Privacy : TP.2.0

TP 2 :

Both are agree on (G, p, g, h_0, h_1)

Q 1) ...... :

Prover		Verifier $h_0 = g^{x_0} \% p$, $h_1 = g^{x_1} \% p$
Secrets $x_0 \in \mathbb{Z}_p, x_1 \in \mathbb{Z}_p$ $r_0, r_1 = \mathrm{rand}(\mathbb{Z}_p^2)$ $s_0 = g^{r_0} \% p$, $s_1 = g^{r_1} \% p$	$\xrightarrow{s_0,\ s_1}$	$e_0, e_1 = \mathrm{rand}(\mathbb{Z}_p^2)$
	$\xleftarrow{e_0,\ e_1}$	
$\begin{cases} b_0 = (r_0 + e_0 x_0) \% p \\ b_1 = (r_1 + e_1 x_1) \% p \end{cases}$	$\xrightarrow{b_0,\ b_1}$	Check : $\begin{cases} g^{b_0} \% p = s_0 \cdot h_0^{e_0} \% p \\ g^{b_1} \% p = s_1 \cdot h_1^{e_1} \% p \end{cases}$

1) Correctness : $g^{b_i} \% p = s_i \cdot h_i^{e_i} \% p$

$$\iff g^{(r_i + e_i x_i) \% p} \% p = \left(g^{r_i} \cancel{\% p}\right) \cdot \left(g^{x_i} \cancel{\% p}\right)^{e_i} \% p$$

$$\iff g^{(r_i + e_i x_i) \% p} \% p = g^{(r_i + e_i x_i) \% p} \% p \implies \text{Correct}$$

Completeness : See correctness, if prover is honest and gives correct values then maths ensures that equations will satisfy.

~~Soundness : Prover has to be able to answer at least~~ two transcripts:

$(s, e, b) = [(s_0, s_1), (e_0, e_1), (b_0, b_1)]$

Soundness : Prover has to be able to answer right for two transcripts of the same commitment s.

$(s_0, s_1, e_0, e_1, b_0, b_1)$ and $(s_0, s_1, e_0', e_1', b_0', b_1')$

$$\begin{cases} g^{b_0} = s_0 \cdot h_0^{e_0} \ (\% p) \\ g^{b_1} = s_1 \cdot h_1^{e_1} \ (\% p) \end{cases} \text{ and } \begin{cases} g^{b_0'} = s_0 \cdot h_0^{e_0'} \ (\% p) \\ g^{b_1'} = s_1 \cdot h_1^{e_1'} \ (\% p) \end{cases} \quad \text{(Correctness)}$$

$$\begin{cases} g^{b_0 - b_0'} = h_0^{e_0 - e_0'} \ (\% p) \wedge h_0 = g^{x_0} \% p \\ g^{b_1 - b_1'} = h_1^{e_1 - e_1'} \ (\% p) \wedge h_1 = g^{x_1} \% p \end{cases} \iff \begin{cases} x_0 = \frac{b_0 - b_0'}{e_0 - e_0'} \\ x_1 = \frac{b_1 - b_1'}{\ldots} \end{cases}$$

HVZK: The verifier learns nothing except the fact the statement is true or not. We have to find a polinomial-time ~~simulator~~ simulator that produces transcript (s, e, f) that is accepted by the verifier without ~~the~~ knowing secret x. TP.2.1

$\exists$ simulator producing random (s, e, f) accepted by the verifier.

$e_0, e_1 = \text{rand}(\mathbb{Z}_q^*)$ and $f_0, f_1 = \text{rand}(\mathbb{Z}_q^*)$

$s_0 = g^{f_0} \cdot (h_0^{e_0})^{-1}$ and $s_1 = g^{f_1} \cdot (h_1^{e_1})^{-1}$

Then the verifier will check:

$$\begin{cases} g^{f_0} = s_0 \cdot h_0^{e_0} \\ g^{f_1} = s_1 \cdot h_1^{e_1} \end{cases} \iff \begin{cases} g^{f_0} = g^{f_0} \cdot \cancel{(h_0^{e_0})^{-1}} \cdot \cancel{h_0^{e_0}} \\ g^{f_1} = g^{f_1} \cdot \cancel{(h_1^{e_1})^{-1}} \cdot \cancel{h_1^{e_1}} \end{cases}$$

$\Rightarrow$ Conclusion if verifier lets pass simulator it means verifier has no ideas of secret x.

In the real world is not runnable:

If e_i is random, the attacker has to predict e_i before computing s_i and send it to verifier.

If e_i is on hash, the hash has to request s_i as parameter then (NIZK) attacker will need e_i to compute s_i but also s_i to compute e_i that is impossible.

2) $e = (e_0 = e_1)$

Correctness still hold because e_0 and e_1 were never interacting with each other before.

Soundness ~~even~~ responds at two transcripts for a same commitment still demonstrate that prover knows x.

HVZK: Same think we can treat each e_i independently where $e_i = e$ it gives the same conclusions.

$\Rightarrow$ There ~~as~~ was no interaction between these two independent random variables so all properties still hold.

3) $\cancel{g^{b_0} = s \cdot h_0^{l}}$ $\cancel{g^{b_1} = s \cdot h_1^{l}}$ $\cancel{\Leftrightarrow \frac{g^{b_0}}{h_0^{l}} = \frac{g^{b_1}}{h_1^{l}} \Leftrightarrow g^{(r+lx_0)} \cdot g^{-lx_0} =}$ TP.2.2

$b_0 = r + l x_0 \% q$, $b_1 = r + l x_1 \% q$

Verifier does: $b_0 - b_1 = (r + l x_0) - (r + l x_1) \pmod q$

$\Leftrightarrow \; = l(x_0 - x_1) \% q$

He knows l so: $\dfrac{b_0 - b_1}{l} = (x_0 - x_1) \% q$

⇒ Knowing the difference between the two secrets is a leak of info

Thus it is not zero-Knowledge anymore.

(Q2) $\boxed{\text{Chaum-Pedersen}}$ Everyone agree $(G, p, g_0, g_1, h_0, h_1)$

P		V
Secret $x \in \mathbb{Z}_q$		$h_0 = g_0^{x} \% p$, $h_1 = g_1^{x} \% p$
$r = rand(\mathbb{Z}_q)$	$\xrightarrow{s_0,\, s_1}$	$l = rand(\mathbb{Z}_q)$
$\begin{cases} s_0 = g_0^{r} \% p \\ s_1 = g_1^{r} \% p \end{cases}$	$\xleftarrow{l}$	
$b = (r + l x) \% q$	$\xrightarrow{b}$	Check: $\begin{cases} g_0^{b} \% p = s_0 \cdot h_0^{l} \% p \\ g_1^{b} \% p = s_1 \cdot h_1^{l} \% p \end{cases}$

h_0, h_1, g_0, g_1 are bases of group G

~~g is a base of G then $s \cdot g \% p$ also and $g^{s} \% p$ also~~

~~g, g' is a base of G then $g \cdot g' \%$ also~~

⚠ If g is a generator then all $g^{k} \% p \mid gcd(k, q = |G|) = 1$ are also ~~bases.~~ generators.

As p, q are primes all $g^{k} \% p \mid k \in \mathbb{Z}_q^{*}$ are generators.

$c = (c_0 = g^r, c_1 = g^m h^r)$, $d = (d_0 = g^{r'}, d_1 = g^m h^{r'})$ TP.2.3

$c = Enc^r(m)$, $d = Enc^{r'}(m)$

$v = c \cdot d^{-1} = Enc^r(m) \cdot Enc^{r'}(-m) = Enc^{r-r'}(m-m) = Enc^{r-r'}(0)$

(=)

$c \cdot d^{-1} = \left(v_0 = g^{r-r'}, v_1 = h^{r-r'}\right) = \left(\underset{(v_0)}{g^{r-r'}}, \underset{(v_1)}{g^{x(r-r')}}\right) \pmod p$

All this stuff has been generated by P!!

$\cancel{\text{So } h = v_1 \cdot v_0^{-1} = g^{x(r-r') - (r-r')}}$

As G is a cyclic group and g a generator by definition $g^k \mid k \in \mathbb{Z}_p$ is a ~~gen~~ generator and remember $a^l \% p = a^{l \% q} \% p$

So $\cancel{g_1 = g^{r-r'}}$ is a generator $\cancel{\pmod p}$, $\cancel{g_2 = g^{x(r-r')} = h^{r-r'}}$ is a generator $\pmod p$

Both g and h are generators.

We want to prove two results are the same without divulging them.

P

Secrets $m_1, m_2 \in \mathbb{Z}_p$

Private key $x \in \mathbb{Z}_p$

ElGamal: $r, r' = rand(\mathbb{Z}_p^*)$

$c = Enc^r(m_1), d = Enc^{r'}(m_2)$

$v = c \cdot d^{-1} = Enc^{r-r'}(m_1 - m_2)$

$g_1 = g, g_2 = h$

$r'' = rand(\mathbb{Z}_p)$

$$\begin{cases} a = g_1^{r''} \% p = g^{r''} \% p \\ b = g_2^{r''} \% p = h^{r''} \% p \end{cases}$$

Knows r_1, r_2 so

$R = r_1 - r_2$

$f = (r'' + R \cdot l) \% p$

P → V: $(G, p, g_1 = g, g_2 = h, h_0 = v_0, h_1 = v_1)$

P → V: $(a_0 = a, b = a_1)$

V

$l = rand(\mathbb{Z}_p)$

V → P: l

P → V: f

If:

$$\begin{cases} g^f \% p = a \cdot v_0^l \% p \\ h^f \% p = b \cdot v_1^l \% p \end{cases}$$

Then it means $m_0 == m_1$

Q3) (REGISTER) ~~(LOGIN)~~ TP.2.4

1) Client — Server

Client:

$q = |G|$

$usk = rand(\mathbb{Z}_q)$

$y = g^{usk} \% p$

~~store i~~

Messages:

- ~~Hello~~ register →
- ← (G, p, g)
- y →
- ← i

Server: store y in Y ~~at index i~~ at a random place

~~(LOGIN)~~

~~usk, y — Client — Server~~

~~login →~~

~~← $Y = (y_1, \dots, y_n)$~~

Run a Schnorr proof without revealing who I am.

(LOGIN)

Client — Server

$i \equiv$ Index where prover pub-key is.

Build a disjunction of n proofs

$r_i = rand(\mathbb{Z}_q)$

$a_i = g^{r_i} \% p$

$\forall j \mid j \neq i :: \quad l_j, b_j = rand(\mathbb{Z}_q^2)$

$\wedge\ a_j = g^{b_j} \cdot (y_j^{l_j})^{-1} \pmod p$

$l_i = l - \sum_{j \neq i} l_j$

$b_i = (r_i + usk \cdot l_i) \% p$

Messages:

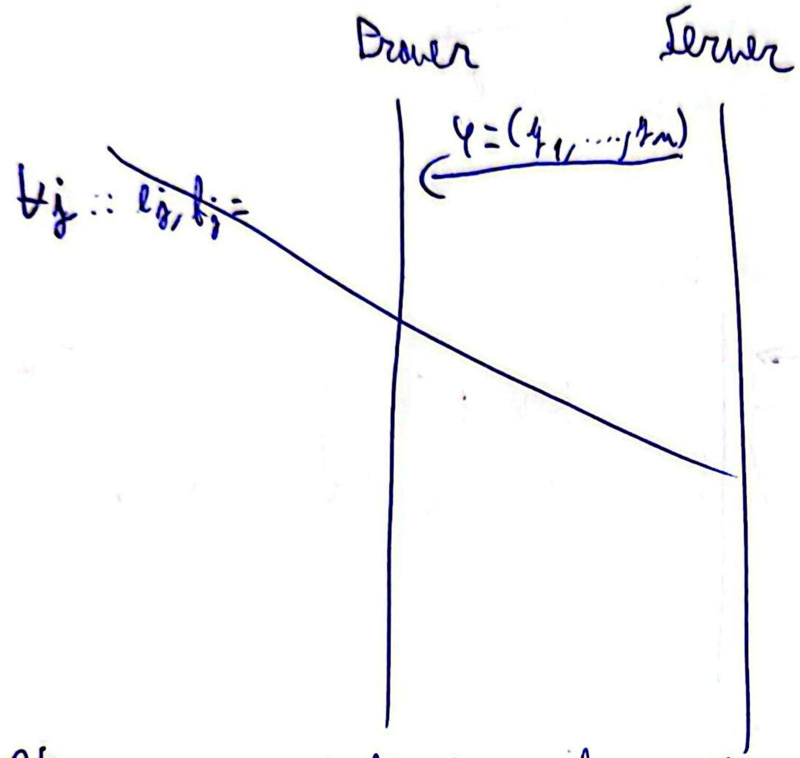
- login →
- ← $Y = (y_1, \dots, y_n)$
- $A = (a_1, \dots, a_n)$ →
- ← l
- $F = (b_1, \dots, b_n)$, $E = (l_1, \dots, l_n)$ →

Server:

$l = rand(\mathbb{Z}_q)$

Check: $\forall k :: g^{b_k} = a_k \cdot y_k^{l_k} \pmod p$

2) The server can cheat by sending a fraction of the tokens and TP.I.S if the user authenticate then the server knows that the cheat user is in this portion.



User cannot cheat by only sending random proofs as A has to be sent before receiving e, to ensure $\sum_j e_j = e$ we need to compute one e_j after but its a_j is already commited and its f_j cannot be computed.