Note Title 2/21/2013

Basic IBE: that satisfies

IND-ID-CPA.

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

1) Run $\bar{G}(k)$ to get

$(p, q, G_1, G_2, \hat{e})$

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

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

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

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

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

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

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

2) $d_{ID} \leftarrow s \cdot Q_{ID}$

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})

Let $c = \langle U, V \rangle$

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

$$= V \oplus H_2(\hat{e}(sQ_{ID}, rP))$$

$$= V \oplus H_2(\hat{e}(Q_{ID}, P)^{sr})$$

$$\quad \parallel$$

$$= V \oplus H_2(\underbrace{\hat{e}(sP, Q_{Id})}_{g_{ID}}{}^{r})$$

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

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

$$= M$$

The above system is secure IND-ID-CPA

More precisely If $\exists$ Adv. A attacking above scheme $\mathcal{E}(k)$ than $\exists$ Adv. B attacking

B.D.H with at least prob.

$$\frac{2\,\varepsilon(\varepsilon)}{e\cdot(1+q_E)\,q_{H_2}}$$

$$C_1 = M_1^{e} \bmod n$$

$$M_1^{e} \cdot r^{e}$$

$$C_2 = C_1 \cdot R^{e} \bmod n$$

$$D(C_2) = M_1 \cdot r \bmod n$$

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

Fujisaki-Okamoto scheme for general PKE:

Assume $E_{pk}(M, r)$ IND-CPA secure PKE

than $E^{hy}_{pk}(M) = \langle E_{pk}(\sigma, H_3(\sigma, M)), H_4(\sigma) \oplus M \rangle$

is IND-CCA secure in random

oracle model.

Full Indent Descriptron:

Setup: The same as Basic Ident

Add $H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q^*$

$H_4 : \{0,1\}^n \to \{0,1\}^n$

Extract: No change

Encrypt: $M \in \{0,1\}^n$

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

2) $\sigma \xleftarrow{R} \{0,1\}^{n}$

3.) $r \leftarrow H_3(\sigma, M)$

4. $g_{ID} \leftarrow \hat{e}(Q_{ID}, P_{pub})$

$C \leftarrow \left(rP, \sigma \oplus H_2(g_{ID}^{r}), M \oplus H_4(\sigma)\right)$

Decrypt: Given $C = \langle U, V, W \rangle$

1.) If $(U \notin G_1^{*})$ return reject.

2.) $\sigma \leftarrow V \oplus H_2(\hat{e}(d_{ID}, U))$

3) $M \leftarrow W \oplus H_4(\sigma)$

4) $r \leftarrow H_3(\sigma, M)$

5) if $U \neq rP$ then reject
else output M.

Goal: Bob wants to sen message to Alice

$E_{A_{pub}}(msg)$, $PEKS(A_{pub}, w_1)$ . . .
$PEKS(A_{pub}, w_2)$

we want Alice to generate trapdoor(T_w)

for any word w such that a server can learn all the messages with word w without disclosing anything else.

---

1) KeyGen(s) : Takes a security params s and generates A_{pub}, A_{priv}

2) $S = PEKS(A_{pub}, w)$ that produces searchable encryption.

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

4) $Test(A_{pub}, S, T_w)$:

given Alice's PK searchable enc. $S = PEKS(A_{pub}, w')$

$T_w =$ trapdoor (A_{priv}, w)

outputs 'yes' if $w = w'$

else outputs 'no'

Security Definition:

1) challenge runs the KeyGen(s)

to generate A_{pub}, A_{priv}

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 querried.
challenger send T_{w_b} for random bit b

4) Attacker ask more queries not relate 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 somantically secure against an adaptive chosen keyword attack if for any P.T. attacker A if we have that $Adv_A(s)$ is negligible function $f: R \to \{0,1\}$ is negligible if $f(s) < \frac{1}{g(s)}$ for all poly. g and large enough s.

PEKS $\Rightarrow$ IBE

A non-interactive PEKS that is semantic. sec. against adaptive chosen keyword attack implies IND-ID-CCA secure IBE

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

construct IBE system as follows:

1) run the Keygen for PEKS to get A_{pub}, A_{priv}.

2) set A_{priv} as the master key for IBE.

3) Extract:

The IBE private key associated with ID $x \in \{0,1\}^*$

$$d_x = [\text{Trapdoor}(A_{priv}, x \| 0), \text{Trapdoor}(A_{priv}, x \| 1)]$$

4) Encrypt for ID x a bit $b \in \{0,1\}$ as

$$CT = \text{PEKS}(A_{pub}, \underbrace{x \| b}_{ID})$$

5) Decrypt:

given CT

use $d_x = (d_0, d_1)$

output 0 if $\mathrm{Test}(A_{pub}, CT, d_0) =$ (Yes)

output 1 if $\mathrm{Test}'(A_{pub}, CT', d_1) =$ Yes

else error.

The scheme is IND-ID-CCA secure.

It is believed that opposite direction is not true.

$\left(ID \overset{?}{\not\Rightarrow} PEKS \right)$

Construction:

use two groups:

G_1, G_2 of a prime order p

(In this paper, both G_1 and G_2 are multiplicative groups)

and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$

1) $\hat{e}$ is eff. computable

2) Bilinear: for $\forall x, y \in [1..p)$

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

3) Non-degenerate: If g is a generator G_1 then $e(g,g)$ is a generator of G_2

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

$H_2 : G_2 \longrightarrow \{0,1\}^{\log(p)}$

Keygen: Given s, compute p, G_1, G_2

$s \rightarrow$ security param.

pick $\alpha \in \mathbb{Z}_p^*$ $A_{pub} = g^{\alpha}$ $A_{priv} = \alpha$

$\mathrm{PEKS}(A_{pub}, w)$

$\{$

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

$h = g^\alpha$

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

$\}$

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

$\mathrm{Test}(A_{pub}, S, T_w)$

$\{$ Let $S = (A, B)$

if $H_2(e(T_w, A)) = B$ than

output 'yes'

else output 'no'

}

$$B = H_2(t) = H_2(e(H_1(\omega), h^r)) \quad 1$$

$$\stackrel{?}{=} H_2(e(H_1(\omega)^\alpha, g^r)) \quad 2$$

$$H_2(e(H_1(\omega), g^{\alpha r})) \stackrel{?}{=} H_2(e(H_1(\omega)^\alpha, g^r)) \quad 3$$

$$H_2(e(H_1(\omega), g)^{\alpha r}) = H_2(e(H_1(\omega), g)^{\alpha r}) \quad 4$$

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

A construction that can use any trapdoor function assuming that the searchable keywords are limited.

Also we assume that the PK is such that given ciphertext

it is hard to say which PK
this ciphertext is associated with
(denoted as source-indistinguishable)

when the keyword family Σ is poly-size
it is easy to construct searchable encryption
from any (G, E, D)

Keygen {

{

for each $w \in \Sigma$

run $G(s)$, get PK_w, $Priv_w$

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

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

}

PEKS(Apub, w)

{

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

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

)

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

$\text{Test}(A_{pub}, S, T_w)$
$(M, E[Pk_w, M])$

{

If $D(T_w, S) = M$ output 'yes'

else output 'no'

}

Attribute-Based Encryption:

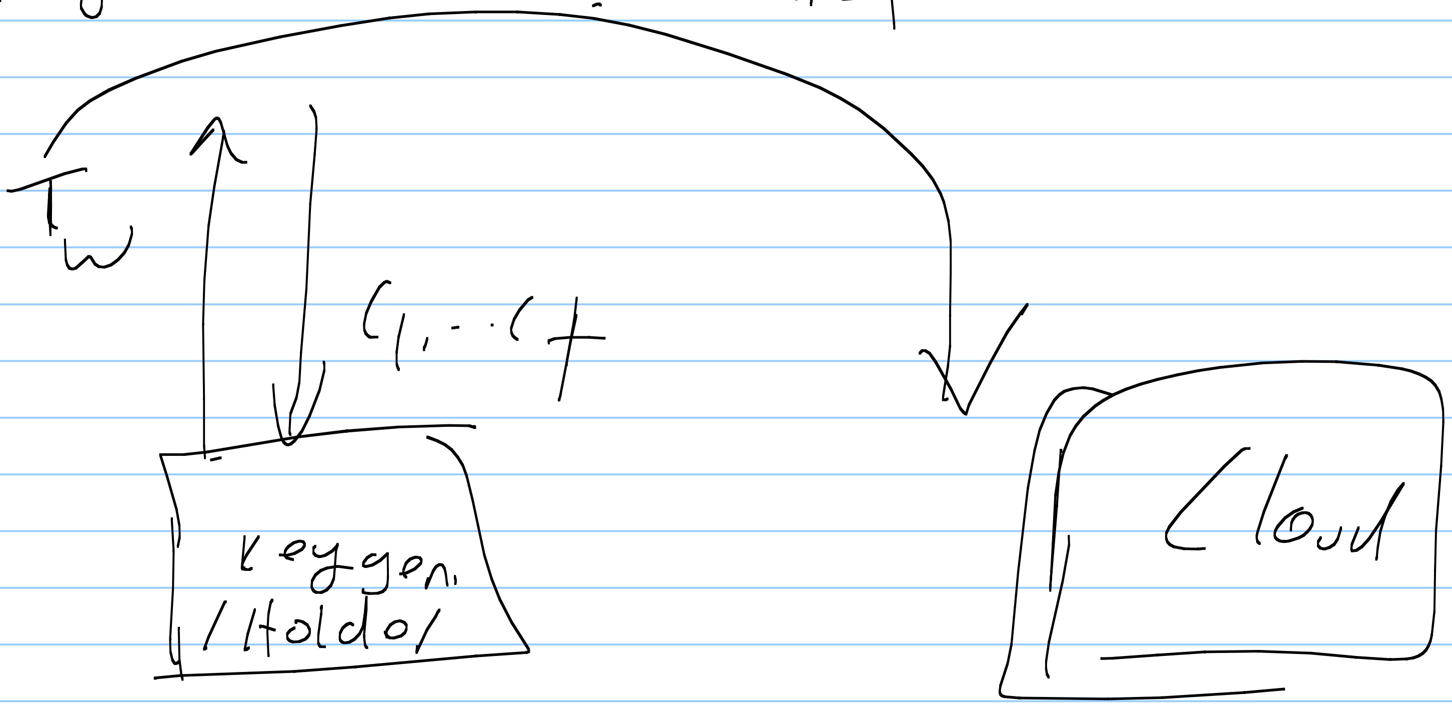
Consider audit logs outsourced to cloud.
Assume audit logs are encrypted for security.
You may want to delegate the auditing of certain logs such as

('the user name is Bob') OR ((the date is between X & Y')

AND

'Data is accessed from Japan'

Fine-grain access control



Definition (Access structure)

Let $\{P_1, \ldots P_n\}$ be a set of parties

A collection $\mathcal{A} \subseteq 2^{\{P_1, \ldots P_n\}}$ is monotone:

if $\forall B, C$: if $B \in \mathcal{A}$, $B \subseteq C$ then $C \in \mathcal{A}$

An monotone access structure is a monotone collection $\mathcal{A}$ of non-empty subsets of $\{P_1, \ldots P_n\}$

The sets in $\mathbb{A}$ are called the authorized sets and the sets not in $\mathbb{A}$ are called unauthorized sets.

- In our context, we will define access structure over attributes

An (Key-Policy) Attribute Based Encryption consists of four algorithms

Setup: $\rightarrow PK, MK$

Encryption: $(m, \underbrace{\gamma}_{\text{set of attributes}}, PK) \rightarrow E$ (ciphertext)

Key Generation: $(\underbrace{A}_{\text{access structure}}, MK, PK) \rightarrow D$

Decryption $(E, \gamma, D) \to M$

ciphertext

for access structure A

Message

output M if $\gamma \in A$

Selective set model for ABE

Init: Adversary (Adv) declares γ that he wishes to be challanged

Setup: Run ABE and give the PK params to Adv

Phase1: Adv issues queries for many access structures A_j where $\gamma \notin A_j$ for all j

challenge: Adv submits M_0, M_1 challenger encrypts $C \leftarrow E(M_b)$ with γ for randomly chosen bit b

C is given to Adv.

Phase 2: Phase 1 is repeated

Guess: The adversary outputs a guess b' of b.

The advantage of Adv is defined as

$$\left| \Pr[\, b' = b \,] - \frac{1}{2} \right|$$

For secure ABE, advantage of any poly-time adv. is small negligibility

A ABE scheme discussed uses a bilinear map defined over G_1, G_2 where both G_1, G_2 are multiplicative, cyclic groups of size p.

~~A~~ The ABE scheme is also based on D-BDH assumption

(Note that D-BDH assumption implies for $\forall a, b, c, z \in \mathbb{Z}_p$ chosen randomly for $\forall$ poly-time adversary, the following two tuples are indistinguish,

① $\left(A = g^a, \ B = g^b, \ C = g^c, \ e(g,g)^{abc} \right)$

$\quad\ \ \, \| \qquad\quad \| \qquad\quad \|$

② $\left(A = g^a, \ B = g^b, \ C = g^c, \ e(g,g)^{z} \right)$)

DH over general g:

① $A = g^a$, $B = g^b$, g^{ab}

② $A = g^a$, $B = g^b$, g^c

ABE construction for Access Tree

Access Tree T: be a tree representing an access structure each non-leaf node X is a threshold gate

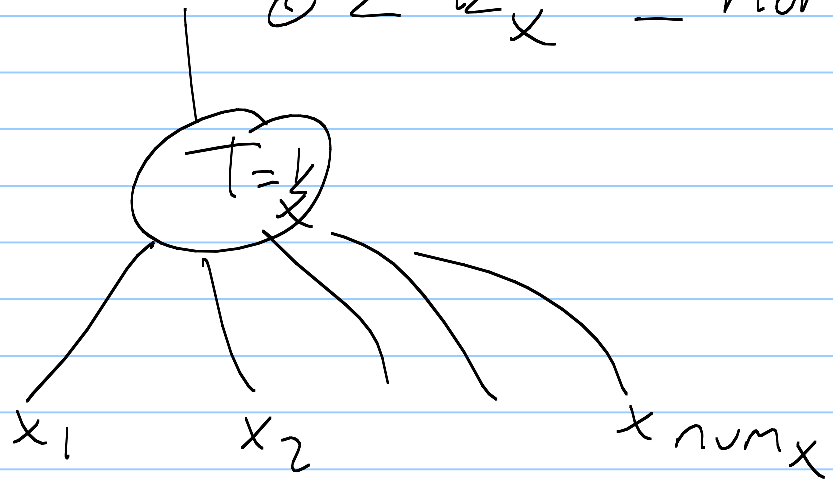
$Y = 1$ if $\sum w_j \cdot x_j \geq T$

$Y = 0$ if $\sum w_j \cdot x_j < T$

In this paper, w_j are $\mathbb{1}$, x_j's are binary.

(num_x is the number of children of x

k_x is the threshold value where $0 < k_x \le num_x$



If $k_x = 1$ then it becomes OR gate

If $k_x = num_x$ then it becomes AND gate

Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$

att(x) denotes the attribute associated with leaf node x

parent (x) is the parent of node x

index(x) returns the index of a node.

we say γ satisfies T with root node r (i.e., $T_r(\gamma) = 1$)

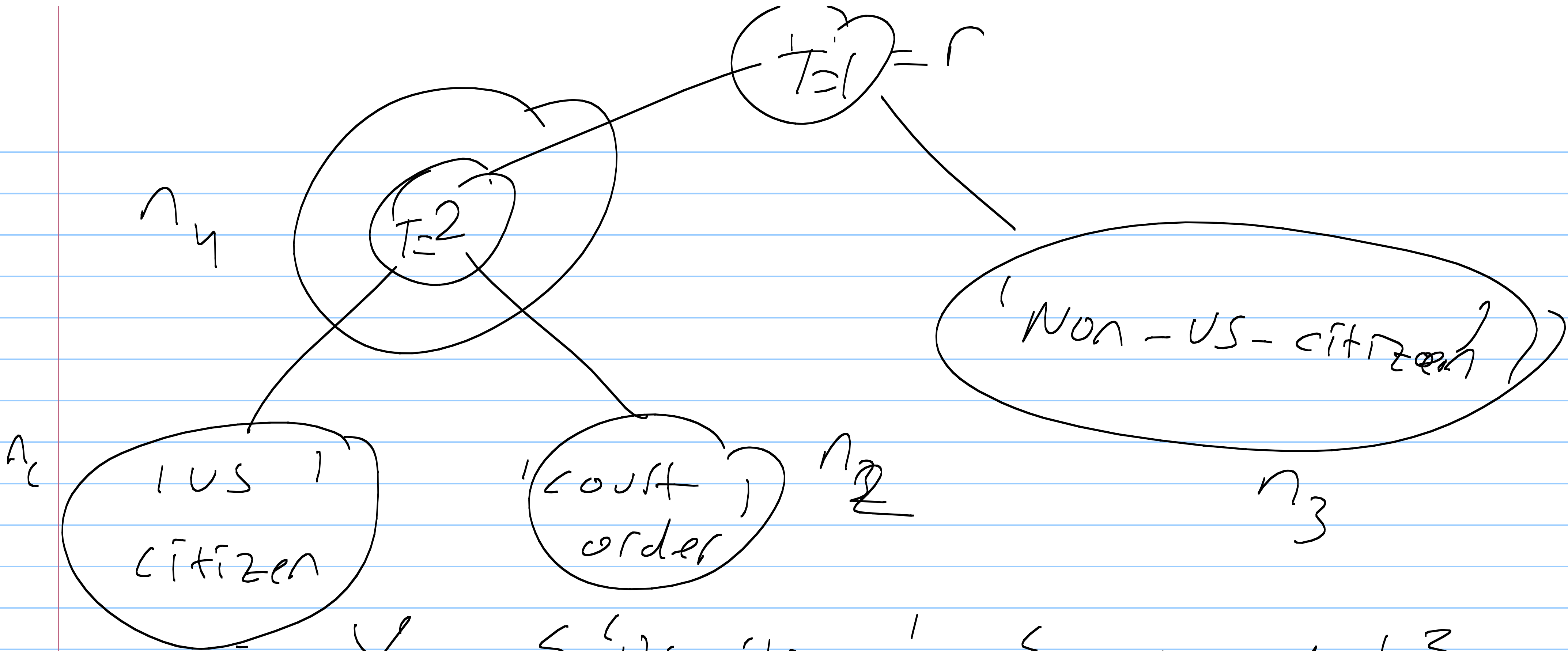
if x is not a leaf node

$$T_x(\gamma) = \left(\sum_{x = \text{parent}(x')} T_{x'}(\gamma) \overset{?}{\geq} k_x \right)$$

than return 1 else return 0

if x is a leaf node

$T_x(\gamma)$ return 1 iff $att(x) \in \gamma$



$$\gamma = \{\text{'US citizen'}, \text{'court order'}\}$$

$$T_r(\gamma) = T_4(\gamma) + T_3(\gamma) \overset{?}{\geq} 1$$

$T_4(8) = T_1(8) + T_2(8) \overset{?}{\geq} 2$

$T_4(8) = 1 + 1 \overset{?}{\geq} 2 = 1$

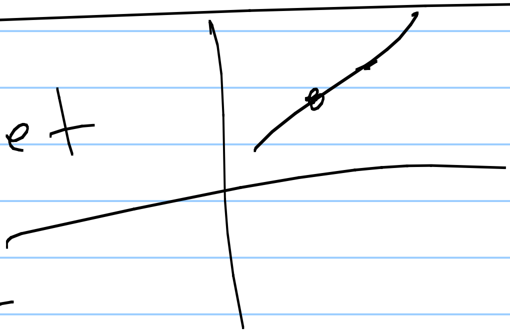
$T_3(8) = 0$

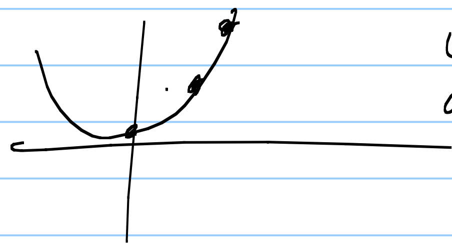
$T_r(8) = 1$

$1,\ P(1) \rightarrow u_1$

$n,\ P(n) \rightarrow u_n$

$\binom{n}{t}$ secret sharing





$y = ax^1 + b$

$y = ax^2 + bx + c$

$P(0) = S$

$\deg(P) = \cancel{k} - 1$

$\forall i \in \mathbb{Z}_p$ and set S define

$$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x - j}{i - j}$$ (Lagrange coefficients)

---

setup: Define the univ. of attribute

$$U = \{1, \ldots, n\}$$

$\forall i \in U \quad t_i \xleftarrow{R} \mathbb{Z}_p$

$y \xleftarrow{R} \mathbb{Z}_p$

PK params are

$$T_1 = g^{t_1} \quad \ldots \quad T_{|U|} = g^{t_{|U|}}$$

$$Y = e(g, g)^y$$

MK is

$$t_1, \ldots t_{U}, y$$

Encrypt(M, γ, PK)

To encrypt $M \in G_2$ using $\gamma \subseteq U$

choose $s \xleftarrow{R} \mathbb{Z}_p$

$E = (\gamma,\ M \cdot Y^{s},\ \{E_i = T_i^{s}\}_{i \in \gamma})$

γ is disclosed.

$Y^{s} = e(g,g)^{y \cdot s}$

Key Generation (T, MK)

Goal: output key that enables decryption of a message associated with γ if $T(\gamma) = 1$

For each node x in the tree set $d_x = k_x - 1$ where d_x is the degree of polynomial q_x associated with node x.

root node set $q_r(0) = y$ and d_r other points of q_r to define it completely

for any other node x,

$$q_x(0) = q_{parent(x)}(index(x))$$

and choose d_x points to define q_x, for each leaf node

give secret value to use

$$D_x = g^{\frac{q_x(0)}{t_i}} \quad \text{where } i = att(x)$$

DecryptNode (E, D_T, x) private key

$\{$ $(\gamma, E', \{E_i\}_{i \in \gamma})$

It will output $F_x \in G_2$ or $\perp$

if x is a leaf where $i = \mathrm{att}(x)$

DecryptNode$(E, D, x) =$

$\{$ return $e(D_x, E_i) = e\left(g^{\frac{q_x(0)}{t_i}}, g^{t_i \cdot s}\right)$

$= e(g, g)^{q_x(0) \cdot s}$ if $i \in \gamma$

return $\perp$ if $i \notin \gamma$

$\}$

if x is non-leaf node

for each child of node(x)

$F_{z_i} =$ DecryptNode(E, D, z_i);

Let S_x be an arbitrary k_x sized set of f_{z_i} where $f_{z_i} \neq \perp$

(if no such S_x, return $\perp$)

else

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S_x'}(0)}$$

where $i = \text{index}(z)$

$$S_x' = \{\text{index}(z) : z \in S_x\}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{s \cdot q_z(0)}\right)^{\Delta_{i,S_x'}(0)}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{s \cdot q_{\underbrace{\text{parent}(z)}_{= x}}(\text{index}(z))} \right)^{\Delta_{i, S_x'}(0)}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{s \cdot q_x(i) \Delta_{i, S_x'}(0)} \right)$$

$$= e(g,g)^{s \cdot \sum q_x(i) \Delta_{i, S_x'}(0)}$$

polynomial interpolation

$$= e(g,g)^{s \cdot q_x(0)}$$

Call decrypt node (E, D_T, r) returns

Y^s if $T(\gamma) = 1$

and compute $E' \cdot (Y^s)^{-1}$ to get M.

Efficiency of Encryption

$O(\gamma)$ many G_2 exponent comp.

PK size $O(\gamma)$

Decryption (worst case for basic model is $O(|T|)$)

If you are careful, you can reduce it to $|S_r|$ many exponentiations

large universe for attributes are possible. As we discussed, attributes are disclosed.

PREDICATE ENCRYPTION

Goal: PK scheme where secret keys correspond to predicates f in some class F and a sender associates a ciphertext with an attribute I in a set Σ

$C(I)$ can be decrypted by a secret key sk_f corresponding to predicate $f \in F$

iff $f(I) = 1$

Goal: Attribute hiding

It requires that a ciphertext hides all info. about I except what is leaked by known private key.

In this & next class, goal is to construct an attribute-hiding scheme for specific Σ, F

$$\Sigma = \mathbb{Z}_N^n,$$
$$(v_1, \ldots, v_n)$$
$$v_i \in \mathbb{Z}_N$$

$$F = \{ f_x \mid x \in \mathbb{Z}_n^n \text{ where } f_x(y) = 1 \text{ iff } \langle x, y \rangle = 0 \bmod N$$
$$= \sum_{i=1}^{n} x_i \cdot y_i = 0 \bmod N \}$$

- uses two other hard problem assumptions (please see the paper for details)

Important applications of any predicate enc. supporting "inner product" queries

Given a vector $x \in \mathbb{Z}_N^\ell$

$$f_x : \mathbb{Z}_N^\ell \to \{0,1\}$$

$$f_x(y) = \begin{cases} 1 & \text{iff } \langle x, y \rangle = 0 \\ 0 & \text{else} \end{cases}$$

Anonymous IBE could be generated easily

IBE_Setup(1^n)

{

Run setup_Predicate(1^n) and get PK, SK

↳ master key

}

IBE_Private-Key(ID)

{

run GenKey$(f_{I'}, SK)$

where $I' = (1, ID)$

$s_{ID} \leftarrow SK_{f_{I'}}$

}

IBE-ENC(PK, ID, M)

{

set $I' = (-ID, 1)$

run $Enc_{PK}(I', M)$

}

IBE_DEC (s_{ID}, ID, c)

{

Note $s_{ID} = sk_{(-ID,1)}$

so run $DEC(sk_{f_I'}, c) = m$

if $f_{I'}(I) = 1 \iff \langle (-ID, 1), (1, ID) \rangle = 0$

}

Predicate Encryption scheme for the class of predicates F over the set of attributes Σ consists of four

P. P. T algs.

setup, GenKey, Enc, Dec

$\mathrm{setup}(1^n) \to (PK, SK)$

$\mathrm{GenKey}(f, SK) \to SK_f$

$\mathrm{Enc}(PK, I, M) \to C$

$\mathrm{Dec}(sk_f, c)\ \{$ outputs $\perp$ if $f(I) = 0$

else /* $f(I) = 1$ */

$\mathrm{Dec}_{sk_f}(\mathrm{Enc}_{pk}(I, m)) = m$

$\}$

Definition of Attribute Hiding:

A predicate enc. scheme w.r.t. F and Σ is attribute hiding if for all P.P.T adversary A, the advantage of A in the following exp. is negligible ((success Prob. $\leq \frac{1}{2}$ (n) → sec. param) in the security parameter n.

1) $A(1^n)$ outputs $I_0, I_1 \in \Sigma$

2) challenger runs setup to get PK, SK. Adversary is given PK.

3) A can get any key S_{f_i} s.t. $\forall i \in [1, \dots, l]$, $f_i(I_0) = f_i(I_1)$

4) A outputs equal length M_0, M_1
(if $\exists i$ s.t $f_i(I_0) \neq f_i(I_1)$ the
it is required $M_0 = M_1$)

5) A is given $Enc_{pk}(I_b, M_b)$

6) step 3 can be repeated

7) A outputs a bit b' and
succeeds if $b' = b$

The advantage of A should be negligible.

$$\left| \Pr(A \text{ is sucessfull}) - \frac{1}{2} \right| < \frac{1}{2^{n}}$$

Hidden Vector Encryption (HUE)

Given a set Σ, $\Sigma_{*} = \Sigma \cup \{*\}$

HUE is a predicate enc. schem. for the class of predicate

$$\Phi^{hve}_{\ell}$$

where

$$\Phi^{hve}_{(a_1, \dots, a_\ell)}(x_1, \dots, x_\ell) = \begin{cases} 1 & \text{if } \forall_i \; a_i = x_i \text{ or } a_i = * \\ 0 & \text{otherwise} \end{cases}$$

we can use the dot product based predicate enc. scheme to implement HUE.,

- setup is the same
- for predicate $\Phi^{hve}_{(a_1, \ldots a_\ell)}$

create vector $A = (A_1, \ldots A_{2\ell})$

if $a_i \neq *$, $A_{2i-1} = 1$, $A_{2i} = a_i$

if $a_i = *$, $A_{2i-1} = 0$, $A_{2i} = 0$

- to get the private key

run $\text{GenKey}_{SK}(f_A)$

- to encrypt a message μ for $(x_1, \ldots, x_\ell)$

Choose random $r_1 \dots r_l$

$$\tilde{X}_r = (\tilde{x}_1, \dots, \tilde{x}_{2l})$$

where $\tilde{x}_{2i-1} = -r_i x_i$, $\tilde{x}_{2i} = r_i$

$x_1 = 1$, $x_2 = 3$

$a_1 = 3$, $a_2 = 1$

Note that

$$\langle A \cdot \tilde{X}_r \rangle = \sum_{\substack{i=1 \\ a_i \neq A}}^{l} \Big((\underbrace{-r_i x_i}_{\tilde{x}_{2i-1}} \cdot \underbrace{1}_{A_{2i-1}}) + \underbrace{r_i}_{\tilde{x}_{2i}} \underbrace{a_i}_{A_{2i}} \Big)$$

$\Phi^{hve}(x_1, \dots x_\ell) = 1 \Rightarrow \langle A, x_r \rangle = 0$ for $\forall r$

$a_1, \dots a_\ell$

$\Rightarrow f_A(x_r) = 1$ for $\forall r$

Assume $x_1 = 1$, $x_2 = 3$

$a_1 = 3$, $a_2 = 1$

generate A and $\tilde{x}_r$ for $r \in (1, 1)$

what is $\langle A, \tilde{x}_r \rangle = 0 \Rightarrow f_A(\tilde{x}_r) = 1$

$A_1 = 1 \quad A_2 = 3 \quad \tilde{x}_1 = -1 \quad \tilde{x}_2 = 1$

$A_3 = 1 \quad A_4 = 1 \quad \tilde{x}_3 = -3 \quad \tilde{x}_4 = 1$

$$\sum_{i=1}^{4} A_i \cdot \tilde{x}_i = 1 \cdot -1 + 3 \cdot 1 + 1 \cdot -3 + 1 \cdot 1 = 0$$

$$\Phi^{hue}_{(a_1, a_2)} (x_1, x_2) = 0$$

If $\gcd(a_i - x_i, N) = 1$ for all i then

$$\Pr[f_A(x_r) = 1] = \Pr[\langle A, x_r \rangle = 0] = \frac{1}{N}$$

If $\underset{(a_1, \ldots a_\ell)}{\Phi}(x_1, \ldots, x_\ell) = 0$