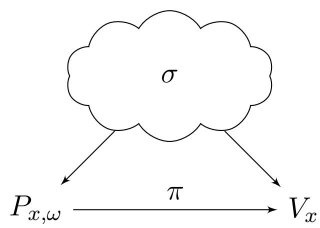
CS 276 – Cryptography March 17, 2016

Lecture 18: Non-interactive Zero-Knowledge

Instructor: Sanjam Garg *Scribe: Katia Patkin*

1 NIZK Proof system

NIZK is a class of Zero-Knowledge proof systems, where no interaction is required: The Prover sends a message to the Verifier, and the Verifier either accepts or rejects. The Prover and the verifier also have access to random public string σ.



Let R be an efficiently computable binary relation. For pairs $(x, \omega) \in R$ we call x statement and ω witness. Let L be the language consisting of statements in R.

Definition 1 *A NIZK proof system for input x in language L, with witness ω, is a set of efficient PPT algorithms (S, P, V) such that:*

1. *S produces a random public string σ: $\sigma \leftarrow S(1^k)$*
2. *Prover P produces a proof π: $\pi \leftarrow P(\sigma, x, \omega)$*
3. *Verifier V outputs 1 if accepts the proof and 0 if rejects: $0/1 \leftarrow V(\sigma, x, \pi)$*

If it has the completeness, soundness and zero-knowledge properties below.

Definition 2 (Perfect Completeness) *$\forall x \in L, \omega \in R_L(x)$:*

$$Pr[\sigma \leftarrow S(1^k); \pi \leftarrow P(\sigma, x, \omega) : V(\sigma, x, \pi) = 1] = 1$$

Definition 3 (Perfect Computational soundness) *For all polynomial size families $\{x_k\}$ s.t $x_k \notin L$ and all adversaries $\mathcal{A}$ we have*

$$Pr[\sigma \leftarrow S(1^k); \pi \leftarrow \mathcal{A}(\sigma, x_k) : V(\sigma, x_k, \pi) = 1] = 0$$

Definition 4 (Computationally perfect ZK) *If there exists a polynomial time simulator $Sim = (Sim_1, Sim_2)$, with τ is a simulation trapdoor, $\forall x \in L, \omega \in R_L(x)$ and for all non-uniform polynomial time adversaries $\mathcal{A}$ we have*

$$Pr[\sigma \leftarrow S(1^k); \pi \leftarrow P(\sigma, x, \omega) : \mathcal{A}(\sigma, x, \pi) = 1] \simeq Pr[(\sigma, \tau) \leftarrow Sim_1(1^k); \pi \leftarrow Sim_2(\sigma, \tau, x) : \mathcal{A}(\sigma, x, \pi) = 1]$$

2 Homomorphic Proof Commitments

In a non-interactive commitment scheme there is a key generator, which generates a public commitment key c_k. The commitment key c_k defines a message space $\mathcal{M}_{c_k}$, a randomizer space $\mathcal{R}_{c_k}$ and a commitment space $\mathcal{C}_{c_k}$. The commitment algorithm $\texttt{com}$: $\mathcal{M} \times \mathcal{R} \to \mathcal{C}$. The commitment scheme must be binding and hiding:

- Binding - Infeasible to find $(m_1; r_1), (m_2; r_2)$ s.t. $m_1 \neq m_2$ and $\texttt{com}(m_1; r_1) = \texttt{com}(m_2; r_2)$.
- Hiding - Given a commitment it is infeasible to guess which message is inside the commitment.

We create commitment keys with some trapdoor t_k such that we can open a commitment to any message. We want (c_k, t_k) in two modes: K_B binding mode or in K_H hiding mode. The two kinds of keys should be computationally indistinguishable.
What makes the homomorphic proof commitments special from other homomorphic commitments is that there is a way to prove that a commitment contains 0 or 1. That is if the key is perfect binding, then it is possible to prove that there exists an opening $(m, r) \in 0, 1 \times \mathcal{R}$ and if it is a perfect hiding key, then the proof will be perfectly witness-indistinguishable, i.e., it is impossible to tell whether the message is 0 or 1.

Definition 5 (Homomorphic Proof Commitment Scheme) $(K_B, K_H, com, Topen, P_{01}, V_{01})$ *is a homomorphic proof commitment scheme if it satisfies the following properties for all non-uniform polynomial time adversaries* $\mathcal{A}$*.*

Key indistinguishability

$$\texttt{Pr}[(c_k, t_k) \leftarrow K_B(1^k) : \mathcal{A}(c_k) = 1] \approx_c \texttt{Pr}[(c_k, t_k) \leftarrow K_H(1^k) : \mathcal{A}(c_k) = 1]$$

Homomorphic property

$$\forall m \leftarrow \{B, H\}, (c_k, *) \leftarrow K_m(1^k), \forall (m_1, r_1), (m_2, r_2) \in \mathcal{M} \times \mathcal{R}$$

then

$$\texttt{com}_{c_k}(m_1 + m_2; r_1 + r_2) = \texttt{com}_{c_k}(m_1; r_1) \cdot \texttt{com}_{c_k}(m_2; r_2)$$

Perfect trapdoor opening indistinguishability

$$\texttt{Pr}[(c_k, t_k) \leftarrow K_H(1^k); (m_1, m_2) \leftarrow \mathcal{A}(c_k); r_1 \leftarrow \mathcal{R}; r2 \leftarrow Topen_{t_k}(m_1, r_1, m_2) : m_1, m_2 \in \mathcal{M} \wedge \mathcal{A}(r_2) = 1]$$

$$= \texttt{Pr}[(c_k, t_k) \leftarrow K_H(1^k); (m_1, m_2) \leftarrow \mathcal{A}(c_k); r_2 \leftarrow \mathcal{R} : m_1, m_2 \in \mathcal{M} \wedge \mathcal{A}(r_2) = 1]$$

where $\texttt{com}_{c_k}(m_1; r_1) = \texttt{com}_{c_k}(m_2; r_2)$

Perfect completeness

$$\texttt{Pr}[m \leftarrow \{B, H\}; (c_k, *) \leftarrow K_m(1^k); (m, r) \leftarrow \mathcal{A}(c_k); \pi \leftarrow P_{01}(c_k, m, r) :$$

$$V_{01}(c_k, \texttt{com}(m; r), \pi) = 1 \text{ if } (m, r) \in \{0, 1\} \times \mathcal{R}] = 1$$

Perfect soundness

$$\texttt{Pr}[(c_k, t_k) \leftarrow K_B(1^k); (c, \pi) \leftarrow \mathcal{A}(c_k) : \exists(m, r) \in \{0,1\} \times \mathcal{R}, c = \texttt{com}(m; r) \text{ if } V_{01}(c_k, c, \pi) = 1] = 1$$

Perfect witness indistinguishability

$$\texttt{Pr}[(c_k, t_k) \leftarrow K_H(1^k); (r_0, r_1) \leftarrow \mathcal{A}(c_k); \pi \leftarrow P_{01}(c_k, 0, r_0) : r_0, r_1 \in \mathcal{R} \wedge \texttt{com}(0; r_0) = \texttt{com}(1; r_1) \wedge \mathcal{A}(\pi) = 1]$$

$$=$$

$$\texttt{Pr}[(c_k, t_k) \leftarrow K_H(1^k); (r_0, r_1) \leftarrow \mathcal{A}(c_k); \pi \leftarrow P_{01}(c_k, 1, r_1) : r_0, r_1 \in \mathcal{R} \wedge \texttt{com}(0; r_0) = \texttt{com}(1; r_1) \wedge \mathcal{A}(\pi) = 1]$$

2.1 Homomorphic Proof Commitments based on SHA

The setup used of the system to build the a homomorphic proof commitment scheme:
Let $\mathcal{G}$ be a randomized algorithm, that on security parameter k outputs $(p, q, \mathbb{G}, \mathbb{G}_T, e, g)$ such that

- p, q are primes with $p < q$
- $\mathbb{G}, \mathbb{G}_T$ are cyclic groups of order $n = pq$
- $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map ,i.e., $\forall u, v \in \mathbb{G}, \forall a, b \in \mathbb{Z} : e(u^a, v^b) = e(u, v)^{ab}$
- g is the generator for $\mathbb{G}$ and $e(g, g)$ generates $\mathbb{G}_T$

Subset Hiding Assumption (SHA) The SHA holds for $\mathcal{G}$, where $s \leftarrow \mathbb{Z}_n^*$ for the first and $s \leftarrow \mathbb{Z}_q^*$ for the second if:

$$(n, e, \mathbb{G}, \mathbb{G}_T, g, h = g^s) \simeq^c (n, e, \mathbb{G}, \mathbb{G}_T, g, h = g^{ps})$$

The protocol for homomorphic proof commitment scheme based on SHA:

- Perfectly binding key generation $K_B(1^k)$:
  1. $(p, q, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^k)$
  2. $n = pq$
  3. $s \leftarrow \mathbb{Z}_q^*$
  4. $h = g^{ps}$
  5. Let $c_k = (n, e, \mathbb{G}, \mathbb{G}_T, g, h)$
  6. Let $x_k = (c_k, q)$
  7. Return (c_k, x_k)
- Perfectly hiding key generation $K_H(1^k)$:
  1. $(p, q, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^k)$
  2. $n = pq$
  3. $s \leftarrow \mathbb{Z}_n^*$

4. $h = g^s$
5. Let $c_k = (n, e, \mathbb{G}, \mathbb{G}_T, g, h)$
6. Let $x_k = (c_k, q)$
7. Return (c_k, x_k)

- Commitment $\texttt{com}_{c_k}(m)$:
  1. $r \leftarrow \mathbb{Z}_n$
  2. Return $\texttt{com}_{c_k}(m; r) = g^m h^r$
- Trapdoor opening $\text{Topen}_{t_k}(m, r, m')$: Given a commitment $c = g^m h^r$ under a perfectly hiding commitment key we have $c = g^{m'} h^{r+(m-m')/s}$. The trapdoor key $t_k = (c_k, s)$. The trapdoor opening algorithm returns $r' = r + \frac{(m-m')}{s} \mod n$
- WI proof $P_{01}(c_k, m, r)$: Given $m, r \in \{0,1\} \times \mathbb{Z}_n$ we make the WI proof for commitment to 0 or 1 as $\pi = (g^{2m-1} h^r)^r$.
- Verification $V_{01}(c_k, c, \pi)$: To verify a WI proof π of commitment c containing 0 or 1, check $e(c, cg^{-1}) = e(h, \pi)$.

Theorem 1 *The protocol described above is a homomorphic proof commitment scheme if the SHA holds for $\mathcal{G}$.*

Proof.

- **Key indistinguishability:** The SHA implies that it is hard to distinguish perfect binding keys and perfect hiding keys.
- **Homomorphic:**

$$\texttt{com}_{c_k}(m_1 + m_2; r_1 + r_2) = g^{m_1+m_2} h^{r_1+r_2} = g^{m_1} h^{r_1} g^{m_2} h^{r_2} = \texttt{com}_{c_k}(m_1; r_1) \cdot \texttt{com}_{c_k}(m_2; r_2)$$

- **Perfect binding:** When h has order q.
- **Unique trapdoor opening:** When h has order n.
- **Perfect completeness:** For $m \in \{0,1\}$ we have

$$e(c, cg^{-1}) = e(g^m h^r, g^{m-1} h^r) = e(g,g)^{m(m-1)} e(h^r, g^{2m-1} h^r) = e(h, \pi)$$

- **Perfect soundness:** $c = g^m h^r$ for some uniquely defined $m \in \mathbb{Z}_p$.
  We have $e(c, cg^{-1}) = e(g,g)^{m(m-1)} e(h, (g^{2m-1} h^r)^r)$. Since h has order q, $e(h, \pi)$ has order 1 or q. The verification $e(c, cg^{-1}) = e(h, \pi)$ implies $e(c, cg^{-1})$ has order 1 or q. Since $e(c, cg^{-1}) = e(g,g)^{m(m-1)} e(h, (g^{2m-1} h^r)^r)$ we get that $e(g,g)^{m(m-1)}$ has order 1 or q. Since $e(g,g)$ is a generator for $\mathbb{G}_T$, this means $m(m-1) = 0 \mod p$ and therefore $m = 0 \mod p$ or $m = 1 \mod p$.

3 Computational NIZK Proof for Circuit SAT

This is an NIZK proof for Circuit SAT. The common reference string is a public key for a homomorphic proof commitment scheme.

- **Common reference string:**
  1. $(c_k, t_k) \leftarrow K_B(1^k)$
  2. The common reference string is $\sigma = c_k$.
- **Statement:** The statement is a circuit C built from NAND-gates. The claim is that there exist wires $\omega = (\omega_1, \ldots, \omega_{out})$ such that $C(\omega) = 1$.
- **Proof:** Input (σ, C, ω) such that $C(\omega) = 1$
  1. Commit to each bit ω_i as $r_i \leftarrow \mathcal{R}; c_i = \texttt{com}(\omega_i; r_i)$
  2. For the output wire let $r_{out} = 0$ and $c_{out} = \texttt{com}(1; 0)$
  3. $\forall c_i$ make a proof π_i of (ω_i, r_i) s.t. $\omega_i \in \{0, 1\}$ and $c_i = \texttt{com}(\omega_i; r_i)$
  4. For all NAND-gates with input wires i, j and output wire k. Using $w_i + w_j + 2w_k - 2$ and $r_i + r_j + 2r_k$ make a proof π_{ijk} for $c_i c_j c_k^2 \texttt{com}(-2; 0)$ containing 0 or 1.
  5. Return the proof π consisting of all the commitments and proofs.
- **Verification:** Input (σ, C, π).
  1. Check that all wires have a corresponding commitment and $c_{out} = com(1; 0)$.
  2. Check that all commitments have a proof of the message being 0 or 1.
  3. Check that all NAND-gates with input wires i, j and output wire k have a proof π_{ijk} for $c_i c_j c_k^2 \texttt{com}(-2; 0)$ containing 0 or 1.
  4. Return 1 if all checks pass, else return 0.