Blockchains

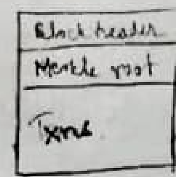
Created by Satoshi Nakamoto (pseudonym) - 2008. It is an "append-only", immutable, transparent ledger.

Genesis block B0: "Chancellor on brink of second bailout for banks"

Mining: It is the process of determining the next block to be appended to the ledger.

A block B_i consists of h_{i+1}, I_i, pk, n_i, which satisfies

$H(h_i, I_i, pk, n_i) = \underbrace{00\ldots0}_{k}XXX\ldots X$. k is known as difficulty parameter. The value of the hash becomes h_{i+1}. This is known as "proof of work", requiring a lot of compute.



→ By capturing >50% of compute, one can come up with a longer fork branching at a past state, this leads to "double spending". $B_i = (h_{i+1}, I_i, pk, n_i)$

→ Larger cryptocurrencies' miners try to perform double spending attacks on smaller cryptocurrencies, called "infanticide".

→ Proof of work is unsustainable in terms of energy.

Proof of Stake

Used by Peercoin in 2012, Ethereum in 2023 (hard fork/system enforced).

→ The next miner who proposes the block is chosen by a pseudo-random process that depends on the miner's amount of coins (stake)

→ The miner with the highest score is chosen for the next block

The score is proportional to $r_{pk} \cdot c_{pk}$ [r_{pk} → random value assoc. w/ pk, c_{pk} → amt. of coins miner has]. c_{pk} → stake, $r_{pk} \cdot c_{pk}$ → score

→ Generation of r_{pk}.

a. Miner chooses r_{pk}: obviously choose larger r_{pk}

b. Miner uses deterministic hash function $H(.)$ s.t. $r_{pk,i+1} = H(B_i, pk_m)$. Here, miner can decide which B_i to select to ensure $H(B_i, pk_m)$ is largest. Miner may also take control for B_{i+1}, etc. as they know B_i. (Unfair advantage on next block)

c. $r_{pk,i+1} = H(i, pk)$

Miner can compute a "predictable attack" i.e. compute hash values before System is run and choose the pk with best random values.

d. $r_{pk,i+1} = H(r^{winner}_{pk,i}, pk)$. This is "semi-predictable" if $r^{winner}_{pk,i+1}$ is known. Then, further r-values for a pk can be computed, and money transfered to appropriate pk.

e. $r_{pk,i+1} = H(Sign_{sk}(r^{winner}_{pk,i}), pk)$.

An issue with proof of stake is "double spending", where two chains can be mined from a block, but there is no idea which chain is legitimate.

Pairing-Based Cryptography

Issue w/ using $\mathbb{Z}_p^*$: General Number Field Sieve (GNFS) attacks dlog in $\exp(O(\log p)^{1/3})$. However, on elliptic curves, $O(\sqrt{q})$ generic group attacks are known, where q is the order of the group.

An elliptic curve group uses a degree 3 curve such as $y^2 = x^3 + ax + b$. Suppose $P, Q \in \mathbb{Q}^2$ lie on the curve. Denote by $-R$ the third intersection of PQ with the curve. Then, $P \boxplus Q = R$ is the group operation where $X = (x, y)$, $-X = (x, -y)$. This is known as the "chord technique".

Elliptic Curve: If $char(\mathbb{F}) \notin \{2, 3\}$, the curve can be described as the set of solutions (x, y) that satisfy $y^2 = x^3 + ax + b$ for some $a, b \in \mathbb{F}$ s.t. $4a^3 + 27b^2 \neq 0$ (to avoid singularities/cusps) along with the special unique point called the "point at infinity". [Short Weirstrass Form]

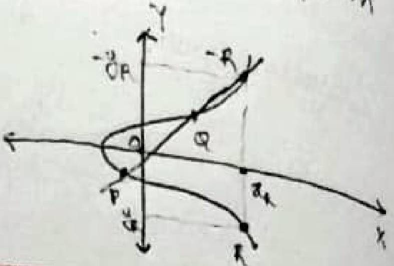
Let $E/\mathbb{F}_p$ be a curve over $\mathbb{F}_p$. We say that (x_1, y_1) is a point over E if (x, y) satisfies the curve equation. The set of points that lie on the curve is $E(\mathbb{F}_p) = \{\mathcal{O}\} \cup \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p : y^2 = x^3 + ax + b,\ a, b \in \mathbb{F}\}$.

Example: $y^2 = x^3 + 4x + 4$. $E(\mathbb{Z}_5) = \{\mathcal{O}, (0,2), (0,3), (1,2), (1,3), (2,0), (4,2), (4,3)\}$

Hasse's Theorem: $|E(\mathbb{F}_{p^e})| = p^e + 1 - t$, where $t \in \mathbb{Z}$ s.t. $|t| \leq 2\sqrt{p^e}$

Asymptotically, $|E(\mathbb{F}_{p^e})| \approx p^e$.

The SEA algorithm counts the number of points in $E(\mathbb{F}_{p^e})$ in $O(\log p^c)$ time.



Elliptic Curve Addition

Case 1: $P \neq Q$, $x_P \neq x_Q$

Suppose P, Q lie on $y = mx + c$.

Then, $m = \dfrac{y_P - y_Q}{x_P - x_Q}$

Thus, $(mx+c)^2 = x^3 + ax + b \Rightarrow x^3 - m^2x^2 + (a - 2mc)x + b - c^2 = 0$

But x_P, x_Q, x_R satisfy the above equation. Thus,

$x_R = m^2 - x_P - x_Q$, $y_R = m^3 - m(x_P + x_Q) + c$.

Case 2, 3: $Q = -P$, $P = Q$ where $y_P = y_Q = 0$. Here, $P + Q = \mathcal{O}$

Case 4: $P = Q$, $y_P \neq 0$. Here, $m = \dfrac{3x_P^2 + a}{2y_P}$ [This is why char $\neq 2, 3$].

Here, $x_R = m^2 - 2x_P$, $y_R = m^3 - 2x_P + c$.

The pair $(E(\mathbb{F}), \boxplus)$ is an elliptic curve group. It is finite and Abelian.

Discrete Logarithm in $E(\mathbb{F})$. If the order of the group is prime q, let P be a point of order q. Let $Q = \alpha P$ [$\alpha \in \mathbb{Z}_q$] for some α. Given P and Q, computing α is hard.

Insecure Curves

1. If $|E(\mathbb{F}_p)| = p$, dlog is polynomial time in this group. Such curves are called "anomalous curves"
2. If $|E(\mathbb{F}_p)|$ is composite, dlog runs in $O(\sqrt{q_{max}})$, where q_{max} is the maximum prime factor of $|E(\mathbb{F}_p)|$.

Examples of secure curves:

1. secp256r1 : used in internet protocols, $P_r = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$. $y^2 = x^3 - 3x + b$
2. secp256k1 : $P_r = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$. $y^2 = x^3 + 7$.

Pairings: Let G_0, G_1, G_T be three cyclic groups of prime order q where $G_0 = \langle g_0 \rangle$, $G_1 = \langle g_1 \rangle$ (we assume that G_0, G_1 are additive and G_T is multiplicative).

A pairing e is an efficiently computable map, $e: G_0 \times G_1 \to G_T$ satisfying the following two properities:

1. **Bilinear**: $\forall u, u' \in G_0$, $\forall v, v' \in G_1$, $e(u+u', v) = e(u,v)\,e(u',v)$ and $e(u, v+v') = e(u,v) \cdot e(u,v')$.
2. **Nondegeneracy**: All the outputs of e should not be the identity element of G_T ($\Leftrightarrow$) A consequence of this is that $e(g_0, g_1) = g_T$, where $\langle g_T \rangle = G_T$.

Due to bilinearity, $\forall \alpha, \beta \in \mathbb{Z}_q$ $e(\alpha g_0, \beta g_1) = e(g_0, g_1)^{\alpha\beta} = e(\beta g_0, \alpha g_1)$.

$e(\alpha g_0, \beta g_1) = e(\underbrace{g_0 + \cdots + g_0}_{\alpha \text{ times}}, \beta g_1) = [e(g_0, \beta g_1)]^{\alpha} = e(g_0, g_1)^{\alpha\beta}$.

We prove the iff in property 2.

($\Rightarrow$) If the pairing is nondegenerate, $\exists (u,v)$ s.t. $e(u,v) \neq 1_{G_T}$

Let $e(u,v) = g' \neq 1_{G_T}$. Since G_0, G_1 is cyclic, $u = \alpha g_0$, $v = \beta g_1$ and $e(g_0, g_1)^{\alpha\beta} \neq 1_{G_T}$. Thus, $e(g_0, g_1) \neq 1_{G_T}$. Since G_T has prime order q, $\langle e(g_0, g_1) \rangle = G_T$.

Consequences of Pairings

1. If $G_0 = G_1$, the pairing is symmetric. Then, DDH is easy in G_0.
This is because $e(\alpha g_0, \beta g_0) = e(g_0, \alpha\beta g_0)$. Thus, given a DH triple $(\alpha g_0, \beta g_0, \gamma g_0)$ check if $e(\alpha g_0, \beta g_0) \stackrel{?}{=} e(g_0, \gamma g_0)$.

2. Computing dlog in G_0 and G_1 is only as hard as computing the dlog in G_T. (Given $u = \alpha g_0$, find α)
We can compute $e(u, g_1) = e(\alpha g_0, g_1) = e(g_0, g_1)^{\alpha} = g_T^{\alpha}$

Usually, for elliptic curves.

G_0: order q subgroup of $E(\mathbb{F}_p)$.

G_1: order q subgroup of $E(\mathbb{F}_{p^d})$, $d > 0$, $G_1 \cap G_0 = \{O\}$.
d is the embedding degree. For small representation, $d \leq 16$.

G_T: order q multiplicative subgroup of finite field $\mathbb{F}_{p^d}$

BLS Signature Scheme [Aggregatable]

Consider groups G_0, G_1, G_T of prime order q, $e: G_0 \times G_1 \to G_T$ and a hash function $H: \mathcal{M} \to G_0$ $[G_0 = EC(\mathbb{F}_p), G_1 = EC(\mathbb{F}_{p^\alpha}), G_T \in \mathbb{F}_{p^\alpha}]$

KeyGen(): $\alpha \xleftarrow{\$} \mathbb{Z}_q$, $u = \alpha g_1 \in G_1$. $pk := u$, $sk := \alpha$.

Sign(m, sk). $\sigma \leftarrow \alpha H(m) \in G_0$

Verify (pk, m, σ): $e(\sigma, g_1) \stackrel{?}{=} e(H(m), u)$

Attack Game for co-CDH (Computational Diffie Hellman)

1. Challenger computes

$$\alpha, \beta \xleftarrow{\$} \mathbb{Z}_q, \quad u_0 = \alpha g_0 \quad v_0 = \beta g_0$$
$$u_1 = \alpha g_1 \quad z_0 = \alpha\beta g_0$$

and sends (u_0, u_1, v_0) to $\mathcal{A}$.

2. Adversary $\mathcal{A}$ outputs $\hat{z}_0 \in G_0$

Adversary's advantage in solving co-CDH is defined as

$$\boxed{Adv^{co\text{-}CDH}_{\mathcal{A}, 1^n} \triangleq Pr(\hat{z}_0 = z_0)}$$

<u>co-CDH Assumption</u>: We say that co-CDH assumption holds for a pairing $e: G_0 \times G_1 \to G_T$ if $Adv^{co\text{-}CDH}_{\mathcal{A}, 1^n}$ is negligible

Security of BLS Signatures

<u>Theorem</u>: Let $e: G_0 \times G_1 \to G_T$ be a pairing and $H: \mathcal{M} \to G_0$ be a hash function. The BLS signature scheme $\mathcal{S}_{BLS}$ is a secure signature scheme if co-CDH holds for e and H is modeled as an RO.

In particular, let $\mathcal{A}$ be a PPT adversary that attacks $\mathcal{S}_{BLS}$ using the standard signature forgery attack game. We assume $\mathcal{A}$ makes Q_{ro} queries to the RO and Q_s queries to the signing oracle. Then, there exists a PPT adversary $\mathcal{B}$ which has

$$Adv_{\mathcal{A}, \mathcal{S}_{BLS}} \leq (Q_{ro} + 1)\, Adv^{co\text{-}CDH}_{\mathcal{B}, e, 1^n}$$

An improvement: $Adv_{\mathcal{A}, \mathcal{S}_{BLS}} \leq 2.72\,(Q_s + 1)\, Adv^{co\text{-}CDH}_{\mathcal{B}, e, 1^n}$ [Usually, $Q_{ro} \gg Q_s$ so this is an improvement]

Proof 1. Adversary $\mathcal{B}$ is given a tuple $(u_0 = \alpha g_0, u_1 = \alpha g_1, v_0 = \beta g_0)$ as in the co-CDH attack game and he has to compute $z_0 = \alpha\beta g_0$.

2. $\mathcal{B}$ calls $\mathcal{A}$ with $pk = u_1 = \alpha g_1$.

3. $\mathcal{A}$ will make Q_{ro} queries to $\mathcal{B}$ and Q_s queries to $\mathcal{B}$

4. $\mathcal{B}$ sets $\omega \in \{1, 2, \ldots Q_{ro}+1\}$

5. For any $j \neq \omega$, for m_j, $\mathcal{B}$ responds to $H(m_j) = \rho_j g_0$, where $\rho_j \xleftarrow{\$} \mathbb{Z}_q$.

For $j = \omega$, $\mathcal{B}$ responds to $H(m_j) = v_0$

6. When $\mathcal{A}$ makes a signing query on m_j, $\mathcal{B}$ responds $\sigma(m_j, \alpha) = \alpha(\rho_j g_0) = \rho_j u_0$

$\mathcal{A}$ comes up with forgery (m, σ) where $\sigma = \alpha H(m) = \alpha v_0 = \alpha\beta g_0$ if $m = m_\omega$.

Thus, $Adv^{co\text{-}CDH}_{\mathcal{B}, e, 1^n} \geq \dfrac{Adv_{A, \mathcal{S}_{BLS}}}{Q_{ro}+1} \Rightarrow Adv_{A, \mathcal{S}_{BLS}} \leq (Q_{ro}+1) Adv_{\mathcal{B}, e, 1^n}$

[Extra query to account for $\mathcal{B}$ not knowing β]

Signature Aggregation

Definition: An aggregate signature scheme, SA is a signature scheme with two additional efficient algorithms.

1. SigAgg / $A(\vec{pk}, \vec{\sigma})$, where $\vec{pk} = (pk_1, \ldots, pk_n)$, $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, outputs σ_{ag}.

2. AggVer / $VA(\vec{pk}, \vec{m}, \sigma_{ag})$, where $\vec{m} = (m_1, \ldots, m_n)$, outputs accept or reject.

Correctness: SA is correct if $\forall\ \vec{pk} = (pk_1, \ldots, pk_n)$, $\vec{m} = (m_1, \ldots, m_n)$, $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, $VA(\vec{pk}, \vec{m}, A(\vec{pk}, \vec{\sigma})) = 1 \Leftrightarrow V(pk_i, m_i, \sigma_i) = 1\ \forall\ i \in \{1, \ldots, n\}$

Attempt at Aggregating BLS

$e: G_0 \times G_1 \longrightarrow G_T$, $G_0 = \langle g_0 \rangle$, $G_1 = \langle g_1 \rangle$, $G_T = \langle g_T \rangle$, all order q. $H: \mathcal{M} \to G_0$

Keygen(): $\alpha \xleftarrow{\$} \mathbb{Z}_q$, $pk \leftarrow \alpha g_0$.

Sign(α, m): $\sigma \leftarrow \alpha H(m)$

Verify(pk, m, σ): $e(H(m), pk) \stackrel{?}{=} e(\sigma, g_1)$

$A(\vec{pk} \in G_1^n, \vec{\sigma} \in G_0^n)$: $\sigma_{ag} \leftarrow \sum_{i=1}^n \sigma_i = \sum_{i=1}^n \alpha_i H(m_i) \in G_0$.

$VA(\vec{pk} \in G_1^n, \vec{m} \in \mathcal{M}^n, \vec{\sigma} \in G_0^n)$: $e(\sigma_{ag}, g_1) \stackrel{?}{=} \prod_{i=1}^n e(H(m_i), pk_i)$

$LHS = e(\sum_{i=1}^n \alpha_i H(m_i), g_1) = \prod_{i=1}^n e(H(m_i), g_1)^{\alpha_i} = \prod_{i=1}^n e(H(m_i), pk_i) = RHS$

For multisignatures $m_1 = m_2 = \ldots = m$, VA eq^n is $e(\sigma_{ag}, g_1) = e(H(m), \overrightarrow{apk})$ $\Sigma_i pk_i$.

Rogue Public Key Attack

Consider Bob with public key $pk_B := u_B \in G_1$. Adversary A wants to make it seem as if Bob has signed on $m \in M$ which Bob did not sign.

1. A chooses $\alpha \xleftarrow{\$} \mathbb{Z}_q$, $u = \alpha g_1$, and set $pk_A = u - u_B$
   (Here, A doesn't know sk_{A} since α_B is unknown.)

2. Aggregate public key for A and B is $(u - u_B) + u_B = u$.
   A can sign a multisignature $\sigma_{ag} = \alpha H(m)$, and then claim A and B signed it.

To secure the scheme from such an attack,

1. Message Augmentation: $S(\alpha, m) = \alpha H(m, pk)$. This gets rid of multisignatures and the verification equation in VA becomes
   $e(\sigma_{ag}, g_1) \stackrel{?}{=} \prod_{i=1}^{n} e(H(m, pk_i), pk_i)$
2. Show a ZKPoK of α corresponding to the public key.
   This keeps the multisignature optimization, but computing ZKPoK is not easy.

Anonymous Credentials

Camenisch et. al. (2004)

$e : G_1 \times G_1 \to G_T$ (symmetric pairing)

$(q, G_1, G_T, g_1, g_T, e)$

KeyGen(): $x, y \xleftarrow{\$} \mathbb{Z}_q$, $sk \leftarrow (x, y)$, $pk \leftarrow (X = xg_1, Y = yg_1)$

Signature(sk, m) : 1. $a \xleftarrow{\$} G_1$
2. Output $\sigma \leftarrow (a, ya, (x + mxy)a)$
(credential)

attribute
practically, com(m) and ZKPoK of correctness of com(m)

Verify $(pk = (X, Y), m, \sigma = (a, b, c))$

1. $e(a, Y) \stackrel{?}{=} e(g_1, b)$
2. $e(X, a)\, e(X, b)^m \stackrel{?}{=} e(g_1, c)$

1. $e(a, Y) = e(a, yg_1) = e(a, g_1)^y = e(ya, g_1) = e(Y, g_1)$

2. $e(X, a) \cdot e(X, b)^m = e(xg_1, a) \cdot e(xg_1, ya)^m$
$$= e(g_1, a)^{x+mxy} = e(g_1, (x+mxy)a) = e(g_1, c).$$

Claim. If $\sigma = (a, b, c)$ is a valid credential on m, then $\sigma^r \triangleq (a^r, b^r, c^r)$ is also a valid credential for all $r \in G_1$.

This gives unlinkability of signatures

LRSW Assumption

Suppose $G = \langle g \rangle$ is a group chosen by the setup algorithm corresponding to pairing-friendly elliptic curve groups. Let $X, Y \in G$ s.t. $X = xg, Y = yg$. Let $\mathcal{O}_{X,Y}$ be an oracle that on input a message m on $\mathbb{Z}_q$ outputs a triple (a, a^y, a^{x+mxy}) for $a \xleftarrow{\$} \mathbb{Z}_q$. Then $\forall$ PPT adversaries $\mathcal{A}$, the probability that $\mathcal{A}$ outputs $(m, (a, b, c))$ s.t. m was not queried to $\mathcal{O}$, $m \neq 0$, $a \in G$, $b = a^y$ and $c = a^{x+mxy}$ is negligible.

Security Proof

Claim: Let $a = g^\alpha$, $b = g^\beta$, $c = g^\gamma$ and $(m, (a, b, c))$ satisfies the verification equations. Then $\beta/\alpha = y$, $\gamma/\alpha = x + mxy$.

Proof. $e(a, Y) = e(g, b) \Rightarrow e(\alpha g, yg) = e(g, \beta g) \Rightarrow \beta = \alpha y$ or $y = \beta/\alpha$

$$e(X, a)\, e(X, b)^m = e(g, g)^{x\alpha + mx\beta} = e(g_1, c) = e(g_1, g_1)^\gamma$$

$$\Rightarrow \gamma = x\alpha + mx(\alpha y) \text{ or } \frac{\gamma}{\alpha} = x + mxy.$$

Identity-Based Encryption

Definition: An identity-based encryption scheme $\mathcal{E}_{id}$ = (Setup, KeyGen, Enc, Dec) where:

1. (mpk, msk) ← Setup() [Generates Trudy / trusted authority's keys]
2. sk_{id} ← KeyGen(msk, id). [Generates Bob's private key]
3. c ← Enc(mpk, id, m) [Done by Alice]
4. m ← Dec(sk_{id}, c) [Done by Bob]

s.t. $\Pr[\text{Dec}(\overbrace{\text{KeyGen}(msk, id)}^{\text{secret key}}, \text{Enc}(\overbrace{mpk, id}^{\text{public key}}, m)) = m] = 1$

Semantic Security of IBE Scheme

An adversary $\mathcal{A}$ who obtains secret keys for a polynomial number of identities of his choice should not be able to break the semantic security of another identity.

Attack Game

$\mathcal{E}_{id} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ is defined over identities $id \in ID$, message space $\mathcal{M}$, ciphertext space $\mathcal{C}$.

1. The challenger $\mathcal{C}$ will compute $(msk, mpk) \leftarrow \text{Setup}()$ and send mpk to $\mathcal{A}$
2. $\mathcal{A}$ will make N key queries for an identity $id_j \in ID$. $\mathcal{C}$ will return $sk_{id_j} \leftarrow \text{KeyGen}(msk, id_j)$.
3. $\mathcal{A}$ will come up with equal length messages m_0, m_1 and send it to $\mathcal{C}$.
4. $\mathcal{C}$ will choose $b \xleftarrow{\$} \{0,1\}$ and compute $c \leftarrow \text{Enc}(mpk, id, m_b)$ where $id \notin \{id_1, \ldots, id_N\}$, and send (c, id)
5. $\mathcal{A}$ will return $b' \in \{0,1\}$. If $b = b'$, then $\mathcal{A}$ wins

We define $\text{Adv}^{IBE}_{\mathcal{A}, 1^n} \triangleq \Pr[b = b']$. An IBE scheme is secure if $\text{Adv}^{IBE}_{\mathcal{A}, 1^n} \leq \frac{1}{2} + \text{negl}()$.

Instantiation

1. $e : G_0 \times G_1 \longrightarrow G_T$ be an asymmetric group.
2. A symmetric cipher $\mathcal{E}_s = (\text{Enc}_s, \text{Dec}_s)$
3. Hash function $H_0 : ID \rightarrow G_0$
   $H_1 : G_1 \times G_T \longrightarrow K$

<u>Setup()</u>:	<u>KeyGen(msk, id)</u>	<u>Enc(mpk, id, m)</u>	<u>Dec(sk_id, C)</u>
$\alpha \xleftarrow{\$} \mathbb{Z}_q$	$sk_{id} := (H_0(id))^{\alpha}$	$\beta \xleftarrow{\$} \mathbb{Z}_q$	$(c, \omega_1) \leftarrow C$
$u_1 \leftarrow g_1^{\alpha}$	Send sk_{id} to Bob.	$\omega_1 \leftarrow g_1^{\beta}$	$z \leftarrow e(sk_{id}, \omega_1)$
$msk := \alpha$, $mpk := u_1$		$z \leftarrow e(H_0(id), u_1^{\beta})$	$k \leftarrow H_1(\omega_1, z)$
		$k \leftarrow H_1(\omega_1, z)$	$m \leftarrow D_s(c, k)$
		$c \leftarrow \text{Enc}_s(m, k)$	
		$C \leftarrow (c, \omega_1)$	

For the scheme to be correct, the encryption and decryption key must be equal. Since H_1 is deterministic, it should have same inputs. Thus, the z computed should be the same. But

$$z_{Enc} = e(H_0(id), u_1^{\beta}) = e(H_0(id), g_1^{\alpha\beta}) = e((H_0(id))^{\alpha}, g_1^{\beta}) = e(sk_{id}, w_1) = z_{Dec}$$

<u>Decision Bilinear Diffie-Hellman (Decision BDH) Assumption</u>

Given random elements $g_0^{\alpha}, g_0^{\tau}, g_1^{\alpha}, g_1^{\beta}$, the quantity $e(g_0, g_1)^{\alpha\beta\tau}$ is indistinguishable from a random element in G_T.

<u>Security of IBE Scheme</u>

If decision BDH holds for e, H_0 is modeled as an RO, H_1 is a secure KDF and $\mathcal{E}_s = (E_s, D_s)$ is semantically (CPA) secure, then $\mathcal{E}_{id}$ is secure.

Let A be an adversary attack the $\mathcal{E}_{id}$ scheme using the IBE Attack Game. Assume A issues at most Q_k key queries and Q_{ro} random oracle queries. Then, there exists a decision BDH adversary B_e, a KDF adversary B_{KDF}, a symmetric key adversary B_s where B_e, B_{KDF} and B_s are elementary wrappers around A such that $Adv^{IBE}_{A, 1^n} \leq (Q_{ro}+1) Adv^{DBDH}_{B_e, 1^n}$

<u>Proof</u> B_e receives a DBDH instance $(u_0 = g_0^{\alpha}, u_1 = g_1^{\alpha}, v_0 = g_0^{\tau}, w_1 = g_1^{\beta}, z)$

1. B_e calls A with $mpk = u_1$ $(msk = \alpha)$.
2. When A queries $H_0(id^{(j)})$, B_e responds as follows:
   a. If $j \neq l$, B_e returns $H_0(id^{(j)}) \leftarrow g_0^{\rho_j}$, where $\rho_j \xleftarrow{\$} \mathbb{Z}_q$
   b. If $j = l$, B_e returns $H_0(id^{(j)}) \leftarrow v_0$
3. When A queries key for $id^{(j)}$, B_e responds as follows:
   a. If $j \neq l$, B_e returns $sk_{id^{(j)}} \leftarrow (H_0(id^{(j)}))^{\alpha} = g_0^{\rho_j \alpha} = u_0^{\rho_j}$
   b. If $j = l$, B_e aborts.
4. When A makes an encryption query (id, m_0, m_1) as in the IBE attack game, B_e chooses $b \xleftarrow{\$} \{0,1\}$. If $id = id^{(l)}$, then B_e does $H_0(id) = v_0$, $k = H_1(w_1, z)$, $c = E_s(k, m_b)$ and send (c, w_1) to A.
5. If $b = b'$, B_e outputs <u>YES</u>, else B_e outputs <u>NO</u>.

$\mathcal{B}_{dDDH}$'s success probability/advantage is at least $\frac{1}{2}$ of $\mathcal{A}_{IBE}$'s advantage if $id = id^{(l)}$. Hence,

$$Adv_{\mathcal{B}}^{dDDH} \geq \frac{1}{2} \cdot \frac{1}{Q_m + 1} Adv_{\mathcal{A}}^{IBE}$$

$$\Rightarrow Adv_{\mathcal{A}}^{IBE} \leq 2(Q_m + 1) Adv_{\mathcal{B}}^{dDDH}$$

Lattice - Based Cryptography

In LWE, $\mathbb{Z}_q$ is represented as integers in $(-q/2, q/2)$.

Lₒₒ norm: The L_∞ norm of $[a_1 \dots a_n]^T$ is $L_\infty([a_1 \dots a_n]^T) = \max_{i=1}^{n} |a_i|$

B-bounded Distribution: χ_B is a B-bounded distribution if $Pr[\|e\|_\infty \leq B] = 1$. For example, the uniform distribution on $\{-B, \dots, 0, \dots, B\}$

$e \leftarrow \chi_B$

$LWE(\overset{\#unknowns}{n}, \underset{\#eq^n s}{m}, \overset{size\ of\ field}{q}, \chi_B)$ (search-LWE)

Let $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $s \leftarrow \mathbb{Z}_q^n$, $e \leftarrow \chi_B^m$. Given $(A, A\underline{s} + \underline{e})$, find $\underline{s}'$ s.t. $\|A\underline{s}' - (A\underline{s} + \underline{e})\|_\infty \leq B$.

Decision LWE

It should be hard to distinguish vectors close to the image of A from random vectors in $\mathbb{Z}_q^n$. ($A\underline{s} + \underline{e}$ and $A\underline{s}$)

Regev Encryption

KeyGen(): $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $s \xleftarrow{\$} \mathbb{Z}_q$, $e \xleftarrow{\$} \chi_B^m$, choose B s.t. $q/4 > mB$.

$b := A\underline{s} + \underline{e}$. $sk := \underline{s}$. $pk := (A, b)$

Encrypt $(pk, x \in \{0,1\})$:

$r \xleftarrow{\$} \{0,1\}^m$

$c_0 = r^T A$, $\underline{c}_1 = r^T b + \lfloor q/2 \rfloor x$

$c = (c_0, \underline{c}_1)$

Decrypt $(sk, (c_0, c_1))$:

$\tilde{x} = \underline{c_1 - c_0 \underline{s}}$

if $\tilde{x} < q/4$, $x = 0$

else, $x = 1$

<u>Correctness</u> [Security uses LWE and hybrid arguments]

$\underline{c}_1 - c_0 \underline{s}$

$= \underline{r}^T \underline{b} + \lfloor q/2 \rfloor x - \underline{r}^T A \underline{s}$

$= \underline{r}^T \underline{e} + \lfloor \frac{q}{2} \rfloor x \leq mB + \lfloor \frac{q}{2} \rfloor x.$

Thus, $x = 0 \Rightarrow \underline{c}_1 - c_0 \underline{s} \leq mB < q/4$

and $x = 1 \Rightarrow \underline{c}_1 - c_0 \underline{s} = \underline{r}^T \underline{e} + \lfloor \frac{q}{2} \rfloor \geq q/4$

<u>Lattices</u>: A set of points in $\mathbb{Z}^n$ that are integer linear combinations of basis elements.

Let $B = \{\underline{b}_1, \ldots, \underline{b}_n\}$ be a linearly independent basis set. Then,

$\mathcal{L}(B) \triangleq \{\sum_{i=1}^{n} a_i \underline{b}_i \mid a_i \in \mathbb{Z}\}.$

1. SVP: Finding the shortest lattice vector in $\mathcal{L}(B)$ (SVP^{∞} is NP hard)
2. CVP: Given $\underline{t} \in \mathbb{Z}^n$, find $\underline{v} \in \mathcal{L}(B)$ that minimizes $\|\underline{t} - \underline{v}\|$
3. Approximate Versions: Setting γ as an approximation factor.
   - γ-SVP: Find $\underline{t} \in \mathcal{L}(B)$ s.t. $\underline{t} = \gamma \underbrace{\lambda_1(\mathcal{L}(B))}_{\text{Length of shortest vector in } \mathcal{L}(B)}$
   - γ-CVP. Find $\underline{v} \in \mathcal{L}(B)$ s.t. $\|\underline{t} - \underline{v}\| \leq \gamma$ (dist. b/w closest vector and $\underline{t}$).