A group is a set G along with a binary op. '$\circ$' for which the following conditions hold:

- closure:

  $\forall g, h \in G, \ g \circ h \in G$

- Existence of Identity:

  $\exists \ e \in G$ s.t $\forall g \in G \quad e \circ g = g \circ e = g$

- Existence of Inverses:

  $\forall g \in G \left(, \exists h \in G \text{ s.t } g \circ h = h \circ g = e \right)$

- Associativity:

  $\forall g_1, g_2, g_3 \in G \Rightarrow (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$

when G has finite number of elements we denote it as $|G| \Rightarrow$ order of the group

We call a group, Abelian Group if

* commutativity:

$\forall g, h \in G \Rightarrow g \circ h = h \circ g$

$(\mathbb{Z}_n^*, * \text{ modulo } n)$ is an abelian group under multiplication modulo N

$\mathbb{Z}_n^* = \{ a \mid a \in \{1, \ldots, n-1\} \text{ \& } \gcd(a,n)=1 \}$

proof closure:

* If $a \in \mathbb{Z}_n^*$, $b \in \mathbb{Z}_n^*$ $ab \bmod n \in \mathbb{Z}_n^*$

$\gcd(a,n)=1 \text{ \& } \gcd(b,n)=1 \Rightarrow \gcd(a \cdot b, n)=1$

$\Rightarrow \gcd(ab \bmod n, n))=1$

proof of identity element:

* 1 is the identity element

proof of commutativity:

* $a \cdot b = b \cdot a \bmod n$ so multiplication mod n is commutative

Proof of assocrativity:

$a \star (b \star c) = (a \star b) \star c \mod n$

Proof of existence of inverse:

$\forall a \in \mathbb{Z}_N^{\star}, \quad \exists a^{-1}$ s.t

$a \cdot a^{-1} = 1 \mod n$

If $\gcd(a,n)=1$, $\exists a^{-1}$ s.t

$a \cdot a^{-1} = 1 \mod n$

$a \cdot x = 1 \mod n$

$\Rightarrow$ if $\gcd(a,n)=1$ than

$\exists x, y$ s.t

$\underline{a x + n y} = 1$

$\Rightarrow a \cdot x = 1 - ny$

$\Rightarrow a \cdot x = 1 \mod n$

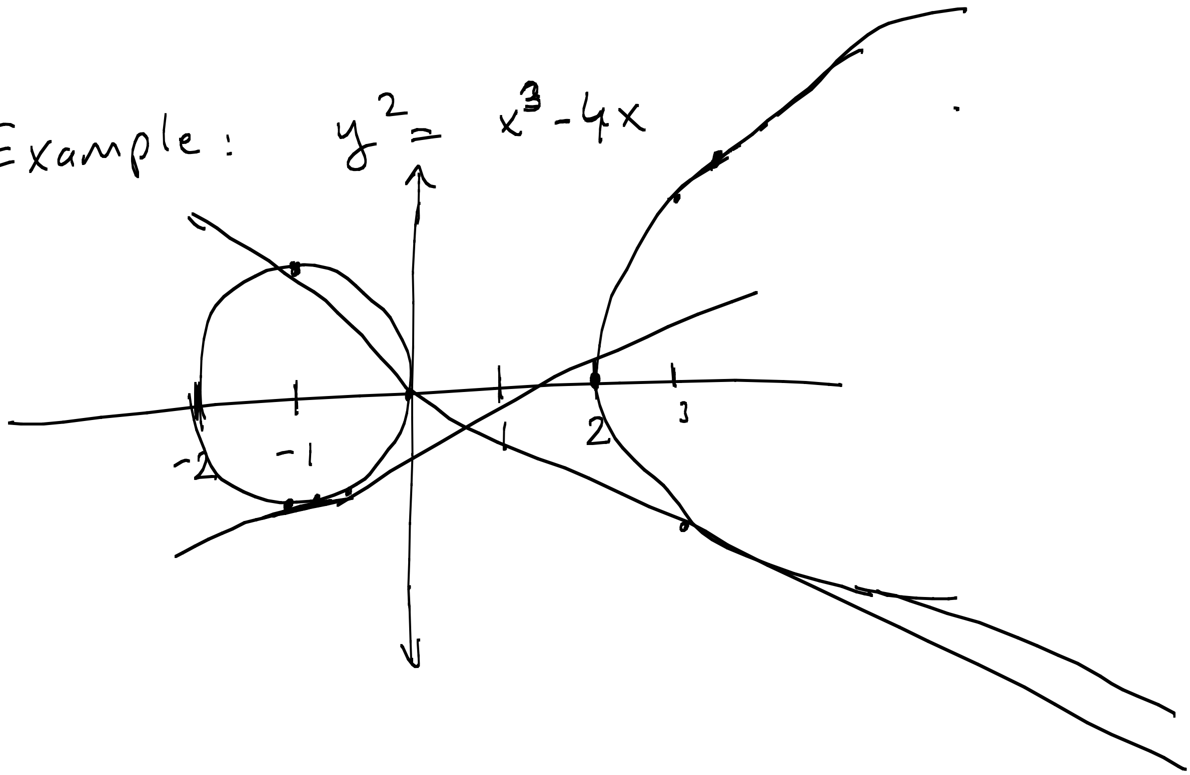
Let $a, b \in \mathbb{R}$ s.t. $4a^3 + 27b^2 \neq 0$

A non-singular elliptic curve is the set E of solutions $(x,y) \in \mathbb{R} \times \mathbb{R}$ to the equation

$$y^2 = x^3 + ax + b$$

plus special point $\mathcal{O}$ called the point at infinity

Example: $y^2 = x^3 - 4x$



Define binary op '+' to make (E,'+') Abelian group. 'O' is the identity element

suppose $P, Q \in E$ where $P = (x_1, y_1)$, $Q = (x_2, y_2)$

cases : 1) $x_1 \neq x_2$

2) $x_1 = x_2$ & $y_1 = -y_2$

3) $x_1 = x_2$ & $y_1 = y_2$

$P + Q = R$ where $R = (x_3, y_3)$

s-t $\boxed{x_3 = \lambda^2 - x_1 - x_2}$, $y_3 = \lambda(x_1 - x_3) - y_1$

$\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$

(x_1, y_1) & (x_2, y_2) = $y = \lambda x + c$

$(\lambda x + c)^2 = x^3 + ax + b$

case 2: $(x_1, y_1) + (x_1, -y_1) = O$

case 3:
$$(x_1, y_1) + (x_1, y_1') = (x_3, y_3)$$

$$x_3 = \lambda^2 - 2x_1$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

1) Addition is closed on the set E

2) // is commutative

3) O is an identity element

4) Every point on E has an inverse.

$P = (x, y) \in E$ $P^{-1} = (x, -y) \in E$

5) Associativity is satisfied.

Let $p > 3$ be prime. EC $y^2 = x^3 + ax + b$ over $\mathbb{Z}_p$ is the set of solutions $(x, y) \in \mathbb{Z}_p$ s.t

$$y^2 = x^3 + ax + b \pmod p$$

where $a, b \in \mathbb{Z}_p$ & $4a^3 + 27b^2 \neq 0 \mod p$

plus a special point $\mathcal{O}$ called the point at infinity.

$P = (x_1, y_1)$, $Q = (x_2, y_2)$

If $x_1 = x_2$ & $y_1 = -y_2$ then

$$P + Q = \mathcal{O}$$

else

$$x_3 = \lambda^2 - x_1 - x_2 \mod p$$

$$y_3 = \lambda (x_1 - x_3) - y_1 \mod p$$

$$\lambda = \begin{cases} (y_2 - y_1)(x_2 - x_1)^{-1} \mod p & \text{if } P \neq Q \\ (3x_1^2 + a)(2y_1)^{-1} \mod p & \text{if } P = Q \end{cases}$$

Example : $y^2 = x^3 + x + 6$ over $\mathbb{Z}_{11}$

x	$x^3+x+6 \bmod 11$	QR ?	y
0	6	$y^2 \equiv 6 \bmod 11$ NO	
1	8	$y^2 \equiv 8 \bmod 11$ NO	
2	5	$y^2 \equiv 5 \bmod 11$ Yes	(4, 7)
3	3	$y^2 \equiv 3 \bmod 11$ Yes	5, 6
4	8	$y^2 \equiv 8 \bmod 11$ NO	
5	4	$y^2 \equiv 4 \bmod 11$ Yes	2, 9
6	8	$y^2 \equiv 8 \bmod 11$ NO	
7	4	Yes	2, 9
8	9	Yes	3, 8
9	7	NO	

10	4	$y^2 = 4 \bmod 11$ y	2, 9

The order of E (including O) is 13.

Any group of prime order is cyclic.

This means you could represent each element of E as $k.\alpha$ where $\alpha \in E$

$k \in \mathbb{Z}_{13}$ — α: generator of the group

$2.\alpha$ where $\underline{\alpha = (2,7)}$

$2.\alpha = (2,7) + (2,7) = (5,2)$

$3.\alpha = (8,3) = (2,7) + (2,7) + (2,7) = (5,2) + (2,7)$

$4.\alpha = (10,2)$

$5\alpha = (3,6)$

$6\alpha = (7,9)$, $7\alpha = (7,2)$

$8\alpha = (3,5)$, $9\alpha = (10,9)$

$10\alpha = (8,8)$

$11\alpha = (5,9)$, $12\alpha = (2,4)$ $0.\alpha = \mathcal{O}$

I can represent $(7,2)$ as $7.\alpha$

El-Gamal over $\mathbb{Z}_p$

$a \to$ secret key $g^a \bmod p$, p is publickey

generator $\mathbb{Z}_p^*$

$$E(m) = (g^b, M.(g^a)^b) = C$$

$$\underbrace{g^b}_{u}, \underbrace{M.(g^a)^b}_{v}$$

$$D(C) = v.((u)^a)^{-1} = (M.\underbrace{g^{ab})((g^b)^a)^{-1}} = M$$

Analog of El Gamal on EC

Given E over $\mathbb{Z}_p$, choose random $s \in \mathbb{Z}_p$

publish $\boxed{s.B}$ where B is a generator of E

Given m, embed m to E, let's call it P_m

$$E(P_m) = (\underset{\text{random}}{kB},\ \underbrace{P_m + k(sB)}_{c_2}) = C$$

$$\begin{aligned} D(C) = D(kB, c_2) &= c_2 - s\cdot(kB) \\ &= P_m + k\cdot(sB) - s(kB) \\ &= P_m \end{aligned}$$

An elliptic curve E defined over $\mathbb{Z}_p$ will have roughly p points.

More precisely

$$p+1-2\sqrt{p} \leq |E| \leq p+1+2\sqrt{p}$$

First $1 \le m \le M$ choose k s.t.

$k \cdot m < p$

for (j=1; j<k; j++)

{ $m' = mk + j$

$y' = m'^3 + am'^2 + b \mod p$

if y' is QR then

return $(x, \sqrt{y})$ ↳ mod p

}

Goal: Create PK with following four algorithms to realize IBE

{ params, mkey } ← setup (k) (1^k)

↳ description of Message space

" " [illegible] space

M key will be kept secret.

Extract:

$d \leftarrow$ Extract (params, Mkey, ID)

d → private key for ID

ID → any string

$ID \in \{0,1\}^*$

Encrypt:

$c \leftarrow$ Enc (ID, params, m)

c → $c \in C$

m → $m \in M$

Decrypt:

$m \leftarrow$ Dec (params, c, d)

d → private key for ID

$\forall m \in M \quad \text{Dec}(\text{params}, \text{Enc}(\text{params}, ID, m), d) = m$

---

security game:

Setup: The challenger takes a security parameter k and runs set up algoritmn keeps Mkey secret, sends params to Adversary

Phase 1: Adv. issues $q_1 \dots q_m$

each q_i is answered as follows

If $q_i = \langle ID_i \rangle$

$d_i \leftarrow \text{Extract}(params, Mkey, ID_i)$

else if $q_i = \langle ID_i, C_j \rangle$

$m_j \leftarrow \text{Decrypt}(params, C_j, d_i)$

challenge: Adv. A creates a challenge

(m_0, m_1, ID) where $m_1, m_2 \in M$

ID has not been queried before

challenger responds with

$b \xleftarrow{R} \{0,1\}$

$c \leftarrow \mathrm{Enc}(\text{params}, ID, m_b)$

phase 2: Adv. issues $q_{m+1}, \ldots, q_n$

each q_i is one of the followin

- $\langle ID_i \rangle$ where $ID_i \neq ID$
- $\langle ID_i, C_i \rangle$ where $\langle ID_i, C_i \rangle \neq \langle ID, C \rangle$

queries are answered as in phase 1

Guess: Adv guesses b' & wins if $b' = b$

Adv. A's advantage in attacking E as the following function

$$\mathrm{Adv}_{E,A}(k) = \left| \Pr[b = b'] - \frac{1}{2} \right|$$

---

we say that IBE system E is semantically secure against an adaptive chosen ciphertext attack if for any poly. time IND-ID-CCA adversary A the function $Adv_{E,A}(k)$ is negligible.

E is called IND-ID-CCA secure.

Main components of IDE scheme we discuss. We will use bilinear maps.

Let G_1, G_2 groups of order q where q is a large prime number

$\hat{e} : G_1 \times G_1 \rightarrow G_2$, $\hat{e}$ must satisfy following:

1) Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P,Q)^{ab}$ for all $P, Q \in G_1$ $a, b \in \mathbb{Z}$

2) Non-degenerate: $\hat{e}$ does not map all pairs in $G_1 \times G_1$ to identity in G_2.

$\Rightarrow$ If P is generator G_1 then $\hat{e}(P,P)$ is a generator of G_2

3) Computable: $\hat{e}(P, Q)$ is efficiently computable for any $P, Q \in G_1$

In practice G_1 will be $E/\mathbb{F}_p$

G_2 will be $\mathbb{F}_{p^2}^*$

Notes on this bilinear maps

D.L. in G_1 is no harder than D.L. in G_2

DL: aQ, Q it is hard to find a

Discrete Logarithm problem

$g \in \mathbb{Z}_p^*$ $g^a \bmod p$ it is hard to compute a

For $Q = \alpha P$, given P, we want to find $\boxed{\alpha \in \mathbb{Z}_q}$

Let $g = \hat{e}(P,P)$ & $h = \hat{e}(Q,P)$ than

$h = g^{\alpha}$

$$\boxed{\begin{aligned} \hat{e}(\alpha P, P) &= (\hat{e}(P,P))^{\alpha} \\ h &= g^{\alpha} \end{aligned}}$$

Given g^{a}, g^{b}, g, It is hard to compute g^{ab}. (computational D.H assumption)

$\langle g^{a}, g^{b}, g^{ab} \rangle$, $\langle g^{a}, g^{b}, g^{c} \rangle$

$A \xrightarrow{g^{a}} B$

$A \xleftarrow{g^{b}} B$

g^{ab} g^{ab}

D.H. Key Exchange

The Bilinear Diff. Hell Assumption (BDH)

Let G_1, G_2 be two groups of prime order q. Let $\hat{e} : G_1 \times G_1 \to G_2$ (admissible bilinear map.)

Let P be a generator of G_1

BDH problem is defined as

Given $\langle P, aP, bP, cP \rangle$ for some $a, b, c \in \mathbb{Z}_q^*$

compute $w = \hat{e}(P,P)^{abc} \in G_2$

Alg. A has advantage ε in solving

BDH over $(G_1, G_2, \hat{e})$

if $Pr[A(P, aP, bP, cP) = \hat{e}(P,P)^{abc}] \geq \varepsilon$

Note: If you can solve CDH in G_1 or G_2

than you can solve BDH

Assume CDH is easy in G_1

Given aP, bP, I can compute abP

than I can compute

$w = \hat{e}(abP, cP) = \hat{e}(P,P)^{abc}$

Basic Identity Based Encryption

set up: Given $k \in \mathbb{Z}^+$

1) Run $\bar{G}(k)$ to get $(P, q, G_1, G_2, \hat{e})$

2) $P_{pub} = s.P \qquad s \xleftarrow{R} \mathbb{Z}_q^*$

3) Choose $H_1 : \{0,1\}^* \to G_1^*$

$H_2 : G_2 \to \{0,1\}^n$

$\mathcal{M} = \{0,1\}^n, \quad C = G_1^* \times \{0,1\}^n$

params = $\langle q, G_1, G_2, \hat{e}, P, n, P_{pub}, H_1, H_2 \rangle$

Extract: for any $ID \in \{0,1\}^*$

1) $Q_{ID} \leftarrow H_1(ID) \quad (Q_{ID} \in G_1^*)$

2) $d_{ID} \leftarrow \underset{\text{Mkey}}{\underbrace{s}} \cdot Q_{ID}$

Encrypt(params, m, ID)

{ 1) $Q_{ID} \leftarrow H_1(ID)$

2) $r \xleftarrow{R} \mathbb{Z}_q^*$

3) $g_{ID} \leftarrow \hat{e}(Q_{ID}, P_{pub}) \in G_2$

4) $c = \langle rP, M \oplus H_2(g_{ID}^r) \rangle$

}

Decrypt(params, c, d_{ID})

{ $c = \langle U, V \rangle$ $M = V \oplus H_2(\hat{e}(d_{ID}, U))$

}

The above system is secure in IND-ID-CPA.

$$\hat{e}(d_{ID}, U) = \hat{e}(sQ_{ID}, rP) = \hat{e}(Q_{ID}, P)^{s.r}$$

$$= \hat{e}(Q_{ID}, P_{pub})^r$$

$$= g_{ID}^r$$

$$\boxed{V = M \oplus H_2(g_{ID}^r)}$$

$$V \oplus H_2(\hat{e}(d_{ID}, U)) = V \oplus H_2(g_{ID}^r) = M$$

$$= M \oplus H_2(g_{ID}^{r}) \oplus H_2(g_{ID}^{r}) = M$$

we can use Fujisaki – Okamoto scheme to convert BasicIdent to IBE secure against IND – ID – CCA

$$E_{pk}^{hy}(M) = < E_{pk}(\sigma, H_3(\sigma, M)), H_4(\sigma) \oplus M >$$

$E_{pk}(M, r)$ is enc of M using random bits of r

Different Models for searching data privately

- protect data vs protect access patterns
- searching on private-key encrypted data
  * user encrypts the data
  * encrypted data & additional data structures

are stored in the cloud

- searching public-key encrypted data
  - next class.

- PIR $\Rightarrow$ hides which data has been accessed. PIR requires linear time search time

Notations

- $\Delta = \{w_1, \ldots, w_d\}$ // set of all possible words
- $D_i \subseteq 2^{\Delta}$ // Document
- $D = \{D_1, \ldots, D_n\} \subseteq 2^{2^{\Delta}}$ // Document set
- $id(D_i)$ is the identifier of Document D.
- Given query w, $D(w)$ is the sorted list of ids for Docs that contain w.
- $(D(w_1) \ldots, D(w_n))$ is the access pattern of the client

- $x \leftarrow A(y)$ output of an algorithm
- '||' concatenation
- $v : N \to N$ negligible if
  $\forall$ poly - $p(k)$ for sufficiently large k $v(k) < \frac{1}{p(k)}$

Model

* single-user:
  Owner has $D = \{D_1 \ldots, D_n\}$
  sends encrypted D to server S.
  and query encrypted D stored on S.

* multi-user:
  owner O, Server S
  set of users N, $G \subseteq N$
  where G is allowed to do search on encrypted D.

Not discussed see paper for more details

* S is honest but currious.

* construction uses symm. key enc. scheme

* pseudo-random permutation (e.g., AES)
* pseudo-random function (e.g., keyed Hash function)

---

searchable Symm. Enc. scheme

security parameter.

$K \leftarrow \text{Keygen}(1^k)$ ← run by user

in poly-time with respect to k.

$I \leftarrow \text{BuildIndex}(K, D)$

$T_w \leftarrow \text{Trapdoor}(K, w)$ returns a trapdoor value for search.

$D(w) \leftarrow \text{Search}(T_w, I)$

↳ run by server

security definition idea:

- For any two adversarially consisten history with equal length and trace, no adversary can distinguish the view of one from the other with non-negligbly better than random.

- This def. assume revealing search pattern is ok.

SSE that is adaptive security

KeyGen(1^k)
{
    $s \xleftarrow{R} \{0,1\}^k$
    $K \leftarrow s$
    return K;
}

Build Index (K, D) ← set of unique key-words in D

{

Build Δ' from D;

Build $D(w_i)$ for each $w_i \in \Delta'$;

For each $w_i \in \Delta'$

{

for $1 \le j \le |D(w_i)|$

$T[\pi_s(w_i || j)] = \underline{id(D_{i,j})}$

// T is Look-up Table of size m

// $\pi : \{0,1\}^k \times \{0,1\}^p \to \{0,1\}^p$ is pseudo-random permutation

//

}

Let $m' = \sum_{w_i \in \Delta'} |D(w_i)|$

Let $m = \{$size of largest (max) document in $D\} \times n$

If $m > m'$

for remaing $(m - m')$ entries

put random $id(D)$ for exactly

max entries.

- Address fields would be set to random values

return $I \leftarrow T$;

Trapdoor (w)

{

$T_w \leftarrow (\pi_s(w||1), \ldots \pi_s(w||n))$;

return T_w;

}

search (I, Tw)

{

for $1 \le i \le max$

retrieve id $= T[T_{w_i}]$

return all retrieved ids

}

Public Key Encryption with Search (PEKS)

Goal: Bob wants to send message to Alice.

$$E_{A_{pub}}(msg), \; PEKS(A_{pub}, w_1) \; \ldots \; PEKS(A_{pub}, w_k)$$

Goal is to enable Alice to generate T_w (trapdoor) for word w s.t. server can learn all the messages with word w without disclosing anything else.

Formal Definition

1) Keygen(s): Takes a security param s and generates A_{pub}, A_{priv}.

2) $PEKS(A_{pub}, w)$ produces searchable enc. S.

3) $T_w \leftarrow Trapdoor(A_{priv}, w)$

w produces a searchable encryption of w.

4) Test (A_{pub}, S, T_w) : Given Alice's A_{pub}, searchable enc. $S = PEKS(A_{pub}, w')$ and T_w outputs yes if $w = w'$

else outputs no.

Security Definition:

1) Challenger runs the KeyGen(s) to get A_{pub}, A_{priv}. and sends A_{pub} to attacker

2) Attacker queries any w of his choice and gets T_w

3) Attacker sends $\{w_0, w_1\}$ such that T_{w_0} or T_{w_1} never queried.
challenger chooses $b \xleftarrow{R} \{0,1\}$
and sends T_{w_b}

4) Attacker asks more queries not related to T_{w_0} or T_{w_1}

5) Attacker predicts b' if $b = b'$
attacker wins

$$Adv_A(s) = \left| Pr[b = b'] - \frac{1}{2} \right|$$

we say that PEKS is semantically secure against an adaptive chosen keyword attack if for any P.T. attacker A the $Adv_A(s)$ is negligible.

PEKS $\Rightarrow$ IBE

Thm: A non-interactive PEKS that is semantically sec. against adaptive chosen keyword attack implies the existence of a chosen ciphertext secure IBE (IND_ID_CCA).

Proof Idea:

Given a PEKS (Keygen, PEKS, Trapdoor, TEST)

then we construct the IBE system

as follows:

1) Run the keygen for PEKS
get A_{pub}, A_{priv}

2) A_{priv} is the master key for
the IBE scheme

3) Generate the IBE private key
associated with public key
$X \in \{0,1\}^*$

$$d_X = [\, \text{Trapdoor}(A_{priv}, X \| 0), \; \text{Trapdoor}(A_{priv}, X \| 1) \,]$$

4) Encrypt:
Encrypt a bit $b \in \{0,1\}$
using public key $X \in \{0,1\}^*$

$C = \text{PEKS}(A_{pub}, X \| b)$

5) Given C, use $d_X = [d_{X_0}, d_{X_1}]$

output 0 if $\text{Test}(A_{pub}, C, d_{X_0}) =$ 'Yes'

output 1 if $\text{Test}(A_{pub}, C, d_{X_1}) =$ 'Yes'

else output 'error'

The above scheme is IND-ID-CCA secure.

It is belived that (IBE $\Rightarrow$ PEKS) is not correct.

Construction of a PEKS

Use two groups G_1, G_2 of a prime order p where both G_1 & G_2 are multiplicative groups.

and a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$

1) $\hat{e}$ is efficiently computable

2) Bilinear: for any integers $x, y \in [1, p-1]$

$$\hat{e}(g^x, g^y) = \hat{e}(g, g)^{x \cdot y}$$

3) Non-degenerate: if g is a generator of G_1, then $\hat{e}(g, g)$ is a generator of G_2.

Given $H_1 : \{0,1\}^* \rightarrow G_1$

$H_2 : G_2 \rightarrow \{0,1\}^{\log p}$

KeyGen (s)
{
Given s, compute p, G_1, G_2

Pick random $\alpha \in \mathbb{Z}_p^*$,
$A_{pub} = g^{\alpha} = h$, $A_{priv} = \alpha$
}

PEKS (A_{pub}, w)
{
$t \leftarrow \hat{e}(H_1(w), h^r)$ for random $\mathbb{Z}_p^*$

output $[\underbrace{g^r}_{A}, \underbrace{H_2(t)}_{B}]$
}

Trapdoor $(A_{priv}, w) = (H_1(w))^{\alpha} \in G_1$

Test (A_{pub}, S, T_w)

$S \to [A, B]$

if $H_2(\underline{\hat{e}(T_w, A)}) = B$ than output 'yes'

else output 'no'

why it works?

$H_1(w) = g^m$

Given $t \leftarrow \hat{e}(H_1(w), h^r)$

$t \leftarrow \hat{e}(g^m, g^{\alpha r}) = \hat{e}(g,g)^{m\alpha r}$

$\Rightarrow B = H_2(\hat{e}(g,g)^{m\alpha r})$

Also consider $\hat{e}(T_w, A) = \hat{e}(H_1(w)^{\alpha}, g^r)$

$= \hat{e}(H_1(w), g)^{\alpha r}$

$= \hat{e}(g^m, g)^{\alpha r}$

$= \hat{e}(g,g)^{m\alpha r}$

$$\Rightarrow \quad H_2(\hat{e}(T_w, A)) = B \quad \text{if } T_w$$

is a trapdoor for w.

It can be proven that above scheme is secure under BDH assumption in the random oracle model.

A construction that can use any P.K. assuming that the searchable keywords are limited. Also we assume that the public key scheme is such that given ciphertext, it is hard to say which PK this ciphertext is associated with. (source-indistinguishability)

When the keyword family Σ is poly size with respect to s, it is easy to construct searchable encryption from any PKE (G, E, D)

generated Enc Dec.

Keygen :

for each $w \in \Sigma$ run $G(s)$

and get PK_w, $Priv_w$

$A_{pub} = \{ PK_w \mid w \in \Sigma \}$

$A_{priv} = \{ Priv_w \mid w \in \Sigma \}$

}

PEKS (A_{pub}, w)

{ Pick random $M \in \{0,1\}^s$

return $(M, E[PK_w, M])$

}

$$\text{Trapdoor}(A_{priv}, w) = Priv_w$$

$$\text{Test}(A_{pub}, \underbrace{S}_{[A,B]}, T_w)$$

{

if $\left(D(Priv_w, B) = D(T_w, B) \right) \stackrel{?}{=} A$ then output 'Yes'

else output 'No'.

}