Non-Interactive ZK: The Feige-Lapidot-Shamir protocol

Ben Riva

April 20, 2009

Remainders

Definition (Interactive proof system)

A pair of interactive machines (P, V) is called an **interactive proof system** for L if V is polynomial-time and the following conditions hold:

- Completeness: for every $x \in L$,
$$\Pr[< P, V > (x) = 1] \geq 1 - \nu(x)$$
- Soundness: for every $x \notin L$ and every machine B,
$$\Pr[< B, V > (x) = 1] \leq \nu(x)$$

Remainders

Definition (Computational Zero Knowledge)

Let (P, V) be an interactive proof system for L.
We say that (P, V) is **computational zero knowledge** if for every PPT interactive machine V^* exists a PPT simulator S such that the following

$$\{< P, V^*(z) > (x)\}_{x \in L, z \in \{0,1\}^n}$$
$$\{S(x, z)\}_{x \in L, z \in \{0,1\}^n}$$

are *computationally indistinguishable*.

Motivation

- ZK with constant number of rounds is good..
- But, think of a ZK proof consists of **one message** only from P to V. It opens a door for many cryptosystems: signatures (where the signer "proves" he knows some secret), CCA-encryptions (where the encryptor "proves" he knows the plaintext), identification and etc.

Is one message NIZK possible?

Lets assume we have a protocol which is NIZK with one message
$\Rightarrow$ there exists a simulator $S \in BPP$
$\Rightarrow$ S can decide on L
$\Rightarrow$ $L \in BPP$

Proof.

If $x \in L$ then S convinces by definition.
If $x \notin L$ then, by the soundness of the NIZK, S must not be able to convince. □

So, we will go to a NIZK with **pre-processing**...

High level roadmap

- Introduce the **Common random string** (CRS) model ("realistic model") and the definition of **Non-interactive zero-knowledge**.

As a result, this yields a construction of a NIZK proof for every language in NP in the CRS model assuming the existence of trapdoor permutations .

High level roadmap

- Introduce the **Common random string** (CRS) model ("realistic model") and the definition of **Non-interactive zero-knowledge**.
- Introduce the **Hidden bits** (HB) model.

As a result, this yields a construction of a NIZK proof for every language in NP in the CRS model assuming the existence of trapdoor permutations .

High level roadmap

- Introduce the **Common random string** (CRS) model ("realistic model") and the definition of **Non-interactive zero-knowledge**.
- Introduce the **Hidden bits** (HB) model.
- Show that any NIZK proof system in the HB model can be transformed to a NIZK proof system in the CRS model, assuming the existence of trapdoor permutations.

As a result, this yields a construction of a NIZK proof for every language in NP in the CRS model assuming the existence of trapdoor permutations .

High level roadmap

- Introduce the **Common random string** (CRS) model ("realistic model") and the definition of **Non-interactive zero-knowledge**.
- Introduce the **Hidden bits** (HB) model.
- Show that any NIZK proof system in the HB model can be transformed to a NIZK proof system in the CRS model, assuming the existence of trapdoor permutations.
- Show how to construct a NIZK proof for the **Hamiltonian cycle problem** in the HB model.

As a result, this yields a construction of a NIZK proof for every language in NP in the CRS model assuming the existence of trapdoor permutations .

High level roadmap

- Introduce the **Common random string** (CRS) model ("realistic model") and the definition of **Non-interactive zero-knowledge**.
- Introduce the **Hidden bits** (HB) model.
- Show that any NIZK proof system in the HB model can be transformed to a NIZK proof system in the CRS model, assuming the existence of trapdoor permutations.
- Show how to construct a NIZK proof for the **Hamiltonian cycle problem** in the HB model.

As a result, this yields a construction of a NIZK proof for every language in NP in the CRS model assuming the existence of trapdoor permutations .

The common random string (CRS) model

Intuition[Wiki]: Captures the assumption that a trusted setup in which all involved parties get access to the same string *crs* taken from uniform distribution exists.

When used for interactive proofs

- Besides P and V, the system also consists of a PPT algorithm G.
- G gets as an input 1^k and outputs a uniform distributed string σ^k.
- Except for the regular inputs, P and V also get σ^k.
- The probabilities for the system are taken over the coin tosses of P, V and G. Note that even when we consider cheating provers, we still assume that σ^k is correctly generated.

Definition (Non interactive zero knowledge proof system)

A pair of interactive machines (P, V) is called a **non-interactive zero-knowledge** proof system for L if machine V is polynomial-time and the following conditions hold:

- Completeness: for every $x \in L$,
$$\Pr(V(x, R, P(x, R)) = 1) \geq 1 - \nu(x)$$
- Soundness: for every $x \notin L$ and every machine B,
$$\Pr(V(x, R, B(x, R)) = 1) \leq \nu(x)$$
- Zero-knowledge: there exists a PPT algorithm S such that the ensembles
$$\{(x, R_{|x|}, P(x, R_{|x|}))\}_{x \in L}, \{S(x)\}_{x \in L}$$
are *computationally indistinguishable*.

where R is a random variable uniformly distributed in $\{0,1\}^{poly(|x|)}$ (CRS).

HB model - Informal intuition

- The prover is initially given some sequence of bits which are hidden from the verifier.
- In the course of proving that $x \in L$, the prover can choose to reveal some arbitrary set of these bits to the verifier.
- The verifier never learns the bits of the string that are not revealed to it by the prover, and the prover cannot cheat and change the values in the string it is given.

Formally, we imagine that the prover is given a string R of length n and sends to the verifier (along with other information) a set of indices $I \subseteq [n]$. The verifier is then given the bits, denoted by R_I.

Definition (Proof system in the HB model)

A pair of probabilistic machines (P, V) is called a **hidden-bits** proof system for L if V is polynomial-time and the following conditions hold:

- Completeness: for every $x \in L$,
$$\Pr(V(x, R_I, I, \pi) = 1) \geq 1 - \nu(x)$$
where $(I, \pi) = P(x, R)$.
- Soundness: for every $x \notin L$ and every machine B,
$$\Pr(V(x, R_I, I, \pi) = 1) \leq \nu(x)$$
where $(I, \pi) = B(x, R)$.
- Zero-knowledge: similar to previous definition.

where R is a random variable uniformly distributed in $\{0,1\}^{poly(|x|)}$ (CRS).

Transforming from HB to CRS: The Construction

Let (P, V) be a HB proof system for L, $f : \{0,1\}^* \to \{0,1\}^*$ trapdoor permutation, and $b : \{0,1\}^* \to \{0,1\}$ hard-core of f. Follows is a construction of (P', V') in the CRS model:

- Common input $x \in \{0,1\}^n$.
- Common random string $s = (s_1, s_2, \ldots s_m)$ where each $|s_i| = n$.

- Prover P'
  - Computes $r_i = b(f^{-1}(s_i))$ for each s_i.
  - Executes P to get $(I, \pi) = P(x, r_1, r_2, \ldots r_m)$,
  - Outputs (I, π, r'_I) where $r'_I = \{f^{-1}(s_i)\}_{i \in I}$.
- Verfier V'
  - Verifies that $r_i = f(r'_i)$ for each $i \in I$. Outputs reject if something is invalid.
  - Computes $b_i = b(r'_i)$ for each $i \in I$.
  - Executes V on $(x, \{r_i\}_{i \in I}, I, \pi)$ and outputs the result.

Theorem

Provided that $Pr(b(U_n)) = 1/2$, (P', V') *is a non-interactive ZK proof system for L. Moreover, assuming the existence of families of trapdoor permutations for which membership in the family can be decided in BPP, the prover* P' *can be implemented efficiently.*

- Completeness: trivial.
- Zero knowledge: use $S(x)$ to get indexes, and select pre-images r'_i such that $b(r'_i)$ equals the revealed bits.
- Soundness: the only difference is the tdp, a cheater can choose specific *best* f.
  f works on n bits:
  $\longrightarrow$ there are $O(2^n)$ different f
  $\longrightarrow$ he can cheat w.p. $2^n * \nu(n)$
  $\longrightarrow$ we can run the protocol $O(n)$ times in order to get negligible soundness.

HB NIZK for Hamiltonian cycle in directed graph: The construction

- Common input: directed graph $G = (V, E)$ where $|V| = n$.
- Common random string: $n^3 * n^3$ boolean matrix M where each entry has 1 with probability n^{-5}.
- Definition: matrix is called **useful** if it contains $n * n$ submatrix with hamiltonian cycle and all other entries are 0 (occurs with probability $\Omega(n^{-3/2})$).

- Prover
  - If M is not useful, reveals all M's entries.
  - If M is useful, (lets denote by H the hamiltonian submatrix),
    - Reveals all entries which are not in H.
    - Outputs a $1-1$ mapping π of vertices from V to columns/rows of H.
    - Reveals the entries corresponding to *non-edges* of G.
- Verifier
  - If the prover revealed all entries of M, accept iff M is not useful.
  - Else, verifies that all entries, except for the entries $\{(\pi(u), \pi(v))|(u, v) \in E\}$, are revealed as 0.

Proofs

- Completeness and ZK are trivial.
- Efficiency: also easy (prover only computes the permutation).
- Soundness: lets assume G is non-hamiltonian. If M is useful the prover can't lie- the proof that each non-edge of G is mapped to 0-entry of H basically means that **each 1-entry of H must be mapped to an edge of G**. This means that G has hamiltonian cycle, in contradiction with our assumption. So, using the fact that $\Pr(M \text{ is useful}) = \Omega(n^{-3/2})$ we run the protocol $O(n^2)$ to get negligible soundness.

Finally..

Theorem (FLS Theorem)

Assuming existence of OWP, each language in NP has zero-knowledge non-interactive proof system. Furthermore, assuming the existence of families of trapdoor permutations for which membership in the family can be decided in BPP, the prover can be implemented efficiently.

Recycle the reference string- The problem

- For practical reasons, we prefer using the same random string for many assertions.
- But then, our simulator fails and we lose ZK.

The solution

We show how to transform any **bounded** NIZK proof system, into a **general** NIZK proof system.

General Zero-knowledge

There exists a PPT algorithm S such that for any polynomial sequence of $x_1, x_2, \ldots \in L$, the ensembles

$$\{(R_{|x|}, x_1, P(x_1, R_{|x|}), x_2, P(x_2, R_{|x|}), \ldots)\}, \{S(x_1, x_2, \ldots)\}$$

are *computationally indistinguishable*.

The construction

Let $G : \{0,1\}^l \rightarrow \{0,1\}^{2l}$ be a pseudorandom generator, and let (P, V) be a bounded NIZK proof system for NPC language L. Follows is a construction of (P', V'):

- Common input $x \in \{0,1\}^l$.
- Common random string $y|R$ where $y = \{0,1\}^{2l}$ and $R = \{0,1\}^{n-2l}$.

- Prover P'
  - Lets denote
    $$L' = \{(a, b) | a \in L \bigvee \exists w', G(w') = b\}$$
  - Using standard reduction of L' to L, reduces (x, y) to X (L's instance). It also reduces the witness W.
  - Executes P with X, common random string R and auxiliary input W, and outputs the same.
- Verifier V'
  - Reduces (x, y) to X using the same reduction.
  - Executes V with X, common random string R and P''s output, and returns the same.

Informal proofs

- Completeness: trivial.
- Soundness: P' can cheat with a negligible probability 2^{-l} (if y is in the range of G).
- Zero knowledge: the simulator simply chooses $w' \in \{0,1\}^l$ and sets $y = G(w')$. Next, he follows the protocol using (reduction of) w' as a witness for P.
  The output of the simulator is computationally indistinguishable: because 1) G is pseudorandom generator 2) ZK implies WI, therefore, the cases are indistinguishable (see FLS for the full proof).

Adaptive non-interactive zero-knowledge

Until now we dealt with non-adaptive adversaries, i.e., they choose the input message before the reference string is given to them.

Definition (Adaptive non-interactive zero-knowledge proof system)

A pair of interactive machines (P, V) is called a **adaptive non-interactive zero-knowledge** proof system for L if, for some polynomials p_1, p_2: (Completeness omitted)

- Soundness: for every $x \notin L$ and every (even unbounded) machine B,

$$\Pr(R \leftarrow \{0,1\}^{p_2(k)}; (x, \pi) \leftarrow B(R) : V(x, R, \pi) = 1) \leq \nu(k)$$

- Zero-knowledge: there exists PPT algorithms S_1, S_2 such that for all PPT adversaries A, the next are comp. indist.

$$\{R \leftarrow \{0,1\}^{p_2(|x|)}; (x, w) \leftarrow A(R); \pi \leftarrow P(R, x, w) : (R, x, \pi)\}_{x \in L}$$

$$\{(R, state) \leftarrow S_1; (x, w) \leftarrow A(R); \pi \leftarrow S_2(x, state) : (r, x, \pi)\}_{x \in L}$$

where $A()$ outputs a pair (x, w) such that $|x| \leq p_1(k)$.

Adaptization of previous constructions

Our previous constructions are very close to being adaptive-

- Zero knowledge: we can use our original simulator for the HC-NIZK system because it didn't care about the input, it just plays with the crs, so it can work also in the adaptive case.
- Soundness: instead of using one $p(k)$ random string, use $2k * p(k)$ random string and run (P, V) $2k$ times. V will accept only if **all** proofs were valid.
  Now, for each x the probability to cheat is at most 2^{-2k}, therefore, the overall probability to cheat is $2^{-2k} * 2^k$.

Done.