# CS6190: ADVANCED TOPICS IN CRYPTOLOGY

## Hardness Problems in Modern Cryptography

### Discrete Logarithm Problem

Consider a group $G = \langle g \rangle$. Then given $g, \in G$, it is difficult to find $g$..

Factoring: $N = pq$ ($p, q$ are large primes)

} NOT Quantum-safe

---

Post Quantum Cryptography → lattice-based crypto, isogeny-based, hash-based, etc.

### Topics

→ ZKPs : Vipul Goyal's notes, Goldreich (Ch. 4)

→ ZKPs over Blockchains (ZKSNARKS)

→ Pairings, use in privacy preserving schemes : Dan Boneh

→ Lattice-based cryptography. (Peikert, Vinod V. . . . . )

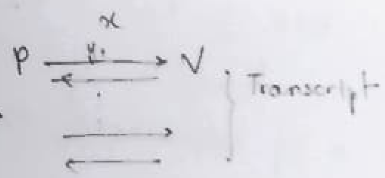
## Zero Knowledge Proofs

NP: Each language $L \in NP$ is characterized by a polynomial time recognisable relation $R_L$ such that

| Prover (P) | Verifier (V) |
|---|---|
| - Computationally unbounded | - "Easy" |
| - untrusted | - PPT |

$$\boxed{L = \{x : \exists y \text{ s.t. } (x, y) \in R_L\}, \ (x, y) \in R_L \Leftrightarrow |y| \le poly(|x|)}$$

## Interactive Proof Systems (IP)




Let $L$ be a language, and $x$ be a statement. We need to show $x \in L$. Suppose $w$ is a witness of $x \in L$. An interactive protocol for $L$ consists of the interactive PPT stateful algorithms $P$ and $V$, where

(i) $P(x, w, m_i^V) = m_{i+1}^P \quad [m_0^V = 1]$

(ii) $V(x, m_i^P) = m_{i+1}^V$

such that at the end of this protocol, $V$ outputs accept/reject and it should satisfy:

i) [Completeness] If $P, V$ are honest and $\exists w$ a valid witness for $x \in L$, $V$ outputs accept

ii) [Soundness] $\forall x \notin L$, $P^*$ using $x$, if $V$ interacts with $P^*$, $V$ rejects wp $\ge p^*$ (Soundness Parameter)

IP: Class IP consists of all languages with interactive proof systems.

Clearly, $NP \subseteq IP$,

and $P \subseteq IP$

since there is no interaction in both of these cases (further the proof is generated in poly time in the case of P, not for NP).

Exercise: $GNI \in IP$.

Zero-Knowledge An interactive proof system is perfectly zero-knowledge if $\forall x \in L$ $\forall$ PPT Verifiers $V^*$ $\exists$ a simulator $S(x, V^*)$ that outputs a transcript $\tau_{sim} = \tau_{real}$ where $\tau_{real}$ is the actual interaction between $P^*, V^*$.

→ Modifications:

- $\tau_{sim} \approx_c \tau_{real}$: computationally zero-knowledge
- $\tau_{sim} \approx_s \tau_{real}$: statistical zero-knowledge

Class heirarchy.

$$BPP \underset{\text{(most likely)}}{\subsetneq} PZK \subseteq SZK \subseteq CZK \subseteq IP$$

* If existence of one-way functions is assumed, $CZK = IP$.

Example: (ZKP for GI)

Two graphs $G_0 = (V_0, E_0)$ and $G_1 = (V_1, E_1)$ are isomorphic iff $\exists$ a permutation (bijective function on a set) $\pi : V_0 \to V_1$ s.t. $\forall (i,j) \in E_0 \Leftrightarrow (\pi(i), \pi(j)) \in E_1$

GI is not known to be in P. or in NP-complete.

→ If $G_0 \not\cong G_1$, then for any $G \cong G_0$, $G \not\cong G_1$. GI is an equivalence relation

→ Suppose $G_0 \cong G_1$ and $\mathbb{P}$ is the group of permutations on these graphs. Then, $\{\Pi(G_0) : \Pi \leftarrow \mathbb{P}\} = \{\Pi(G_1) : \Pi \leftarrow \mathbb{P}\}$

The protocol between P and V is

$(G_0, G_1)$

$(\pi)$ P $\xrightarrow{G = \sigma(G_0)}$ V

chooses random permutation $\sigma$

$\xleftarrow{b}$ V chooses $b \in_R \{0,1\}$

If $b = 0$, $k = \sigma$ else $k = \sigma \circ \pi^{-1}$ $\xrightarrow{k}$ check if $k(G_b) = G$

If yes, accept. else reject.

Completeness: Assume $G_0 \cong G_1$ and $\pi$ is a valid witness

Since $\sigma(G_0) = G$, $G \cong G_0$, $G \cong G_1$

If $b = 0$, $k = \sigma$, and $k(G_0) = \sigma(G_0) = G$

If $b = 1$, $k = \sigma \circ \pi^{-1}$ and $k(G_1) = \sigma(\pi^{-1}(G_1)) = \sigma(G_0) = G$.

Thus, the verifier accepts in both cases, as desired

Soundness: Suppose $G_0 \not\cong G_1$. Then, $\forall G$, $G \not\cong G_0$ or $G \not\cong G_1$.

⇒ $\exists b' \in \{0,1\}$ s.t $G \not\cong G_{b'}$

⇒ $\Pr(b' = b) = 0.5$

If $b = b' = 0$, $\sigma(G_0) \neq G$. V rejects

If $b = b' = 1$, $\sigma\pi^{-1}(G_1) \neq G$.

## Zero-Knowledge

1. S generates a random permutation $\phi$, $b' \in_R \{0,1\}$, sets
   $G = \phi(G_{b'})$ $[P \xrightarrow{G} V]$

2. S generates random bit $b \in_R \{0,1\}$

3. If $b = b'$ send $\phi$. else erase transcript and start from step 1.
4. Output accept, return the transcript.

Using $V^*$.

2. S runs $V^*$, sends $b$ to S
4. Run $V^*$, output what $V^*$ outputs

Exercise: Show that GNI $\in$ IP.

Consider graphs $G_0$, $G_1$. The protocol is as follows

1. Verifier chooses $i \in_R \{0,1\}$ and computes $G = \sigma(G_i)$ where $\sigma$ is an arbitrary permutation. G is sent to the prover
2. The prover computes $b$ s.t. $G_b \cong G$ and sends $b$ to the verifier.
3. The verifier accepts if $i = b$.

Completeness: If $G_0 \ncong G_1$, then $G = \sigma(G_i) \cong G_i \ncong G_{1-i}$, $i \in \{0,1\}$. Thus, the prover will always respond with $b = i$. Hence, any two $(G_0, G_1) \in$ GNI are always accepted by the verifier.

Soundness: Suppose $G_0 \cong G_1$. Then, $G = \sigma(G_i) \cong G_0 = G_1$. Thus, $b = i$ wp $1/2$. Hence this protocol is sound with soundness probability $p = 1/2$.

This is an example of Honest Verifier Zero Knowledge (HVZK) protocol. To convert to CZK, the verifier must pick $G \cong G_0$ or $G \cong G_1$ instead of any arbitrary graph

## Amplifying Soundness

Soundness Parameter ($p$) : It is essentially the probability that a dishonest prover can convince the verifier that $x \in \mathcal{L}$ when in fact $x \notin \mathcal{L}$.

To set $p$ as close to 1, performing the ZKP protocol $k$ times entirely.

$\Pr[V \text{ outputs accept all } k \text{ times} \mid x \notin \mathcal{L}] = (1-p)^k$.

For soundness of $\epsilon$, run $\log_{(1-p)} \epsilon$ trials.

## Commitment Schemes

These schemes have two properties: hiding/binding. This protocol is executed b/w a committer ($C$) and receiver ($R$), and happens in two stages:

1. Commit phase: Given a message $m$, $C$ sends $c = Com(m, s)$ for some randomness $s$ and for a PPT algo $Com$ to $R$.
2. Decommit/Reveal/Open Phase: $C$ sends $(m, s)$ and $R$ executes $c \stackrel{?}{=} Com(m, s)$. $R$ accepts if equal, else reject.

s.t. it satisfies the following two properties.

i. Hiding Property: $\forall m_0, m_1 \in \mathcal{M}$ s.t. $m_0 \neq m_1$,

$$\{Com(m_0, s) \mid s \leftarrow \mathcal{U}\} \approx_c \{Com(m_1, s) \mid s \leftarrow \mathcal{U}\}.$$

ii. Binding Property

- (Perfect binding) $\forall m_0, m_1 \in \mathcal{M}$ s.t. $m_0 \neq m_1$, $\forall s_0, s_1 \in \mathcal{U}$,

$$Com(m_0, s_0) \neq Com(m_1, s_1)$$

- (Computational binding). $\forall m_0, m_1 \in \mathcal{M}$ s.t. $m_0 \neq m_1$

$$\Pr[\exists s_0, s_1 \in \mathcal{U} \text{ s.t. } Com(m_0, s_0) = Com(m_1, s_1)]$$

is negligible.

## Decision Diffie Hellman (DDH) Problem

Consider a multiplicative cyclic group $G$ of order $q$ and generator $g$, the DDH assumption states that given $g^a$ and $g^b$, $a, b \in \mathbb{Z}_q$, the following two probability distributions are computationally indistinguishable in $\log_2 q = \lambda$ (security parameter):

1. $(g^a, g^b, g^{ab})$ $\forall a, b \in \mathbb{Z}_q$
2. $(g^a, g^b, g^c)$ $\forall a, b \in \mathbb{Z}_q$, $c \xleftarrow{\$} \mathbb{Z}_q$

## El-Gamal Commitment Scheme

Commit: Given message $m \in G$, C picks random $a, b \in \mathbb{Z}_q$ and

$s = (a, b)$, $c = (g^a, g^b \cdot m \cdot g^{ab})$

Decommit: C sends $m, (a,b)$ to R and R will check if $c = (g^a, \ldots)$

→ Binding Property: Suppose $\exists (m_1, a_1, b_1)$ s.t. $(g^{a_1}, g^{b_1}, m_1 g^{a_1 b_1})$ $(g^a, g^b, m g^{ab})$. But by property of $g$, $g^{a_1} = g^a \Rightarrow g^{a_1} g^{-a} = 1$, $a_1 = a$.

Similarly $b_1 = b$. Thus, $m_1 g^{a_1 b_1} = m g^{ab} \Rightarrow m_1 = m (g^{ab})(g^{ab})^{-1} = m$

This is perfect binding.

→ Hiding Property: From DDH.

$$\{(g^a, g^b, g^{ab})\} \approx_c \{(g^a, g^b, g^r)\} \quad \forall r \xleftarrow{\$} \mathbb{Z}_q$$

$$\Rightarrow \{(g^a, g^b, m g^{ab})\} \approx_c \{(g^a, g^b, m g^r)\} \approx \{(g^a, g^b, g^r)\}$$

This is computational hiding

## Blum Commitment Scheme

Consider a one-to-one one-way function $f: D \to R$ and a hardcore predicate $h$ of $f$. [$f \to$ mod exp, $h \to$ MSB]

Commit: Given $m \in \{0,1\}$, C samples $s$ uniformly sampled at random from D, and sends
$$c = (f(s), m \oplus h(s))$$

Decommit: C will send $(m, s)$ to R and R computes $(f(s), m \oplus h(s)) \stackrel{?}{=} c$

→ Binding Property: Suppose $(f(s_0), m_0 \oplus h(s_0)) = (f(s_1), m_1 \oplus h(s_1))$. clearly, since $f$ is injective, $s_0 = s_1 \Rightarrow h(s_0) = h(s_1)$. Thus,
$$m_0 = m_0 \oplus h(s_0) \oplus h(s_0) = m_1 \oplus h(s_1) \oplus h(s_1) = m_1.$$

→ Hiding Property: Given $f(s)$, $h(s)$ is simply a random bit. Thus, $m \oplus h(s)$ resembles an OTP.

For $r$ bit strings $(m_0, m_1, \dots, m_r)$, the commitment is
$$((f(s_0), m_0 \oplus h(s_0)), (f(s_1), m_1 \oplus h(s_1)), \dots)$$

## Pedersen Commitment Scheme

(Computationally binding scheme)

Discrete Logarithm Problem: Let $G$ be a multiplicative cyclic group of prime order $p$ with generator $g$. Given $a \in \mathbb{Z}_p$ $g^a \bmod p$, it is hard to find $a$ in PPT.

Setup:
1. Choose two very large primes $p$ and $q$ s.t. $q | p-1$.
2. Find a generator $g$ of the order $q$ which is a subgroup of $\mathbb{Z}_p^*$.
3. $a \xleftarrow{\$} \mathbb{Z}_q$ [$a$ is a secret]
4. $h = g^a \bmod p$
5. $p, q, g, h$ are public, $a$ is secret.

Commit: To commit to some $m \in \mathbb{Z}_q$, C will choose a random $r \in \mathbb{Z}_q$ and send

$$c = g^m h^r \pmod p = g^{m+ar} \pmod p$$

Decommit: C sends $(m, r)$ to R, and R computes $g^m h^r \pmod p$

Perfectly Hiding: $c = g^{m+ar} \pmod p$. We have to show $\forall m' \in \mathbb{Z}_q \ \exists r' \in \mathbb{Z}_q$ s.t. $g^{m'+ar'} \equiv g^{m+ar} \pmod p$

$$\Rightarrow m' + ar' \equiv m + ar \pmod q \Rightarrow r' = a^{-1}(m + ar - m') \pmod q$$
$$= a^{-1}(m - m') + r \pmod q$$

Computational Binding: We claim that if C can find $x'$ s.t. $x \neq x'$ and $Com(x) = Com(x')$, then C can solve the discrete logarithm problem i.e. find a given h.

Proof: From before, $a = \frac{x - x'}{r' - r}$, which means C can solve the Discrete Logarithm Problem.

Extending this scheme, $\forall m_i \in \mathbb{Z}_q$

$$Com(m_1, \ldots, m_n, r) = g^{\sum_i m_i} h^r$$

which is a single group element, unlike the Blum commitment scheme.

* Pedersen scheme is additively homomorphic i.e.

$Com(m_1, r_1) \cdot Com(m_2, r_2) = Com(m_1 + m_2, r_1 + r_2)$

In general, $\prod_i Com(m_i, r_i) = Com(\sum_i m_i, \sum_i r_i)$

Graph n-Colouring Problem

Fact 1: Suppose we are given a graph and its 3-colouring. By permuting its colours, we get another valid 3-colouring (3! permutation)

Fact 2: Suppose we are given a graph which is not 3-colourable, then any 3-colouring must contain at least one edge such that the two vertices that define the edge have the same colour

ZKP for Graph 3-Colouring

$G(V, E)$

P | | V
---|---|---
1a) P permutes the colours randomly to obtain a new valid colouring of G | 1. $c_1, \ldots, c_n$ $\longrightarrow$ $n = \lvert V \rvert$ | 2. Selects a random edge $(i, j) \in E$ and asks P to decommit the colours on $v_i$ and $v_j$
1. $\forall v_i \in V$, P sends $c_i = com(color_i, s_i)$; $color_i \to$ colour of $v_i$; $s_i \to$ randomness | $\longleftarrow$ 2. $(i, j)$ | 
3. P sends $open(c_i)$, $open(c_j)$ | 3. $(color_i, s_i)$ $(color_j, s_j)$ $\longrightarrow$ | decommits are valid and 4. If $color_i = color_j$ then reject else accept.

Completeness: Suppose that G is 3-colourable. Then, in step 1a P produces a valid 3-colouring. Hence, $open(c_i) \neq open(c_j)$ for any edge $(i, j)$ and the verifier will always accept.

Soundness: Suppose that G is not 3-colourable then from fact 2, $\exists$ at least one edge in E that does not satisfy the 3-colouring. Hence, $p = 1/|E|$

10

Zero-Knowledge Let the colours be $\{1, 2, 3\}$. Then, the simulator S works as follows:

1a. S picks a random edge $(i', j')$ and colours $k_{i'}, k_{j'} \in \{1, 2, 3\}$ at random s.t. $k_{i'} \neq k_{j'}$

b. S generates commits $c_i \; \forall v_i \in V$:

- if $i \in \{i', j'\}$, S commits according to step 1.
- else, S commits to 0

2. Invoke $V^*$ and provide commitments $c_i \; \forall v_i \in V$.

3. Upon receiving an edge $(i, j)$ from $V^*$, S does the following:

- if $i = i', j = j'$, S will reveal the colours by decommiting $c_{i'}, c_{j'}$
- else, restart and go to step 1.

Proof of zero-knowledge uses hybrid arguments, to show $\tau_{sim} \approx_c \tau_{real}$ i.e., $\tau_{sim} = H_0 \approx_c H_1 \approx_c \ldots H_n = \tau_{real}$

→ $H_0$: In this hybrid, S has a valid witness $w$ and code of $V^*$. S runs $V^*$ and its interaction. S behaves as an honest prover, and outputs transcript $H_0$.

Claim: $\tau_0 = \tau_{real}$

Proof: Trivial since S has $w$ and interaction proceeds as described.

→ $H_1$. S has valid witness $w$ and code of $V^*$. S runs $V^*$. S picks $(i', j') \xleftarrow{\$} E$ and commits to all colours honestly. When S receives $(i, j)$ from $V^*$, it restarts if $(i, j) \neq (i', j')$, else outputs the actual transcript of the communication

Claim: $\tau_1 = \tau_0 = \tau_{real}$.

- $H_2$: Sim behaves as in $H_1$, except for $i'j'$, it commits to actual colouring and for all others, it commits to 0

Claim: $\tau_2 \approx_c \tau_1$. [Hiding property of commits]

- $H_3$: Sim does not have witness w. S will sample $i'j'$ from E sample random colours $k_i' \neq k_j'$ to commit. For others, it commits to zero.

Claim: $\tau_2 = \tau_3 = \tau_{sim}$

To amplify soundness, we run k times, soundness parameter becomes $1-(1-1/|E|)^k$.

## Candidate Parallel ZKP

1. P generates k permutations $\sigma_1, \ldots, \sigma_k$:
2. In the first round, P commits to colours based on k permutations.
3. V sends k challenges to P
4. P opens commits related to k challenges. V accepts iff it would accept each of the k challenges individually, else it rejects.

## Simulator for ZKP

Sim $\qquad\qquad\qquad\qquad$ $V^*$

$(i'[1], j'[1])$
$\vdots \xleftarrow{\$} E$
$(i'[k], j'[k])$

$\xrightarrow{\text{com}(0, \ldots, c_i'[t], 0 .., c_j'[t] .. ) \ \forall t \in \{1..k\}}$

$\xleftarrow{(i[t], j[t]) \ \forall t \in \{1..k\}}$

Abort if $\exists\, p \in \{1..,k\}$ s.t. $(i[p], j[p]) \neq \{(i'[t], j'[t])\}$. Else, $\xrightarrow{\text{decommit}(i'[t], j'[t]) \ \forall t \in \{1..,k\}}$ Accepts.

Thus ZKP is a public coin protocol, since outputs of V are random. We convert it into a Non-Interactive Zero Knowledge (NIZK) Protocol using the Fiat-Shamir Transform. Here, a public hash function H is used.

$H : \{0,1\}^* \to \{0,1\}^n$

| P | | V |
|---|---|---|
| $M_1 = \mathrm{Com}(c(i); r_1)$ | | |
| $H(M_1) = M_2$ | $\xrightarrow{(M_1, M_3)}$ | Verify if $M_3 = \mathrm{resp}(H(M_1))$ and accept/reject |
| $M_3 = \mathrm{resp}(M_2)$ | | |

Examples of NIZK : ZK-SNARKS, STARKS.

__Proofs of Knowledge__

An IP system P, V for an NP-relation R is a proof of knowledge with knowledge error K if ∃ an algorithm E called an extractor that runs in expected polynomial time such that for every input y and every prover $P^*$, where x is a witness of x.

$$\Pr[(x,y) \in R : x \leftarrow E^{P^*}(y)] \geq \Pr[\langle P^*, V\rangle(y) = 1] - K$$

__Schnorr's Identification Protocol__

1. Consider a cyclic group G of prime order q with generator $g \in G$ i.e. $G = \langle g \rangle$ where DL is hard.
2. P has a secret key $\alpha \in \mathbb{Z}_q$ and a verification key $u = g^\alpha \in G$
3. P needs to prove to V that they know α without revealing α.

| P | | V |
|---|---|---|
| __Keygen__ $sk = \alpha \xleftarrow{\$} \mathbb{Z}_q$, $vk = g^\alpha \in G$ | | |
| $\alpha_t \xleftarrow{\$} \mathbb{Z}_q$, $u_t = g^{\alpha_t}$ | $\xrightarrow{u_t}$ | |
| | $\xleftarrow{c}$ | $c \xleftarrow{\$} C \subseteq \mathbb{Z}_q$ (challenge space) $\lvert C \rvert$ is superpoly in λ. |
| $z = \alpha_t + \alpha c$ | $\xrightarrow{z}$ | $g^{\alpha_t} (g^\alpha)^c \stackrel{?}{=} z$ |

Honest Verifier Zero Knowledge (HVZK)

Here, $V^*$ does nothing. If a malicious $V^*$ does not select $c$ randomly, the simulator transcript distribution is not computationally equal to a real transcript

Sim | $V^*$

$z \xleftarrow{\$} \mathbb{Z}_q$

$c \xleftarrow{\$} \mathbb{Z}_q$

$u_t = \frac{g^z}{(g^\alpha)^c}$

* Schnorr's identification scheme provides "deniability" which is not so in signing protocols.

Proof of Knowledge

We require to build an extractor $\mathcal{E}$ that runs in PPT

$\mathcal{E}$ extracts $\alpha$ from $P^*$ by rewinding $P^*$ to second interaction.

$\mathcal{E}$ | $P^*$

$\xleftarrow{u_t = g^{\alpha_t}}$

$\xrightarrow{c_1 \xleftarrow{\$} C} \quad \xrightarrow{c_2 \xleftarrow{\$} C}$

$\xleftarrow{z_1} \quad \xleftarrow{z_2}$

$z_1 = \alpha_t + \alpha c_1$, $\quad z_2 = \alpha_t + \alpha c_2$

$\Rightarrow \alpha = \frac{z_1 - z_2}{c_1 - c_2}$

PoK of Multiple Discrete Logarithms

Setup: $G = \langle g \rangle$ of prime order $q$.

Witness: $a_1, \ldots, a_m$ s.t. $A_i = g^{a_i}$

Common String: $(A_1, \ldots, A_m)$.

This is a ZK-SNARK

Succinct, Non-interactive.

P | V

Pick $a_0 \leftarrow \mathbb{Z}_q$

Set $A_0 \leftarrow g^{a_0}$

$\xrightarrow{A_0}$

$\xleftarrow{c \leftarrow \mathbb{Z}_q}$

Set $z \leftarrow \sum_{i=0}^{m} a_i c^i$ $\xrightarrow{z}$ Accept if $\prod_{i=0}^{m} A_i^{c^i} = g^z$

Attacks on Schnorr's Identification Protocol

→ Direct attack: Here, sk known $\underset{(sk)}{A} \longrightarrow C$

→ Eavesdropping attack: C acts as transcript oracle, A gets honest transcripts $(u_t, c, z)$

→ Active attack: Here, A can tamper $(u_t, c, z)$. Schnorr's protocol is not secure wrt this

Direct Attack Game

Given $I = (G, P, V)$ and adversary $A$
$G$ ↳ keygen

→ Challenger $C$ runs $G$, gets $(sk, vk)$ and sends $vk$ to $A$
→ $C$ acts as verifier, $A$ as prover
→ $A$ wins if $C$ accepts

Define $Adv_{A,I}(1^n) \triangleq Pr[Direct_{A,I}(1^n) = 1]$ as the advantage of the adversary. Then, the protocol is secure iff $Adv_{A,I}(1^n) \leq negl(n)$, where $n$ is the security parameter.

Theorem: Under DL for group $G$, and assuming $|C|$ is super-poly "n n", Schnorr's Identification Protocol is secure against direct attack

In particular if $\exists$ an efficient adversary $A$ with advantage $\epsilon$, $\exists$ an efficient adversary $B$ with advantage $\epsilon' \geq \epsilon - \frac{\epsilon}{N}$ for the DL problem.

(Forking)
Rewinding Lemma: Let $S$ and $T$ be finite nonempty sets and let $f: S \times T \to \{0,1\}$ be a function. Let $X, Y, Y'$ be mutually independent random variables where $X$ takes values in $S$ and $Y, Y'$ are uniformly distributed in $T$. Let $\epsilon \coloneqq Pr[f(X,Y) = 1]$ and $N := |T|$. Then, $Pr[f(X,Y) = 1 \wedge f(X,Y') = 1 \wedge Y \neq Y'] \geq \epsilon^2 - \frac{\epsilon}{N}$

Proof: Let $I_{sch}$ denote Schnorr's identification protocol and $A$ the PPT adversary attacking $I_{sch}$. We construct $A'$, a PPT adversary that solves the dlog problem in $G = \langle g \rangle$.

1. $A'$ is given $g^\alpha$ as an input from the challenger of dlog.
2. $A'$ sets pp: $G = \langle g \rangle$, $vk = g^\alpha$ and calls $A$ with these parameters.
3. $A$ will output $u_t$, $A'$ sends $c_1 \xleftarrow{\$} C$, and will receive $z_1$
4. $A'$ will rewind $A$, send $c_2 \xleftarrow{\$} C$, and will receive $z_2$.

5. If $g^{z_1}\cdot(g^{\alpha})^{-c_1} = u_t$, then A outputs $\frac{z_1 - z_2}{c_1 - c_2}$ else $\perp$ output.

→ Let $\omega$ be the randomness in the execution except for the challenge itself

→ Define $V(\omega, c) = 1$ if A correctly responds to challenge c for a randomness $\omega$.

→ For any fixed $\omega$ define $\delta_\omega := \Pr_c[V(\omega, c) = 1]$

→ Define $\delta(1^n) := Adv_{A,\mathcal{I}_{sch}} = \sum_\omega \delta_\omega \Pr[\omega]$

Here, A' outputs correct $\alpha$ if A succeeds twice. Thus, by rewinding lemma:

$$\Pr[\mathrm{DLog}_{A',\mathcal{G}}(1^n) = 1] = \Pr_{\omega, c}[V(\omega, c_1) = 1 \wedge V(\omega, c_2) = 1 \wedge c_1 \neq c_2]$$

$$\geq \delta(1^n)^2 - \frac{\delta(1^n)}{N} = \epsilon^2 - \frac{\epsilon}{N}$$

## Eavesdropping Attacks Against General Identification Scheme ($\mathcal{I}$)

Eavesdropping Game: For a given adversary A and the identification protocol $\mathcal{I} = (V, P_1, P_2)$, the eavesdropping attack game is as follows.

| Prover (sk) | | Verifier (pk) |
|---|---|---|
| $(I, st) \leftarrow P_1(sk)$ | $\xrightarrow{I}$ | $r \xleftarrow{\$} \mathcal{C}$ |
| | $\xleftarrow{r}$ | |
| $s \leftarrow P_2(I, sk, st, r)$ | $\xrightarrow{s}$ | $V(pk, r, s) \stackrel{?}{=} I$ |

1. Key Generation: Challenger $\mathcal{C}$ runs $(sk, vk) \leftarrow \mathrm{KeyGen}()$ and sends vk to A.
2. Eavesdropping Phase: A requires $Q(n)$ transcripts between P and V where vk is the public key, sk the secret key. $\mathcal{C}$ sends $T_1, \ldots, T_{Q(n)}$ to A.
3. Impersonation Attempt: A and $\mathcal{C}$ interact, with $\mathcal{C}$ following the algorithm of V, and A will act as the prover (not necessarily the same algorithm)

Active Game: Here, we replace the eavesdropping phase with a probing phase.

* Probing Phase: A will ask to interact with P. $\mathcal{C}$ will comply by playing the role of the prover. A will play the role of the verifier but not necessarily follow the same algorithm.

Also, A may interact with multiple instantiations of P with same (sk, vk)

**Theorem**: If $\mathcal{I}$ is secure against direct attack and is HVZK, then it is secure against eavesdropping attack.

**Proof**: If $\mathcal{I}$ is HVZK, it has a simulator Sim. Let $\mathcal{A}$ be an impersonation adversary that attacks $\mathcal{I}$ in an eavesdropping attack with access to Q transcripts.
Let $\mathcal{B}$ be a direct attack adversary that attacks $\mathcal{I}$.

**Claim**: $Adv^{direct}_{\mathcal{B},\mathcal{I}} = Adv^{eav}_{\mathcal{A},\mathcal{I}}$

**Proof**: $\mathcal{B}$ is a wrapper around $\mathcal{A}$, and supplies the transcripts to $\mathcal{A}$ from Sim. $\mathcal{B}$ will reply to C what $\mathcal{A}$ returns. Thus, $\mathcal{A}$ will be as successful as $\mathcal{B}$.

**Corollary**: $\mathcal{I}_{sch}$ is secure from an eavesdropping attack.
**Proof**: Using DL and HVZK property of $\mathcal{I}_{sch}$.

## Signatures From Identification Schemes

Let $\mathcal{I} = (Gen, P_1, P_2, V)$, $(pk, sk) \leftarrow Gen(1^n)$.

Suppose $H: \{0,1\}^* \rightarrow \mathbb{C}$ be a hash function/random oracle. Here $\mathbb{C}$ is the challenge space.

**Sign(m, sk)** $m \in \{0,1\}^*$

1. Compute $(I, st) \leftarrow P_1(sk)$
2. Compute $r := H(I, m)$
3. Compute $s := P_2(sk, st, r)$

$\Big\}$ Fiat-Shamir Transform.

Return $(r, s) \leftarrow Sign(m, sk)$

**Verify(pk, m, (r, s))**

1. Compute $V(pk, r, s) = I$
2. Check $H(I, m) \stackrel{?}{=} r$

**Note**: $H(I, m)$ can be written as $H_1(I, H_2(m))$ [Merkle Damgard Construction]

Theorem: Let $\pi$ be an identification scheme and $\pi'$ be the signature scheme that resulted by the Fiat-Shamir transform. If $\pi$ is secure against eavesdropping attacks and H is modeled as a random oracle, then $\pi'$ is existentially unforgeable.

Proof: Let $A'$ be a PPT adversary attacking $\pi'$ with $q(n)$ queries made to H. We make the following assumptions.

1. $A'$ can make any given query to H only once.
2. If the signing oracle responds with $(r, s)$ as the signature on m and $V(pk, r, s) = I$, then $A'$ never queries $H(I, m)$.
3. If $A'$ outputs a forged signature $(r, s)$ on m where $V(pk, r, s) = I$, then we assume $A'$ had previously queried $H(I, m)$.

We construct PPT A that uses $A'$ as a subroutine and attacks $\pi$.

$A(pk, \mathcal{O}_{trans})$

1. choose $j \xleftarrow{\$} \{1, \ldots, q(n)\}$
2. Run $A'(pk)$, and answer its queries as follows.
   a. When $A'$ makes its $i^{th}$ random oracle query $H(I_i, m_i)$, answer as follows:
      → if $i = j$, then output $I_j$ to V of $\pi$ and receive random challenge $\mathcal{C}$.
      → if $i \neq j$, return $k \xleftarrow{\$} \mathcal{C}$.
   b. When $A'$ requests a signature on m, answer it as follows.
      → query $\mathcal{O}_{trans}$ to obtain a transcript $(I, r, s)$
      → return $(r, s)$ as signature on $m_i$.
3. If $A'$ outputs a forgery $(r, s)$ on m compute $I := V(pk, r, s)$ and check if $I = I_j$ and $m = m_j$, then output s.

Note: $A'$'s view is almost identical to the view of the signature forgery experiment because
→ Hash queries are answered uniformly random from $\mathcal{C}$.
→ All signing queries are answered with valid signatures.
Only difference comes when $A'$ receives $(r, s)$ as a signing query on m where $H(I, m) \neq r$ (from a previous hash query).

Hence, $\mathcal{A}$'s advantage of coming up with a forgery is $Adv^{sig\text{-}forge}_{\mathcal{A}',\Pi'}(1^n) - negl(n)$.

$$\therefore Adv^{id}_{\mathcal{A},\Pi}(1^n) \geq \frac{1}{q(n)}\left[Adv^{sig\text{-}forge}_{\mathcal{A}',\Pi'}(1^n) - negl(n)\right]$$

<u>Note</u>: We assume $Id.\pi$ is nondegenerate: For a fixed $I$ and given $sk$, $Pr(I = \rho_1(sk))$ is negligible.

<u>Sigma ($\Sigma$) Protocols</u> (abstraction of $\mathcal{I}_{sch}$ for an NP-relation)

Suppose $R$ is an interactive protocol in which inputs to $P$ and $V$ are $(x, w)$ and $w$ respectively, where $(x,w) \in R$.

A $\Sigma$-protocol consists of three messages:

$P \xrightarrow{u} V$

$P \xleftarrow{c \in C} V$

$P \xrightarrow{z} V$

1. <u>Commitment</u>: $P$ sends $u$ to $V$.
2. $V$ chooses a random challenge $c \xleftarrow{\$} C$.
3. $P$ generates a response $z$, $V$ outputs accept or reject.

Properties of $\Sigma$-protocol.

1. <u>Completeness</u>: $\forall (x,w) \in R$, $Pr[P(x,w) \leftrightarrow V(x) = accept] = 1$
2. <u>Special Soundness</u>: $\exists$ an algorithm $A$ s.t. given any two accepting transcripts $(u, c, z)$ and $(u, c', z')$ and $c \neq c'$ outputs $w$ s.t. $(x, w) \in R$.
3. <u>Special HVZK</u>: $\exists$ an algorithm $M$ s.t. given $x$ and $e$ outputs $(u, e, z)$ which is distributed like a real execution where $V$ sends $e$.

<u>Lemma</u>: A $\Sigma$ protocol for an NP-relation $R$ gives an IP for $L_R$ with soundness error at most $1/|C|$

<u>Proof</u>: It is easy to see completeness.
For every commitment $u$, $\exists$ at most one challenge $c$ such that $(u, c, z)$ is accepting and $x \notin L_R$. If there is more than one such $c$, by special soundness we can find a witness $w$ for $x$ which is a contradiction.

Hence, $\Pr[V \text{ accepts} \mid x \notin L] \leq \frac{1}{|C|}$, or soundness error is at most $1/|C|$. Hence, a $\Sigma$-protocol is an IP with soundness error $1/|C|$

## Diffie-Hellman Tuples (Chaum-Pedersen Protocol)

Diffie-Hellman tuples are of the form $(g, h, u, v) \in R \iff \exists w: u = g^w, v = h^w$.

where $g, h \in G = \langle g \rangle$. Note: Skipping $g$, this is a DH triple $(g^a, g^b, g^c)$ iff $c = ab$.

The Chaum-Pedersen $\Sigma$-protocol is as follows:

$$P \; (r \xleftarrow{\$} \mathbb{Z}_q) \xrightarrow[b = h^r]{a = g^r} V$$

$$P \xleftarrow{c \xleftarrow{\$} C} V$$

$$P \xrightarrow{z = r + cw} V \quad g^z \stackrel{?}{=} au^c, \; h^z \stackrel{?}{=} bv^c$$

1. $P$ chooses $r \xleftarrow{\$} \mathbb{Z}_q$ and sends $a = g^r$, $b = g^r$
2. $V$ sends $c \xleftarrow{\$} C$
3. $P$ sends $z = r + cw$
4. $V$ checks $g^z \stackrel{?}{=} au^c$, $h^z \stackrel{?}{=} bv^c$

**Completeness.** If $(g, h, u, v)$ is a DH tuple and $w$ is its witness,

$$g^z = g^{r+cw} = g^r (g^w)^c = au^c$$
$$h^z = h^{r+cw} = h^r (h^w)^c = bv^c$$

**Special Soundness:** Given two transcripts $(a, b, e_1, z_1)$ and $(a, b, e_2, z_2)$, $A$ outputs $w \leftarrow \frac{z_1 - z_2}{e_1 - e_2}$.

**Special HVZK:** Given $c$ and $(g, h, u, v)$, $M$ chooses $z \xleftarrow{\$} \mathbb{Z}_q$ and outputs $(g^z u^{-c}, g^z v^{-c}, c, z)$.

## Other Properties of $\Sigma$-Protocols

Suppose $t$ is the size of $q$. Then,

1. Any $\Sigma$-protocol is an IP with soundness error $1/2^t$
2. Any $\Sigma$-protocol is a PoK with knowledge error $1/2^t$.
3. Properties of $\Sigma$-protocol are invariant under parallel composition.

AND of $\Sigma$-Protocols : Can run all $\Sigma$ protocols in parallel with same challenge for each.

OR of $\Sigma$-Protocols

Given two statements and two corresponding $\Sigma$-protocols, the prover must show they know the witness of at least one $\Sigma$ protocol without revealing which

Consider the $\Sigma$ protocols $(x_0, P_0)$ and $(x_1, P_1)$. The OR of these two protocols is:

1. $P \xrightarrow{a_0, a_1} V$
2. $P \xleftarrow{e} V$
3. $P \xrightarrow[z_0, z_1]{e_0, e_1 : e_0 \oplus e_1 = e} V$
4. V checks if $e_0 \oplus e_1 = e$ and $(a_0, e_0, z_0)$, $(a_1, e_1, z_1)$ are both accepting

WLOG, suppose P has witness of $x_0$ and not necessarily witness of $x_1$. P chooses $e_1 \xleftarrow{\$} C$ and runs $Sim(x_1, e_1)$ to obtain $(a_1, e_1, z_1)$. P will send $(a_0, a_1)$ to V and receives e. P then computes $e_0 = e \oplus e_1$. Thus, P can generate $(a_0, e_0, z_0)$ which is accepting. Hence, on sending $(e_0, e_1, z_0, z_1)$, V will accept.

Special Soundness:

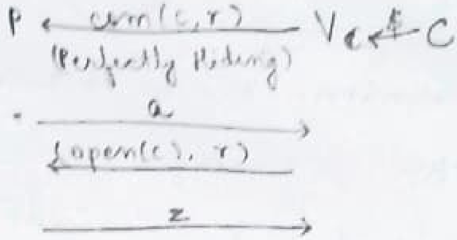
Suppose $((a_0, a_1), e, z)$ and $((a_0, a_1), e', z')$ are accepting conversations and WLOG suppose we require to extract $w_1$. Clearly, since $e' \neq e$, at least $e_0, e_0'$ or $e_1, e_1'$ are different. Hence, $(a_1, e_1, z_1)$ and $(a_1, e_1', z_1')$ are two accepting transcripts of $\Sigma_1$. $\exists$ an algorithm to find $w_1$ (similarly for $w_0$)

Special HVZK: Given $x_1, e$, select $e_1 \xleftarrow{\$} C$. Set $e_0 = e \oplus e_1$. Run $Sim(x_0, e_0)$ to get $(a_0, e_0, z_0)$ and generate $(a_1, e_1, z_1)$ using algo of $\Sigma_1$. Thus, we can generate $((a_0, a_1), e, (z_0, z_1, e_0))$ as required.

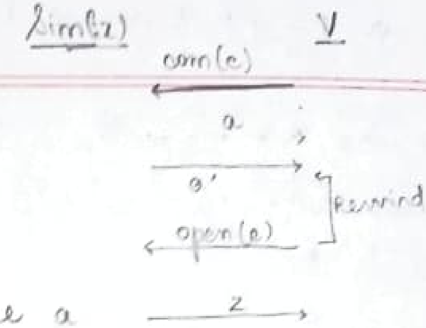
Note: • For 1 out of n, extend the OR

• For k out of n, consider the OR of all $\binom{n}{k}$ AND $\Sigma$-protocols. [NOT PPT]

## ZKP from Σ-Protocols




Here, the simulator rewinds V after finding c, runs special HVZK with inputs a, c to get an accepting conversation (a, e, z), then uses that accepting conversation to generate the transcript.




## Non-Interactive Proofs

To generate a random challenge, we use a hash function H, modeled as an RO to prove the following:

1. If the Σ-protocol is (specially) sound, then the corresponding NIZK is also sound.
2. If the Σ-protocol is special HVZK, then the NI-proof system does not reveal anything about the witness (proved using the different definition of ZKP).

## Voting Protocol

→ $n$ voters vote either 0 or 1.
→ At the end of the protocol, any voter knows the total sum of votes
→ No voter should learn about anyone else's votes (but the counter may know)
→ For encryption of the vote, we use a multiplicative variant of ElGamal.

Let $G = \langle g \rangle$ be a cyclic group of prime order $q$.

Suppose $\alpha \in \mathbb{Z}_q$ is the secret key and $u = g^{\alpha} \in G$ is the public key

* $Enc(m, u)$: $\beta \xleftarrow{\$} \mathbb{Z}_q$, $v := g^{\beta}$, $e := u^{\beta} m$. Return $(v, e)$
* $Dec((v, e), \alpha)$: $m \leftarrow e v^{-\alpha}$. Return $m$.

Vote Tallying Center (VTC): Runs keygen to generate secret key $\alpha$ and public key $u = g^{\alpha}$, available publicly to all voters.

<u>Voting Stage</u>: $i^{th}$ voter encrypts $b_i \in \{0,1\}$ as their vote by encoding $g^{b_i}$ to obtain $(v_i, e_i) \leftarrow Enc(g^{b_i}, u)$, which is published.
Here, $\beta_i \xleftarrow{\$} \mathbb{Z}_q$, $v_i = g^{\beta_i}$, $e_i = u^{\beta_i} m$

<u>Tallying Stage</u>. VTC takes all $(v_i, e_i)$ and aggregates them into a single ciphertext $(v_*, e_*) = \left(\prod_{i=1}^{n} v_i, \prod_{i=1}^{n} e_i\right)$

Thus, $v_* = g^{\sum_{i=1}^{n} \beta_i}$, $e_* = u^{\sum_{i=1}^{n} \beta_i} g^{\sum_{i=1}^{n} b_i}$. Here, $(v_*, e_*) = Enc(\sum_{i=1}^{n} b_i, u)$
Hence, VTC can decrypt $(v_*, e_*)$ to obtain $\sum_{i=1}^{n} b_i$ and publish it.
Since $\sigma$ is small (since $n$ is small), one can get $\sigma$ using a lookup table.

<u>Proving Properties on Encrypted Data</u>

Alice wants to convince the verifier that $(v, e)$ encrypts a bit under Bob's public key.
We require a $\Sigma$-protocol for $\mathcal{R} = \{((b, \beta), (u, v, e)) : v = g^{\beta}, e = u^{\beta} g^{b}, b \in \{0,1\}\}$
* If $b = 0$, $(u, v, e)$ is a DH-tuple
* If $b = 1$, $(u, v, e/g)$ is a DH-tuple.

Thus, the $\Sigma$-protocol is as follows:

Set $w_0 = e$, $w_1 = e/g$

$P((b, \beta), (u, v, e))$ $\qquad\qquad$ $V(u, v, e)$

$\beta_{tb} \xleftarrow{\$} \mathbb{Z}_q$, $v_{tb} = g^{\beta_{tb}}$, $w_{tb} = u^{\beta_{tb}}$

$d = (1-b)$, $c_d \xleftarrow{\$} \mathcal{C}$ [Idea for HVZK Sim] $\xrightarrow{v_{t0}, w_{t0}, v_{t1}, w_{t1}}$

$\beta_{zd} \xleftarrow{\$} \mathbb{Z}_q$, $v_{zd} = \dfrac{g^{\beta_{zd}}}{v^{c_d}}$, $w_{td} = \dfrac{u^{\beta_{zd}}}{w_d^{c_d}}$ $\xleftarrow{c}$

$\xrightarrow{c_0, \beta_{z0}, \beta_{z1}}$ Compute $c_1 \leftarrow c \oplus c_0$

Verify:
1. $g^{\beta_{z0}} \stackrel{?}{=} v_{t0} v^{c_0}$
2. $u^{\beta_{z0}} \stackrel{?}{=} w_{t0} w_0^{c_0}$
3. $g^{\beta_{z1}} \stackrel{?}{=} v_{t1} v^{c_1}$
4. $u^{\beta_{z1}} \stackrel{?}{=} w_{t1} w_1^{c_1}$

<u>Note</u>. Voter must now publish NIZK that $b_i \in \{0,1\}$ additionally.

<u>Note</u>. $w_{td} \neq \dfrac{u^{\beta_{zd}}}{e^{c_d}}$ as vote choice determines DH-triple whose witness known a priori by voter

## k-out-of-n / Threshold Sigma Protocols

Results from algebra:

1. Let $\mathbb{F}$ be a field. Any $(d+1)$ pairs $(a_i, b_i) \in \mathbb{F}^2$ define a unique polynomial $p$ of degree $d$ s.t. $p(a_i) = b_i \ \forall i$
2. The polynomial $p$ is constructed by interpolation
3. Given a polynomial that was interpolated from random $(a_i, b_i)$, it is impossible to identify the points used in the interpolation.

### Lagrange Interpolation Formula

Given points $(x_i, y_i)_{i=1}^{n} \in \mathbb{F}^2$, we need to find $p \in \mathbb{F}[x]$ of degree $n$. s.t. $p(x_i) = y_i$. Here, $p(x) = \sum_{i=1}^{n} \left( \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \right) y_i$.

Consider the $n$ statements $x_1, \ldots, x_n$ and suppose $A$ is the set of statements for which P has a witness, and B the set of remaining statements.

Field elements are represented as $\{1, \ldots, n\}$.

| $\underline{P}$ | | $\underline{V}$ |
|---|---|---|
| 1. $\forall i \in B$, P generates $(a_i, e_i, z_i)$ using special HVZK sims. $\forall i \in A$, P generates $a_i$ according to $\Sigma$-protocols. | $\xrightarrow{(a_1, \ldots, a_n)}$ | |
| | $\xleftarrow{e}$ | 2. $e \xleftarrow{\$} C$ |
| 3. P generates the ONLY polynomial of degree $n-k$ s.t. $f(i) = e_i \ \forall i \in B$ and $f(0) = e$. $f(x) = \sum_{j=0}^{n-k} a_j' x^j$ <br> $\forall i \in A$, P evaluates $e_i := f(i)$ and computes $z_i$ accordingly | $\xrightarrow[(z_1, \ldots, z_n)]{(e_1, \ldots, e_n)}$ | 4. Check if $f(i) = e_i \ \forall i$ and $f(0) = e$ [f is constructed from any $n-k$ points]. and $\forall i$, $(a_i, e_i, z_i)$ are accepting conversations |

If P knows $k-1$ witnesses, $n-k+1$ HVZK sims are run, resulting in a $n-k$ degree poly which will pass if $f(0) = e$, thus soundness error is $\leq 1/|C|$