y_1, g) = e (h, y_3)

Old: $e(c, cg^{-1})^q \stackrel{?}{=} 1$ (But q leaks info!)

new: $e(c, cg^{-1}) = e(y_1, y_2)$

Proof that a commitment is to 0 or 1

- Commitments
- Parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$). Commit (m;r) = c = $g^m h^r$

- Proof = (y_1, y_2, y_3) where
  - s ← Z_n* , t = s^{-1} mod n , u = $g^{2m-1}h^r$
  - $y_1 = h^s$, $y_2 = u^{rt}$, $y_3 = g^s$

Verify (c, y_1, y_2, y_3) → check

- e (c, cg^{-1}) = e (y_1, y_2) and
- e (y_1, g) = e (h, y_3)

Proof that a commitment is to 0 or 1

- Commitments
- Parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$). Commit (m;r) = c = $g^m h^r$

- Proof = (y_1, y_2, y_3) where
  - s ← Z_n^* , t = s^{-1} mod n , u = $g^{2m-1}h^r$
  - $y_1 = h^s$, $y_2 = u^{rt}$, $y_3 = g^s$

Verify (c, y_1, y_2, y_3) → check

- e (c, cg^{-1}) = e (y_1, y_2) and
- e (y_1, g) = e (h, y_3)

Proof that a commitment is to 0 or 1

- Commitments
- Parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$). Commit (m;r) = c = $g^m h^r$

- Proof = (y_1, y_2, y_3) where
  - s ← Z_n^* , t = s^{-1} mod n , u = $g^{2m-1}h^r$
  - $y_1 = h^s$, $y_2 = u^{rt}$, $y_3 = g^s$

Verify (c, y_1, y_2, y_3) → check

- e (c, cg^{-1}) = e (y_1, y_2) and
- e (y_1, g) = e (h, y_3)

These Commitments are Homomorphic

- Commitments
- Public parameters: (g, h) where g ← Gen(G) and h ← Gen($G_{(q)}$)
- Commit (m;r) = $g^m h^r$

- These are homomorphic:

- Commit(m_1;r_1) . Commit(m_2;r_2) = g^{m1+m2} h^{r1+r2} = Commit(m_1+m_2;r_1+r_2)

Why is this useful?

Why is this useful?

- For every NAND gate with input wires (i,j) and output wire k,
  $w_k = w_i$ NAND w_j $\Leftrightarrow$ $w_i + w_j + 2w_k - 2 \in \{0,1\}$

Why is this useful?

- Commit to each wire w_i in the circuit as $c_i = com(w_i; r_i)$
- For the output wire set $c_{out} = com\ (1; 00…0)$
- Prove that for every i, c_i is a commitment to 0 or 1

Why is this useful?

- Commit to each wire w_i in the circuit as $c_i = com(w_i; r_i)$
- For the output wire set $c_{out} = com(1; 00…0)$
- Prove that for every i, c_i is a commitment to 0 or 1
- For all NAND gates with input wires (i,j) and output wire k, prove correctness by proving that $w_i + w_j + 2w_k - 2 \in \{0,1\}$
- Use homomorphism property:
  compute $c = c_i\, c_j\, c_k^2\, g^{-2} = (g^{w\_i + w\_j + 2w\_k - 2}).(h^{r\_i + r\_j + 2r\_k})$
  prove that c is a commitment to 0 or 1