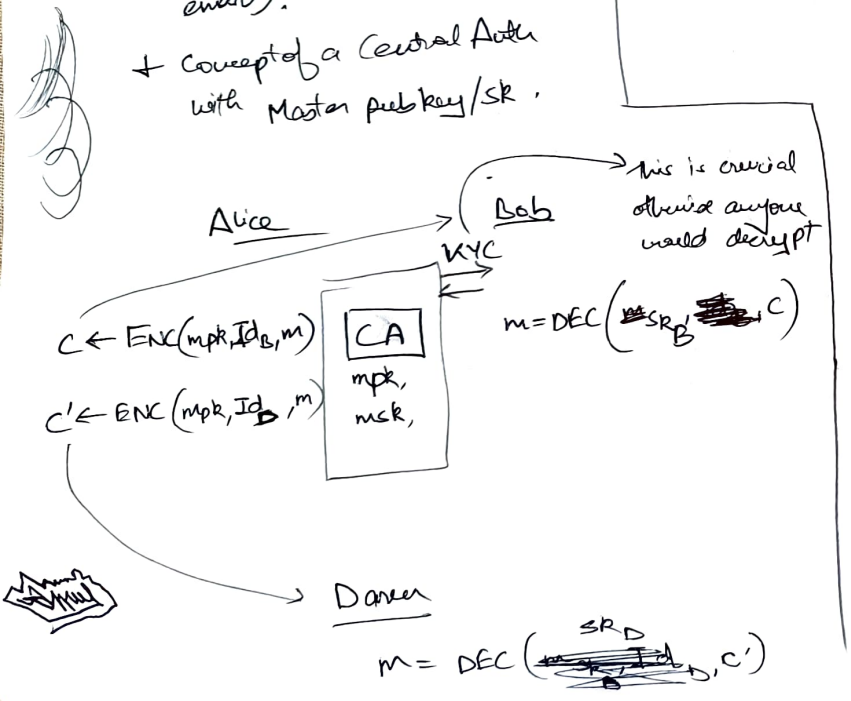
13/02/2023

Identity BASED ENCRYPTION

+ DHKEX requires both to be online,
+ IF offline [only 1 party] then that party publish public key on internet.
+ A doesn't find PK_B, but knows B identity (for eg. email).
+ Concept of a Central Auth with Master pub key/SK.



IBE algo

1) $KGEN(\lambda) \rightarrow (mpk, msk)$
2) $ENC(mpk, Id, m) \rightarrow C$
3) $KExt(msk, Id) \rightarrow sk_{ID}$
4) $DEC(sk_{Id}, C) \rightarrow m$

Secret key is generated separately to each Id

CH_{IBE} | A_{IBE}

$KGEN \rightarrow mpk, msk \longrightarrow mpk$

$KExt(msk, id)$ ← id
↓ skid → skid

$id \leftarrow ID$
$b \xleftarrow{R} \{0,1\}$ ← id^*, m_0, m_1
$c^* \leftarrow ENC(mpk, id^*, m_b)$
c^* →
← b'

Win if $(b = b')$ & $(id^* \notin ID)$

CPA secure.

Use Fujisaki okamoto transformation → CCA secure

BONEH-FRANKLIN '01 (Pairing based IBE)

$\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$

$\langle g_1 \rangle = \mathbb{G}_1$

$\langle g_2 \rangle = \mathbb{G}_2$

$\langle g_T \rangle = \mathbb{G}_T$

order $= p$

$H: \{0,1\}^* \to g \in \mathbb{G}_1$

SETTING

KGEN $\to (msk, mpk)$

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

$msk := s$

$mpk := g_2^s$

KEXT(msk, id)

$sk_{id} = H(id)^s$

$s = msk$

ENC$(mpk, M \in \mathbb{G}_T, id) \to C_1 C_2$

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

$C_1 := g_1^r$

$C_2 := e(H(id), mpk)^r \cdot M$

$= e(H(id)^s, g_2^r) \cdot M$

DEC$(SK, C_1 C_2) \to M$

$M := \dfrac{C_2}{e(sk_{id}, C_1)}$

Lemma 1:

Basic PUB is CPA-secure $\Rightarrow$ BF'01 is CPA secure IBE

[Goal is simulate KExtraction

g_1, g_2 are known.

Ch_{PKE} | $REDU_{PK}$ | A_{IBE}

$mpk \to pk = (g_2^s, h) \to mpk$

$mpk = g_2^s$

$\downarrow \psi$ (isomorphism)

$g_1^s = \alpha$

$id^* \neq id$

$r \xleftarrow{R} \mathbb{Z}_p$ — id KEx

Random Oracle $\leftarrow H(id) = g_1^r$

$sk_{id} = H(id)^s$

$= (g_1^s)^r$

$= (\alpha)^r$

$H(id^*) = h$

m_0, m_1

m_0, m_1

$g_2^r, e(H(id^*)^r, g_2^s) \cdot M$

PROOF : THM: BF '01 is a CPA secure IBE

if BDDH holds & $\psi: \mathbb{G}_2 \to \mathbb{G}_1$ Isomorphism

BASICPUB (PKE)

KGEN $\to (sk, pk)$

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

$h \xleftarrow{R} \mathbb{G}_1$

$sk := h^s$

$pk := (g_2^s, h)$

ENC$(pk, M) \to (C_1 C_2)$

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

$C_1 := g_2^r$

$C_2 := e(h^r, pk) \cdot M$

$pk \Rightarrow (pk_1, pk_2)$

$= g_2^s, h$

DEC$(sk, C_1 C_2)$

$M := \dfrac{C_2}{e(sk, g_2^r)}$

$= e(sk, C_1)$

Lemma 2: Basic BUB is CPA-secure assuming BDDH (~~if~~ & Ψ (isomorphism)

PROOF:

$\forall m_0, m_1 \quad PK, ENC_{PK}(m_0) \approx PK, ENC_{PK}(m_1)$

$h = g_1^{a} \cdot g_2^{s} \cdot g_2^{r} \cdot e(g_1, g_2)^{ars} \cdot M_0$

$\approx$

$g_1^{a} \; g_2^{s} \; g_2^{r} \; U_T \cdot M_0$

$U_T \xleftarrow{R} G_T$

IBE — Signature (Black Box)

Naor ~~Reingold~~

Transform

SIGN KGEN $\rightarrow$ (sk, vk)

KGEN $(1^\lambda) \rightarrow$ mpk, msk

vk := mpk

sk := msk

SIGN (sk, M) (msk)

id = H(M)

skid = Keat(msk, id)

σ = skid

VERIFY (M, σ, vk) uses randomness (not typical)

$M \xrightarrow{H} id = H(M)$

$\sigma := skid$

vk := mpk

$r \xleftarrow{R} M$

$C \leftarrow ENC(mpk, id, r)$

$m' \leftarrow DEC(\underbrace{skid}_{=\sigma}, C)$

if $(m' = m)$ then RET 1

else RET 0

$mpk = g_2^{s} \; sk_2 \cdot H(id)^{s}$

s is common

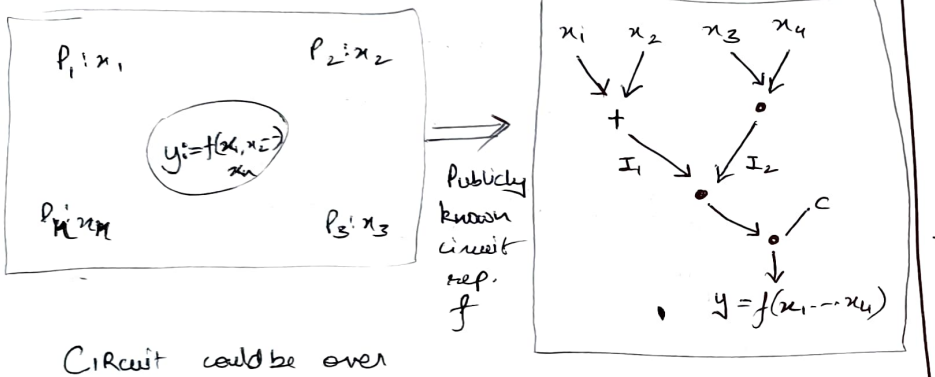
so similarly we need

another common here

GMW Approach – Shared Circuit EVALUATION

SS

– Similar to BGW



Circuit could be over $(\mathbb{R}, +, \cdot)$ or $(\mathbb{F}, +, \cdot)$

+ Inputs: Randomly (n,t) secret shared
+ GMW invariant:
  – If gate I/Ps are randomly (n,t) secret shared, then gate-output is randomly (n,t) secret shared
+ O/P – Publicly reconstruct the secret shared output

Which secret sharing?
- Shamir's imp bcoz Shamir is applicable over Rings
- Also Shamir is $[n, (n-1)]$ secret sharing

– So ADDITIVE SECRET SHARING SCHEME

Additive SSS (Perfectly secure)

+ Threshold $t < n$ — in worst case $t = n-1$
+ Intuition: secret 's' is divided into n random shares
  – $\sum_{n}$ shares $= s$
  – $\forall s' \subset_{(n-1)} s$ s' is independent of s.
+ $Sh_{Add}(s)$
  – Select $s_1, \ldots, s_{n-1} \in_R \mathbb{R}$
  – $s_n := s - (s_1 + \cdots + s_{n-1})$
  – O/P $s_1, \ldots, s_n$
+ $Rec_{Add}(s_1, \ldots, s_n)$
  – O/P $\sum_{i=1}^{n} s_i = s$

s	s_1	s_2	s_3	s_4
	•	•	•	•
s'	s_1'	s_2'	s_3'	s_4'
ss'	u_1	u_2	u_3	u_4

Non linear so it is not a (n,t) ss of (s, s') ✗

linearity prop of Add SSS

s	s_1	s_2	s_3	s_n
	+	+	+	+
c	c	0	0	0
	u_1	u_2	u_3	u_n

(n,t)-sharing of $(c+s)$ ✓ Holds

PROP 1

Dealer(s) $Sh_{Add}(s) \to (s_1, s_2, s_3, s_4)$

s_1 s_2 s_3 s_4

s	s_1	s_2	s_3	s_4
	+	+	+	+
s'	s_1'	s_2'	s_3'	s_4'
	u_1	u_2	u_3	u_4

(n,t) ss of $(s+s')$ ✓ Holds.

PROP 2

s	s_1	s_2	s_3	s_4
	•	•	•	•
c	c	c	c	c
	cs_1	cs_2	cs_3	cs_4

Multiply

(n,t) ss of (cs) ✓

PROP 3

GMW in PREPROCESSING MODEL

SHARED CIRCUIT EVAL

+ Preproc: Random, private additively shared multiplication triples.
+ Circuit evaluation: Beaver's method $(n=4, t=3)$

$f(x_1\ x_2\ x_3\ x_4) := (x_1+x_2)+(x_3\ x_4)$

$a_1\ a_2\ a_3\ a_4$ $\quad \sum a_i = a$]
b_1 —— b_4 $\quad \sum b_i = b$] Additively secret shared
c_1 —— c_4 $\quad \sum c_i = c$]

(a,b,c) random multiplication triples.

PRE PROC. [a, b are randomly chosen]

$d := x_3 - a$]
$e := x_4 - b$] OTP encryption of x_3, x_4

$x_3 x_4 = (d+a)(e+b) = x_3 x_4$

$= de + db + ae + ab$

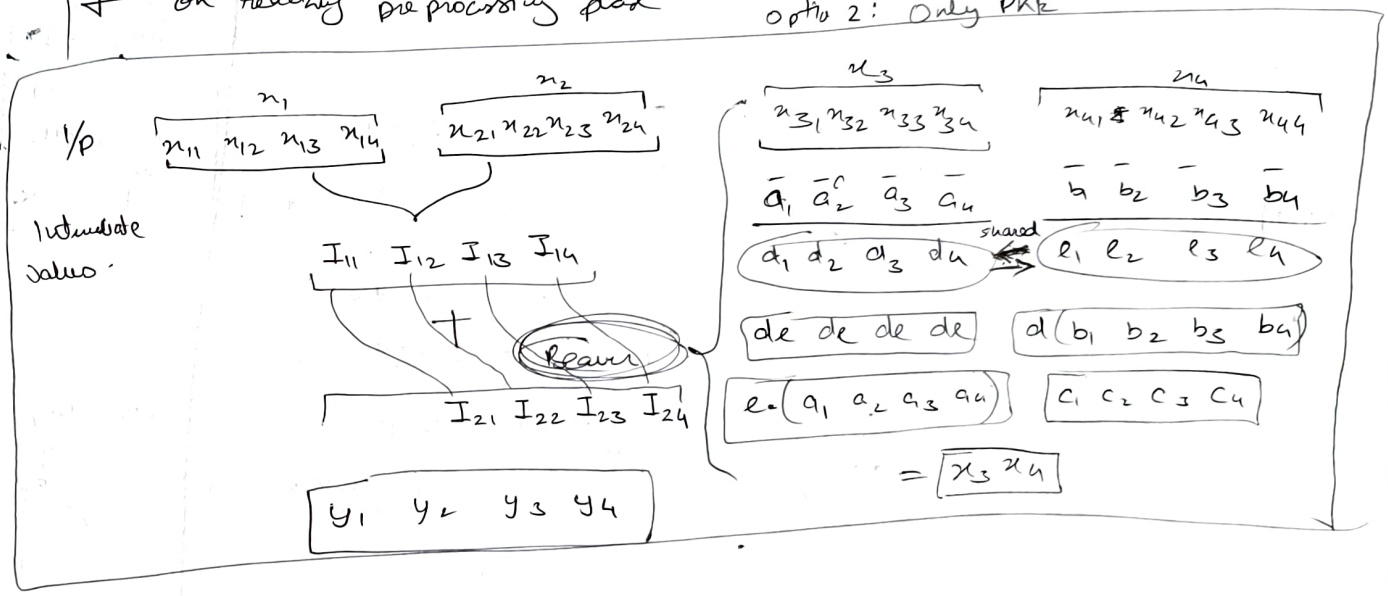
linear function of (a,b,c)

Analysis

+ Perfectly secure, conditioned on secure realization of pre-processing phase
+ $O(D_{\text{multiplication}})$ rounds. [Mult, only place where req interaction
+ Total communication complexity: $O(C_{\text{Mult}}\, n)$ ring elements

C_{Mult} = # Multiplication gates.

Since $t<n$, need to deploy cryptography
on realizing pre processing phase — option 1: PKE + Sym key primitive
option 2: Only PKE



OBLIVIOUS TRANSFER

WHY OT

- Preprocessing phase of GMW, it is used.

2 props
- CORRECTNESS
- PRIVACY

1-out of 2 OT

Sender: m_0, m_1 — Receiver: $b \in \{0,1\}$

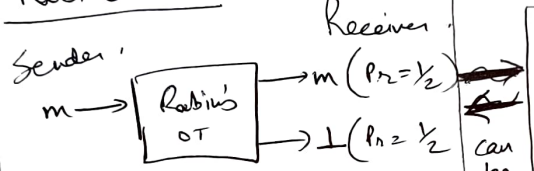
Sender's info (sender security)
- corrupt R should learn no additional info about m_{1-b}
- Simulation ~~Receiver~~ (not done)

Receiver's security
- Corrupt sender should not learn about b

k out of N OT

Variants

Rabin's OT



can be transformed

Sender DNK whether m was transferred or not.

1 out of 2 OT

⇒ 1 out of N OT: N messages, $b \in \{0 \ldots N-1\}$, {m_b} Receiver

⇒ k out of N OT: N messages, $i_1 \ldots i_k \in \{0 \ldots N-1\}$, $m_{i_1} \ldots m_{i_k}$

Constructing OT based on

- Factoring
- DDH
- Trapdoor one way (RSA)
- ~~One way~~ Noisy channel

Can we design OT protocols unconditionally (without any assumptions)?

UNC imp. secure OT → Unconditionally secure 2-party AND protocol

A → B, But ~~A~~ B ~~is impossible~~ is impossible.

∴ NO.

UNC means computationally unbounded

Secure OT → Secure 2-party AND protocol

$m_b = (1 \oplus b) m_0 \oplus b m_1$

AND

$x_1 x_2$	
00	0
01	$0 = x_1 x_2$
10	0
11	$1 = x_1 x_2$

CORRECTNESS

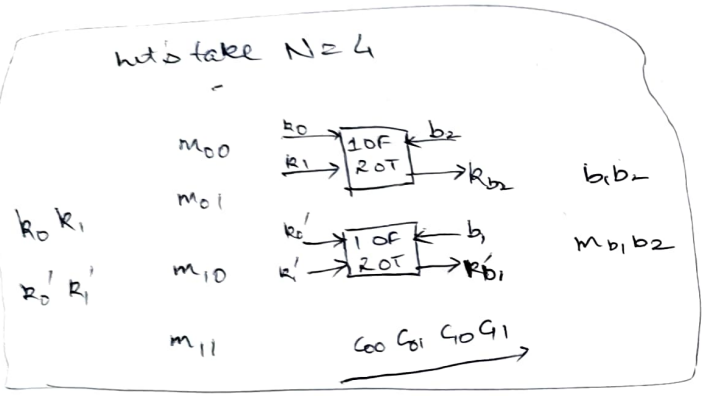
PRIVACY
- Bob corrupt, still do not know x_1
- Alice corrupt, still does not know b

FROM 1 OF 2 OT

TO

1 OF N OT

$N = 2^k$



Process of encryption:

IF scheme is:

$$\begin{aligned} c_{00} &:= k_0' \oplus k_0 \oplus m_{00} \\ c_{01} &:= k_0' \oplus k_1 \oplus m_{01} \\ c_{10} &:= k_1' \oplus k_0 \oplus m_{10} \\ c_{11} &:= k_1' \oplus k_1 \oplus m_{11} \end{aligned}$$

then $c_{00} \oplus c_{01} \oplus c_{10} \oplus c_{11} = m_{00} \oplus \cdots m_{11}$

But this widgets privacy. as Bob learns additional info.

✗

do We use a keyed PRF (AES)

$$\begin{aligned} c_{00} &:= [F_{k_0'}(00)] \oplus [F_{k_0}(00)] \oplus m_{00} \\ c_{01} &:= [F_{k_0'}(01)] \oplus [F_{k_1}(01)] \oplus m_{01} \\ c_{10} &= [F_{k_1'}(10)] \oplus [F_{k_0}(10)] \oplus m_{10} \\ c_{11} &= [F_{k_1'}(11)] \oplus [F_{k_1}(11)] \oplus m_{11} \end{aligned}$$

$c_{00} \oplus \cdots c_{11} \neq$

✓

Another method (Double encryption)

$c_{ij} = ENC_{k_i'}(ENC_{k_j}(m_{00}))$

✓

Key homomorphism $\Rightarrow$ Threshold

sk NPR PRF

$y = H(x)^{sk}$

Key homomorphism:

Ex1: $sk = sk_1 + sk_2$

$y_1 = H(x)^{sk_1}$

$y_2 = H(x)^{sk_2}$

$y = y_1 y_2 = H(x)^{sk_1 + sk_2}$

Ex2: $sk \xrightarrow{\text{SHAMIR}} sk_1 \cdots sk_n$

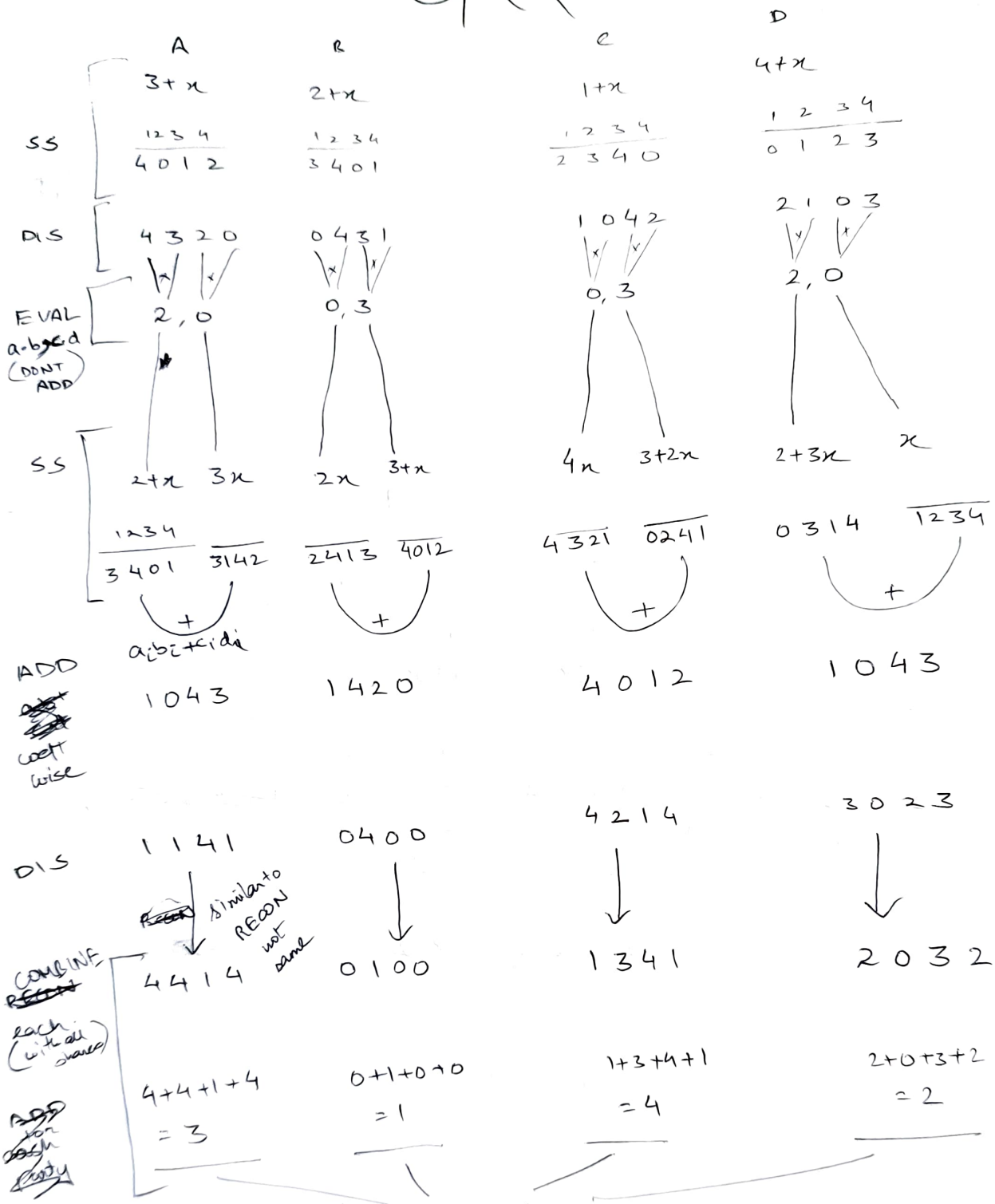
$y_1 := H(x)^{sk_1}$ ~~...~~ $sk = p(0)$

~~...~~

$y = \prod_{i \in I} y_i^{L_{i,I}(0)}$

$p \xleftarrow{R} \mathbb{Z}_p^t[x]$

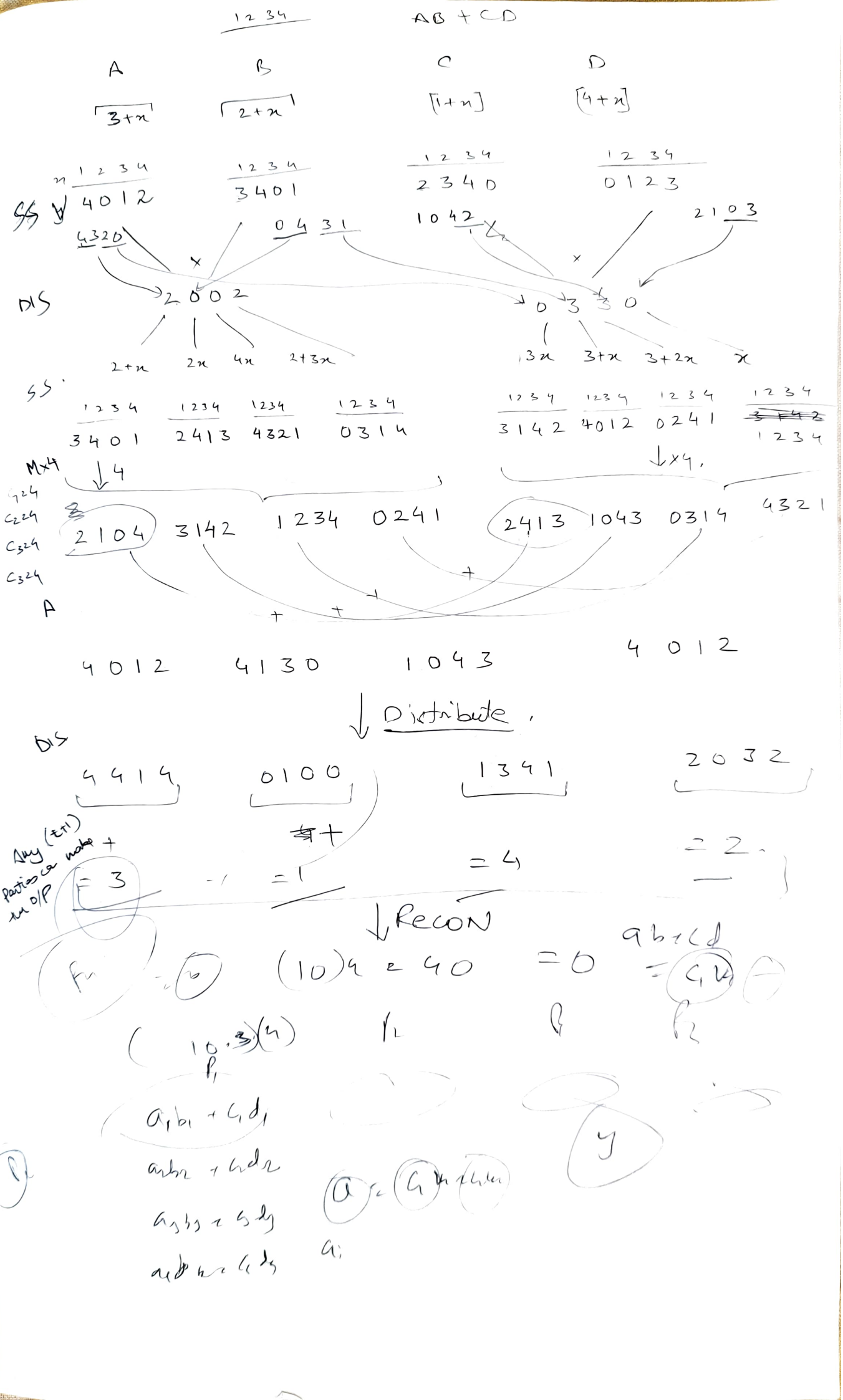
A.B + C.D GRR

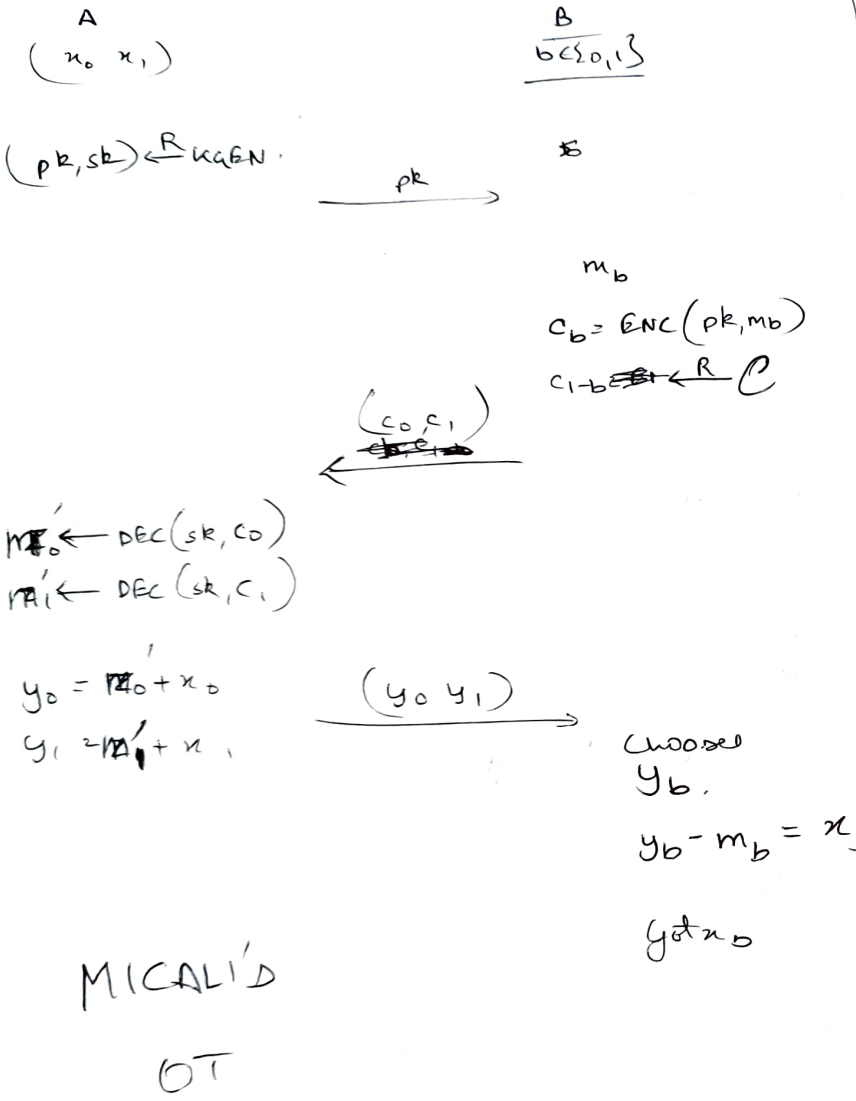


Any 2 party can come together to form AB+CD

$$\frac{3(t+2)}{t+1} + 1\left(\frac{t+1}{1}\right)$$

$$1 - 1 = 0$$


1234
AB + CD
A
B
C
D
3+n
2+n
1+n
4+n
1234
4012
1234
3401
1234
2340
1234
0123
4320
0431
1042
2103
DS
2002
0330
2+n
2n
4n
2+3n
3n
3+n
3+2n
n
SS
1234
3401
1234
2413
1234
4321
1234
0314
1234
3142
1234
4012
1234
0241
1234
1234
M×4
↓4
↓×4.
2104
3142
1234
0241
2413
1043
0314
4321
A
4012
4130
1043
4012
↓ Distribute.
DS
4414
0100
1341
2032
Any (±1) parities ca make + ve O/P
= 3
= 1
= 4
= 2.
↓ RECON
(10)4 = 40
= 0
ab+cd
= c14
y



MICALI'S OT

+ B is corrupt, so create SIM in ideal W.

$SIM(b, x_b)$

↓

has I/P b of corrupt B & desired x_b ahead of A

+ SIM, TTP interaction

SIM ⟶

A ↔ B ⟶ Transcript

SIM - B ⟶ Transcript

A(3) . B(2) C

SS1 [3+x, 2+x, 1; 4 0 1, 3 4 0, 2]

DIS $a_1 b_1$ 4,3 | $a_2 b_2$ 0 4 | $a_3 b_3$ 1 0

EVAL 2 ($a_1 b_1$) | 0 ($a_2 b_2$) | 0 ($a_3 b_3$)

SS 2+2x | 4x | 3x

4,1,3 | 4,3,2 | 3,1,4

AB_{11} AB_{12} AB_{13}

DIS 4,4,3 | 1,3,1 | 3,2,4

AB_{12} AB_{22} AB_{32}

Ck1 (each different) Σc.k. 2+3+3 = 3 | 3+1+1 = 0 | 4+4+4 = 2

$\frac{(-2)(-3)}{(-1)(-2)} = 3$

$\frac{(-1)(-3)}{(-1)} = 2$

$\frac{(+1)(-2)}{2} = 1$

B

AB (GRR) (1) . C (1+x) 2. (1+x) . A .

SS1 1+2x | NO need | 2 3 4 | 1

DIS 3,2 | 0,3 | 2,4

EVAL 1 | | 3

SS 1+2x | x | 3+4x

3,0,2 | 1,2,3 | 2,1,0

DIS (3,1,2) | (0,2,1) | 2,3,0

Ck1 3 | 0 | 2

$\frac{3+2}{(+1)} = 1.$

$\frac{2(-2)}{1} = -4 = 1$

$3\left(\frac{+3}{+2}\right)3 - 2\left(\frac{-1}{2}\right) = 1$

$-1\left(\frac{-2}{1}\right) + 2\left(\frac{+3}{+1}\right) = 2 + 1 = 3$

$\frac{3(+3)}{+2} + 2\left(\frac{-1}{2}\right)$ $2 - 1 = 1$

$\frac{2(-2)}{1} = -4$ 2. $3(2) + 0 = 2.$

$3+1\left(\frac{-1}{1}\right)$ R. $4\left(\frac{+2}{+1}\right) + 1\left(\frac{-1}{1}\right)$ $3 - 1 = 2$

BELLARE-MICALI OT

G = group, order p, $\langle g \rangle$

$H: G \rightarrow \{0,1\}^l$ (random oracle)

$S(m_0, m_1)$ | $R(b \in \{0,1\})$

$c \xleftarrow{R} G$ —c→ $R \xleftarrow{R} \mathbb{Z}_p$

$y_b := g^k$

$y_{1-b} = c\, g^{-k}$

y_0, y_1 ←

IF $(y_0 y_1 \neq c)$ [ABORT]

ELSE

$r_0, r_1 \xleftarrow{R} \mathbb{Z}_p$

$c_0 := (g^{r_0}, H(y_0^{r_0}) \oplus m_0)$

$c_1 := (g^{r_1}, H(y_1^{r_1}) \oplus m_1)$

—c_0, c_1→

$c_b = (v_0, v_1)$ where $v_0 = g^{r_b}$, $v_1 = H(y_b^{r_b}) \oplus m_b$

$m_b := H(v_0^k) \oplus v_1$

Got m_b

Using Elgamal.

Q) Role of c here

Q) Can't we find the inverse of c in G

HICALI OT was ENC based, this is based on DLOG.

Sender's privacy:

- follow from CDH assumption

Receiver's privacy:

- Information theoretic.
- $y_b, y_{1-b} = c$
- Formal proof: Simulator

$SIM(m_0, m_1)$

~~same~~ $c \xleftarrow{R} G$

$y_0 y_1 \leftarrow G$

- conditioned on $y_0 y_1 = c$
- $r_0, r_1 \xleftarrow{R} R$.

o/p $SIM\left(c, y_0, y_1, (g^{r_0}, H(y_0^{r_0}) \oplus m_0), (g^{r_1}, H(y_1^{r_1}) \oplus m_1)\right)$

$\approx$ ~~SIM~~ Actual (Real World)

S: receives y_b, y_{1-b} from R

S: uses y_b, y_{1-b} to ~~decrypt~~ ENC both messages.

$\therefore$ As long as R knows a corresponding decryption key (k) for one of y_0, y_1 & not BOTH, R can only recover one of m_0, m_1.

Goal of OT

ANS 1) S's 'c' is a group element generated by $c \in \langle g \rangle$ so R cannot decrypt y_{1-b} without knowing the discrete log of c.

ANS 2) we don't get anything.

g^{r_0}, $H(g^{r_0 k}) \oplus m_0$

g^{r_1}, $H(g^{r_1 k} \cdot c) \oplus m_1$

can't get the product $(g^{-r_1 k} \cdot c)$ out from H, otherwise, we could multiply c^{-1} (complexity unknown but it is possible)

$a + (b + c) * d + e$

a b c
+ (+

a (b c +) d * e + +

[a b c + d * e + +]

PREPROCESSING

$a, b, c = ab$

GEN SHARES $(a_1\ a_2\ a_3)$ $(b_1\ b_2\ b_3)$ Via GRR

DIS $a_1 b_1$ $a_2 b_2$ $a_3 b_3$

MULL $a_1 \times b_1$ $a_2 \times b_2$ $a_3 \times b_3$

k_1 k_2 k_3

GEN SHARES $\alpha_1\ \alpha_2\ \alpha_3$ $\beta_1\ \beta_2\ \beta_3$ $\gamma_1\ \gamma_2\ \gamma_3$

DIS $\alpha_1, \beta_1, \gamma_1$ $\alpha_2\ \beta_2\ \gamma_2$ $\alpha_3\ \beta_3\ \gamma_3$

RECON with all shares $S_1(0)\alpha_1 + S_2(0)\beta_1 + S_3(0)\gamma_1 = c_1$ c_2 c_3

BEAVER'S EXAMPLE

$t=1$ $n=3$

$de = (d-a+a)(e-b+b)$

$\omega = \frac{(x_1-a+a)}{d}\frac{(x_2-b+b)}{e}$

$= de + db + ae + ab$

$= de + db + ae + c$

linear fn of (a, b, c)

(a_1, b_1, c_1) x_1 x_2 $x_1 \cdot x_2 + x_3$

(a_1, b_2, c_2) $a_3\ b_3\ c_3$

$(x_{11}\ x_{12}\ x_{13})$ $(x_{21}\ x_{22}\ x_{23})$

GSH

DLS: $x_{11}\ x_{21}$ $x_{12}\ x_{22}$ $x_{13}\ x_{23}$

$\bar{a}_1\ \bar{b}_1$ $\bar{a}_2\ \bar{b}_2$ $\bar{a}_3\ \bar{b}_3$

CALC (d_i, e_i): $d_1\ e_1$ $d_2\ e_2$ $d_3\ e_3$

Publicly recon (d, e): $d_1 d_2 d_3$ $e_1 e_2 e_3$ $d_1 d_2 e_3$ $e_1 e_2 e_3$

d, e d, e d, e

$\omega_1 = de + db_1 + ea_1 + c_1$

$\omega_2 = de + db_2 + ea_2 + c_2$

$\omega_3 = de + db_3 + ea_3 + c_3$

ω_1 ω_2 ω_3

BLIND SIGNATURE

Using RSA $(e, N)_{PR}$ public

$K_{RSA} = (e, d)_{PR, sk}$

m [message to be signed but not revealed]

$\xrightarrow{e}$

$c = m^e$

$\xleftarrow{c}$

$\sigma = c^d$

$= m^{d \cdot e}$

$\sigma = m^d$

$\xrightarrow{\sigma}$

σ = signature (blind)

Example : Bank - Envelop - carbon copy - signing.

- sign on a document

MPC - Verifiable Secret sharing.

- shares suffer from 2 problem
  - Dealer dishonest (shares don't combine properly)
  - Users can verify if data is correctly computed.
  - Uses intractibility of factoring.

MPC - Ping Pong.

Alice | Bob

$c_1 = E_B(m)$ $\xrightarrow{E_B(m)}$

$c_2 = E_A(E_B(m))$

$c_2 = E_A(m)$ $\xleftarrow{c_2}$

$m = D_A(c_2)$

swift.

Problem

~~X~~ (Adv) gets.

M which is not his own.

In passive adversary it is secure.

active adversary.

~~X~~

$c_1 = E_B(M)$ $\xrightarrow{E_B(M)}$

$M = D_B'(M)$

$c = E_X(M)$

$D_X(M)$ $\xleftarrow{c}$

FAULT TOLERANT ELECTION PROTOCOL

SETTING → VOTING

- NO DUP VOTE
- AUTHORIZED VOTER
- SECRECY OF VOTE
- TALLY COMP correctly & can be verified
- No Voter can reveal its input
- Fault tolerant.
- m Voters.
  u - any two 'f'.
- C_1, C_2 —— C_u are sent. certificates to at the end they tally.

- Use merrit election

MERRITT - ELECTION PROTOCOL

- C_1 —— C_i —— C_N

PK (pk_{R_1}) pk_{R_i} pk_{R_N}

sk_1 sk_i sk_N

- Voter voter V_j & choose candid S_j

$E_1(\ldots(E_n(E_N(V_j, S_j)))\ldots) = y_{n+1,j}$

j^{th} Voter out of m.

Yao's GC

Boolean circuit C implementing f, security parameter 1^k

GC generation:

1) Wire labels. $\omega_i^b = \left(k_i^b \in_R \{0,1\}^k , p_i^b \in_R \{0,1\} \right)$

$\omega_i^b = (k_i^b, p_i^b)$

$p_i^b = 1 - p_i^{1-b}$

$p_i^b + p_i^{1-b} = 1$

$b \in \{0,1\}$

2 a) $G_i \longrightarrow$ 2 input Boolean gate.

$g : \omega_c = g(\omega_a, \omega_b)$ $\quad \omega_a^b = (k_a^b, p_a^b)$

2 b) Create G_i's garbled table for each 2^2 possible combo of G_i's I/P. $v_a, v_b \in \{0,1\}$

$\omega_a^0 = k_a^0, p_a^0$

$\omega_a^1 = k_a^1, p_a^1$

$\omega_b^0 = k_b^0, p_b^0$

$\omega_b^1 = k_b^1, p_b^1$

} I/P labels.

$\ell_{v_a, v_b} = H\left(k_a^{v_a} \| k_b^{v_b} \| i \right) \oplus \omega_c^{g_i(v_a, v_b)}$

$\omega_c^b = k_c^b, p_c^b$] O/P labels.

Last entries to put ℓ_{v_a, v_b} in position of $\langle p_a^{v_a}, p_b^{v_b} \rangle$

3) O/P Decoding table.

LATTICE BCRYPTO

1) Regevs PKE (HE) } LWE
2) GSW FHE

+ A nad lattice based assumption.

$$[s]\underset{1\times n}{[}\quad \underset{n\times m}{A}\quad]+\underset{1\times m}{[e}\quad]\cdot[\quad]\approx$$

$$+\quad [\quad A\quad]\begin{bmatrix} \\ z \\ \\ \end{bmatrix}=0$$

01) Find z ? — Easy problem

02) Find z where 'z is short' $||z||$

$||z|| < \beta$

$\vec{z} \neq \vec{0}$

$z \leftarrow \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$

$||z|| = \sqrt{|z_1|^2 + |z_2|^2}$

Hard problem
SIS
Short Integer Solution.

+ we need to get small β but it still hard by large enough for z to have a sol?,

$\vec{y}_z = A\vec{z}$

$\vec{y}_z \in \mathbb{Z}_q^n$ $\quad \beta > \#||z|| \geq n\log q$

$\exists \vec{z}_i, \vec{z}_j$ s.t $\vec{y_z}_i = \vec{y}_{z_j}$

$\Rightarrow A(\vec{z_i} - \vec{z_j}) = 0$

+ Hash fn collision

OWF $\not\Rightarrow$ Reg SHA
Crypt - -
demo
Hash fn.

Suppose A is INV.

$A_1^{-1} A = [I_n^{\$} | A_n * B_n]$

HNF optimization

$n[\quad \underset{n}{\overset{m}{A}}\quad] = [A_1 | A_2]$

$A_1 \in \mathbb{Z}_q^{n\times n}$

$A_2 \in \mathbb{Z}_q^{n\times(m-n)}$

Pairing based crypto

Assumptions: $e: G_1 \times G_2 \to G_T$

1) BCDH (Bilinear CDH)

$g_1 \in G_1^*$ $\quad g_2 \in G_2^*$

$g_1^x, g_1^y \in G_1$ $\quad g_2^x, g_2^z \in G_2$

$x, y \xleftarrow{R} U$ (Uniform Random)

Find $e(g_1, g_2)^{xyz}$

2) BDDH

$g_1 \in G_1^*$ $\quad g_2 \in G_2^*$

$g_1^x, g_1^y \in G_1$ $\quad g_2^x, g_2^z \in G_2$

$x, y \xleftarrow{R} U$

Differentiate b/w $e(g_1, g_2)^{xyz}$ & $U \leftarrow G_T$

PROPERTIES

1) $BCDH \leq CDH(G_T)$
2) $BCDH \leq CDH(G_1)$
3) $BCDH \leq CDH(G_2)$
4) $BCDH \leq CDH$
5) $BDDH \leq DDH(G_T)$

THM: IF BCDH breaks → CDH breaks.

Proof:

BCDH Ch	BCDH Adv	CDH Adv
$x \xleftarrow{R} Z_p$		
$y \xleftarrow{R} Z_p$		
$z \xleftarrow{R} Z_p$		
g_1, g_1^x, g_1^y	→	
g_2, g_2^x, g_2^y	→	
	$e(g_1^x, g_2^z) = e(g_1, g_2)^{xz}$	
	$e(g_1^y, g_2) = e(g_1, g_2)^y$	
	$e(g_1, g_2) = \langle G_T \rangle$	
		g_T^{xz}, g_T^y →
		← g_T^{xyz}
	← $e(g_1, g_2)^{xyz}$	

BLS signature

$e: G_0 \times G_1 \to G_T$, $H: M \to G_0$

$G(\,)$: $\alpha \leftarrow Z_p$; $pk = g_1^\alpha$; $sk = \alpha$

$g_0^\alpha = u \in G_0$ ✗

$g_1^\alpha = u \in G_1$ ✓

$S(sk, m)$: $\sigma \leftarrow H(m)^\alpha \in G_0$

$V(pk, m, \sigma)$: $\sigma \in G_0$, $pk \in G_1$, $e(\sigma, g_1) \stackrel{?}{=} e(H(m), pk)$

attack game CoCDH [CDH with two groups, No bilinear pairing used]

Ch		Adv
$\alpha, \beta \xleftarrow{R} Z_p$		
$u_0 \leftarrow g_0^\alpha$		
$u_1 \leftarrow g_1^\alpha$		
$v_0 \leftarrow g_0^\beta$		
$z_0 \leftarrow g_0^{\alpha\beta}$	u_0, u_1, v_0 →	
	← $\tilde{z}_0$	
$z_0 \stackrel{?}{=} \tilde{z}_0$		

VRF

KGEN
- sk: $\alpha \xleftarrow{R} \mathbb{Z}_p$
- vk: $u \leftarrow g^{2\alpha}$

EVAL $(\text{input } x, \alpha = sk)$
- $\Pi \leftarrow H(x)^{\alpha}$
- $\sigma \leftarrow H'(\Pi)$
- o/p σ

VERIFY $(\text{input } x, \sigma, u = vk, \Pi)$
- $e(H(x), u) \stackrel{?}{=} e(\Pi, g^2)$
- $H(\Pi) = \sigma$

What is the use case

IBE based on ~~PKE~~ symmetric cipher BF'01

S()
- $\alpha \xleftarrow{R} \mathbb{Z}_p$
- $u_1 \leftarrow g_1^{\alpha}$
- mpk
- msk
- o/p (mpk, msk)

$e : G_0 \times G_1 \rightarrow G_T$

$\mathcal{E}_S = (E_S, D_S)$ on $(\mathcal{K}, M, C)$ symmetric cipher

$H_0 : ID \rightarrow G_0$

$H_1 : G_1 \times G_T \rightarrow \mathcal{K}$

G(msk, id)
- $sk_{id} \leftarrow H_0(id)^{\alpha} \in G_0$
- o/p sk_{id}

E(mpk, id, m)
- $s \xleftarrow{R} \mathbb{Z}_p$
- $\omega_1 \leftarrow g_1^{s} \in G_1$
- $z \leftarrow e(H_0(id), u_1^{s}) \in G_T$
- $k \leftarrow H_1(\omega_1, z) \in \mathcal{K}$
- $c \xleftarrow{R} E_S(k, m)$ symmetric cipher
- o/p (ω_1, c)

D(sk_{id}, (ω_1, c))
- $z = e(sk_{id}, \omega_1)$
- $k = H_1(\omega_1, z)$
- $m \leftarrow D_S(k, c)$

Process (backtrack)
- Need k, which comes from symmetric cipher.
- For k, we need z, we already have ω_1.
- For z, we can use $sk_{id} = (H_0(id)^{\alpha}, \omega_1)$

$z = e(H_0(id) u_1^{s}) = e((H_0(id)^{\alpha}, g_1^{s})$

$= e(sk_{id}, \omega_1)$

Security game IBE

Adv

Ch_{IBE}

$KGEN \to (mpk, msk)$ — $mpk \to$

$\leftarrow id$ — RO queries.

$G_1(msk, id) = sk_{id}$ — $\leftarrow sk_{id}$

$\leftarrow$ Challenge id^*, m_0, m_1

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

$c^* \leftarrow E(mpk, id^*, m_b)$ — $c^* \to$

$\leftarrow \tilde{b}$

$b \stackrel{?}{=} \tilde{b}$

Poly commitments

KzG:

$e: G_1 \times G_1 \to G_2 \quad |G_1| = p \quad G_1 = \langle g_1 \rangle$

degree d

$S()$ — $\alpha \leftarrow \mathbb{Z}_p$

$pk = (h_0 = g_1, h_1 = g_1^{\alpha}, \dots h_d = g_1^{\alpha^d})$

COMMIT

$(pk, p(x) \in \mathbb{Z}_p^d[x])$

$p(x) = p_0 + p_1 x \dots p_d x^d$

$C_p = (h_0)^{p_0} \cdot (h_1)^{p_1} \dots (h_d)^{p_d}$

$= \prod_{i=0}^{d} (h_i)^{p_i} = \prod (g_1^{\alpha^i p_i})$

$C = g_1^{p(\alpha)}$

OPEN

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

$y = p(R)$

$x - R \mid p(Rx) - y$

$\tilde{p}(x) = (x - R)\psi(x)$

$COM_{\psi} = g_1^{\psi(\alpha)}$

VERIFY (side note)

Prover — Verifier

$C_p = g_1^{p(\alpha)}$ →

← R

$y = p(R)$

$\psi(x) = \frac{p(x) - y}{x - R}$

$C_\psi = g_1^{\psi(\alpha)}$ — y, C_ψ →

$e(C_p, g_1)$

$= e(g_1^{p(\alpha)}, g_1)$

$= e(g_1^{p(\alpha) - p(R)}, g_1) \cdot e(g_1^{p(R)}, g_1)$

$= e(g_1^{(\alpha - R)\psi(\alpha)}, g_1) \cdot e(g_1^{p(R)}, g_1)$

$= e\left(g_1^{\psi(\alpha)}, \frac{g_1^{\alpha}}{g_1^{R}}\right) \cdot e(g_1, g_1)^{y}$

$= e\left(C_\psi, \frac{h_1}{g_1^{R}}\right) \cdot e(g_1, g_1)^{y}$

Thus verification can

THRESHOLD SIGNATURES

- Tuple of 4 efficient algo (G, S, V, C)
  - G → KGEN
  - S → SIGN
  - V → VERIFY
  - C → COMBINER

- KGEN $(pk, pkc, sk_1 \cdots sk_N) \xleftarrow{R} G(N, t)$
  - t ← threshold
  - N ← #shares
  - pkc ↓ combiner public key

- SIGN $\sigma_i' \xleftarrow{R} S(sk_i, m)$
  - σ_i' ↓ i^{th} signature share (Not the final sign)

- VERIFY $acc/rej \leftarrow V(pk, m, \sigma)$
  - σ ← total combined sign

- COMBINER $\sigma / \leftarrow C(pkc, m, \mathcal{J}, \{\sigma_i'\}_{j \in \mathcal{J}})$
  - blame($\mathcal{J}^*$) ↓ alternate O/P
    - $\mathcal{J}^* \subset \mathcal{J}$
    - Non empty
    - (indicating σ_j' $j \in \mathcal{J}^*$ is invalid)
  - $\mathcal{J}$ ↓ set of parties involved (subset of $\{1 \cdots N\}$ of size t)
  - $\{\sigma_i'\}_{j \in \mathcal{J}}$ ↓ set of shares

Correctness

$$P_r\left[V\left(pk, m, C\left(pkc, m, \mathcal{J}, \{S(sk_j, m)\}_{j \in \mathcal{J}}\right)\right) = accept\right] = 1$$

- Generic scheme (N, t) [Every secure signature scheme (G, S, V) can be trivially transformed to (G', S', V', C')]
  - $pkc = pk$
  - Works but performance degrades as N increases
  - Threshold t publicly known / Adversary knows how many to corrupt
  - t needs to be secure (only known to combiner)

G' [$G() \xrightarrow{R} (pk_i, sk_i)$ N times. $pk = (pk_1, pk_2 \cdots pk_N)$

Check each σ_i' individually

Rest is same as above

- BLS scheme (only one group is used)

$G'(N,t)$ [$\alpha = sk$, use shamirs secret sharing [$G_{sh}(N, t, \alpha) \rightarrow \{\alpha_1, \alpha_2 \cdots \alpha_N) \in \mathbb{Z}_q^N$ ↓ sk_1 sk_2 ---]; $g^{\alpha} = pk$; $g^{\alpha_i} = pk_i$ → shares.; $pkc := (pk_1 \cdots pk_N)$; O/P $(pk, pkc, sk_1 \cdots sk_N)$]

NOTE $\prod_{i \in \mathcal{J}} (pk_i)^{\lambda_i} = pk$

$S'(sk_i, m)$ [$\sigma_i' = H(m)^{\alpha_i} \in \mathbb{G}_1$; O/P σ_i']

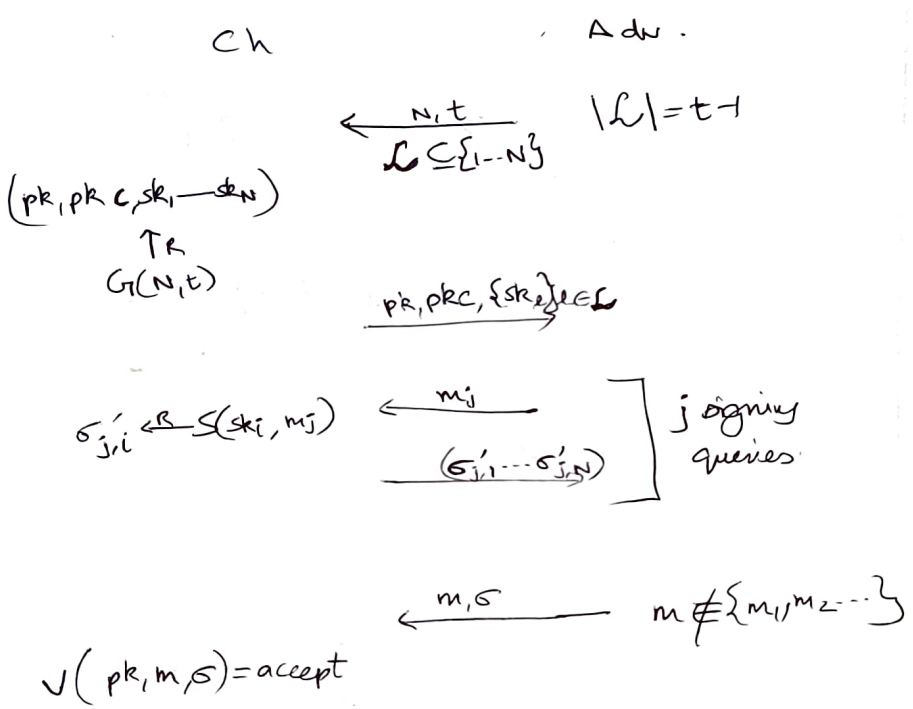
$C'((pkc, m, \mathcal{J}, \{\sigma_j'\}_{j \in \mathcal{J}})$ [1) Verify each σ_j' is valid $V(pk_j, m, \sigma_j)$ Normal BLS; 2) O/P $\mathcal{J}'$ if any shares is invalid OR proceed; 3) $\sigma \leftarrow \prod_{j \in \mathcal{J}} (\sigma_j')^{\lambda_j}$ $\lambda_j \Rightarrow$ lagrange interpolation coeff.]

$V'(pk, m, \sigma)$ [O/P $V_{BLS}^{(pk, m, \sigma)}$]

THRESHOLD SIGNA -- SECURITY

+ Adv controls
  - t-1 sign servers
  - combiner

+ ATTACK game.



~~DGHV~~ Garbled circuit

GC vs Beaver

Boolean circuit

Arithmetic ckt

Const rounds of communication needed

Processing depends on # mult gates

3-party (in preprocessing as well as live execution 3rd party involved)

2.5 party protocol (Processor not involved during live execution of protocol)

Yao's 2-party GC

Garble(1^λ)

Circuit garbling algo

(f) – boolean circuit

e – input encoding data

d – i/p decoding data

$(F, e, d) \leftarrow \text{Garble}(f)$

Encode

$x \leftarrow \text{Encode}(e, x)$ → vector of bits (input of all parties)

Eval

GC evaluation algo

$y \leftarrow \text{Eval}(F, x)$ → Garble o/p

Decode

$y \leftarrow \text{Decode}(d, y)$ → rej / vector of bits

Correctness

$$\text{Decode}\left(d, \text{Eval}\left(F, \text{Encode}(e, x)\right)\right) = f(x)$$

x = vector bits containing both P_1 & P_2 inputs

P_1 ⇄ P_2

P_1: $(F, e, d) \xleftarrow{R} \text{Garble}(f)$

$P_1 \to P_2$: x, F

Special interactive protocol

$x = \text{Encode}(e, x)$

$y = \text{Eval}(F, x)$

$P_2 \to P_1$: y

$y \leftarrow \text{Decode}(y, d)$

$P_1 \to P_2$: y

use the ~~OT~~ encryption based transfer.

PROPERTIES

1) Obliviousness (F, x) reveals nothing about x [Hides f is a stronger condition]

2) Authenticity Given (F, x) hard to find $\hat{y} \neq \text{Eval}(F, x)$ that decodes to something besides (reject)

3) O/P simulatibility y can be efficiently simulated from $f(x)$ & d.

Authenticity is only needed in malicious adversary case not curious adv case

GMW

Assumption:
semi-honest adv.
P2P to peer authenticated channel

setup (n, t)

Boolean circuit

$y = (a+b) \cdot c$

How is t chosen? which OT should be used? (whether 4-1 or 2-1)

$P_1(a)$

GShare $a = a_1 \oplus a_2 \oplus a_3$

Dis a_1, b_1, c_1

$d_1 = a_1 \oplus b_1$

r_{13}

Need to calc $d \cdot c$ (Use OTs)

c_1, d_1 → 4-1 OT P_{12} → $d_1c_2 + d_2c_1 = \alpha_{d_1c_1} + r_{12}$

$c_1 d_1, (d_1c_2 + d_2c_1) + r_{12}$

$y_1 = d_1c_1 + (d_1c_2 + d_2c_1) + r_{12} + r_{13}$

$P_2(b)$

$b = b_1 \oplus b_2 \oplus b_3$

$a_2 b_2 c_2$

$r_{12} \xleftarrow{R} U$

$d_2 = a_2 \oplus b_2$

bits (d_1, c_1)

$\alpha_{00} = 0 \cdot c_2 + 0 \cdot d_2 + r_{12}$
$\alpha_{01} = 0 \cdot c_2 + 1 \cdot d_2 + r_{12}$
$\alpha_{10} = 1 \cdot c_2 + 0 \cdot d_2 + r_{12}$
$\alpha_{11} = 1 \cdot c_2 + 1 \cdot d_2 + r_{12}$

4-1 OT P_{23}

$c_2, d_2, (d_2c_3 + d_3c_2) + r_{23}$

$y_2 = c_2d_2 + (d_2c_3 + d_3c_2) + r_{23} + r_{12}$ (P_2 created this)

$P_3(c)$

$c = c_1 \oplus c_2 \oplus c_3$

$a_3 b_3 c_3$

r_{23}

$d_3 = a_3 \oplus b_3$

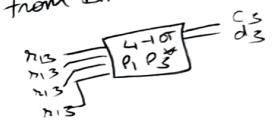
4-1 OT P_{13}

$c_3 d_3, (d_1c_3 + d_3c_1) + r_{31}$

$y_3 = c_3d_3 + (d_1c_3 + d_3c_1) + r_{13} + r_{23}$

Security proof

- Simulator for P_1, (P_2^*, P_3^*) are dishonest
  - $a_2, a_3 \xleftarrow{R} \{0,1\}$
  - sends a_2, a_3 to P_2^*, P_3^*
  - ~~gets~~ b_1, c_1 from P_2^* P_3^*
  - In Interaction with P_2^*, simulator gives $(c_1, d_1) = (0,0)$ to 4-1 OT
  - In interaction with P_2^*, simulator gets $(r_{12} + 0 \cdot c_2 + 0 \cdot d_2) = r_{12}$
  - In interaction P_3^* all I/Ps from sim are r_{13}



P_3^* always got (r_{13})

- Sim eval

SIM

P_2^*

$a_2, a_3 \xleftarrow{R} \{0,1\}$

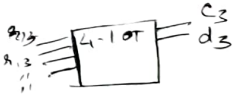
b_1, c_1

a_2, b_2, c_2

P_3^*

a_3, b_3, c_3

$d_3 = a_3 + b_3$



$y_2 = c_2d_2 + d_2c_3 + d_3c_2 + r_{23} + r_{12}$

$y_3 = c_3d_3 + r_{13} + r_{23}$

Same view as in Real World.

GMW Compiler

+ Malicious attacker
+ Wants to enforce good behavior using commitment scheme.
+ At beg' each party commits their i/p

P_i produces $c_i = com(x_i, \omega_i)$

$d_i = com(r_i, \phi_i)$

SS

28092023

Problem : DNSSEC

↓

master key distributed to 7 geographically diverse people.

↓

Atleast 5 have to meet to decrypt the root key.

(How to share the secret to 7 people & reconstruct using ~~exactly~~ atleast 5 people).

TSSS - t of n threshold secret sharing scheme

- Share : Randomized algo $\longrightarrow (s_1 s_2 \cdots s_n) = S$ (shares)

↑

$m \in M$

- Reconstruct : Determ. Algo $\longrightarrow m \in M$

↑

t-shares

- t-threshold, n-#shares

- User i receives share i.

  + $U \subseteq \{1, \ldots, n\}$ be a subset of users

  + $\{s_i \mid i \in U\}$

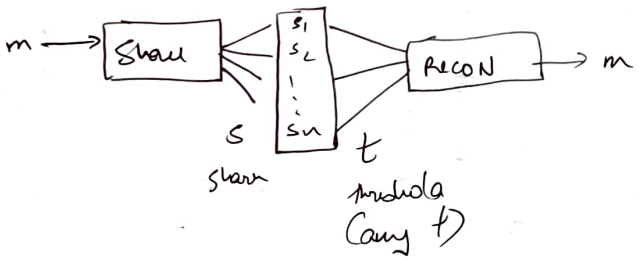
  + If $|U| \geq t$

    U authorized

    else U unauthorized

  + Unauthorized set learns nothing.

-



- Security Defn

  - If one knows only an unauthorized set of shares, then you learn no info about the choice of secret message.

28092023

Problem : DNSSEC

↓

master key distributed to 7 geographically diverse people.

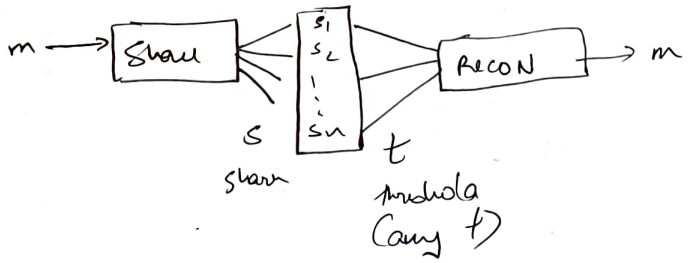
↓

Atleast 5 have to meet to decrypt the root key.

(How to share the secret to 7 people & reconstruct using ~~exactly~~ atleast 5 people).

TSSS – t of n threshold secret sharing scheme

- Share : Randomized algo $\rightarrow (s_1 s_2 \cdots s_n) = s$ (shares)
  ↑ $m \in M$
- Reconstruct : Determ. Algo $\rightarrow m \in M$
  ↑ t-shares
- t – threshold, n – # shares
- User i receives share i.
  - $U \subseteq \{1, \ldots, n\}$ be a subset of users
  - $\{s_i \mid i \in U\}$
  - If $|U| \geq t$
    U authorized
    else U unauthorized
  - Unauthorized set learns nothing.



- **Security Def^n**
  - If one knows only an unauthorized set of shares, then you learn no info about the choice of secret message.

2-2 - Shamir scheme

$s_1 \xleftarrow{R} \Sigma.\text{Keygen}$

$s_2 \leftarrow \Sigma.\text{Enc}(s_1, m)$

return (s_1, s_2)

A secure

$M = \{0,1\}^L$

$t = 2$

$n = 2$

Share(m):
$s_1 \xleftarrow{R} \{0,1\}^L$
$s_2 := s_1 \oplus m$
return (s_1, s_2)

Reconstruct(s_1, s_2):
return $s_1 \oplus s_2$

$L^{\Sigma}_{tsss\text{-}L} \equiv L^{\Sigma}_{tsss\text{-}R}$

$L^{\Sigma}_{tsss\text{-}L}$

SHARE(m_0, m_1, U):
IF $(|U| \geq 2)$: return err
$s_1 \xleftarrow{R} \{0,1\}^L$
$s_2 := s_1 \oplus m_0$
return $\{s_i \mid i \in U\}$

SHARE(m_0, m_1, U):
IF $|U| \geq 2$: return err
IF $U = \{1\}$:
$s_1 \xleftarrow{R} \{0,1\}^L$
$s_2 := s_1 \oplus m_0$
return $\{s_1\}$
ELSEIF $U = \{2\}$:
$s_1 \xleftarrow{R} \{0,1\}^L$
$s_2 := s_1 \oplus m_0$
ELSE return ϕ

SHARE(m_0, m_1, U):
IF $|U| \geq 2$: return err
IF $U = \{1\}$:
$s_1 \xleftarrow{R} \{0,1\}^L$
$s_2 := s_1 \oplus m_1$
return $\{s_1\}$
ELSEIF $U = \{2\}$:
$s_2 \xleftarrow{R}$ EAVESDROP(m_0, m_1)
return $\{s_2\}$
ELSE return ϕ

$L^{OTP}_{ots\text{-}O}$

EAVESDROP(m_0, m_1)
$R \xleftarrow{R} \{0,1\}^L$
$c := R \oplus m_0$
return c

||

$c := R \oplus m_1$

ELSE
$s_1 \xleftarrow{R} \{0,1\}^L$
$s_2 := s_1 \oplus m_1$

$L^{\Sigma}_{tsss\text{-}R}$

$s_1 \xleftarrow{R} \{0,1\}^L$
$s_2 := s_1 \oplus m_1$
return $\{s_i \mid i \in U\}$