On the Possibility of One-Message Weak Zero-Knowledge

Johan Wallén

Helsinki University of Technology
Laboratory for Theoretical Computer Science

johan@tcs.hut.fi

Introduction

We will discuss the possibility of any meaningful type of zero-knowledge using a *one-message* (that is, *non-interactive*) proof system in the *plain model* (that is, without common reference strings, random oracles, . . .).

Our presentation is based on Boaz Barak and Rafael Pass, On the possibility of one-message weak zero-knowledge, *Theory of Cryptology Conference (TCC) 2004*, volume 2951 of *LNCS*, pages 121–132, Springer-Verlag, 2004.

It is well-known that both interaction and randomness are necessary for zero-knowledge in the plain model for a non-trivial language.

Thus, some sort of relaxation of zero knowledge is needed to obtain a one-message protocol in the plain model (unlike in the common reference string or random oracle models).

Zero-knowledge proofs and arguments

Let L be a language in $\mathbf{NP}$ and let R_L be its witness relation (that is, for all $x \in L$, there is a w of length $\mathsf{poly}(|x|)$ such that $(x, w) \in R_L$ and the relation R_L can be decided in deterministic polynomial time).

We write $R_L(x) = \{y : (x, y) \in R_L\}$, and $L(x) = 1$ if $x \in L$ and $L(x) = 0$ otherwise.

For an interactive system (P, V) for L, where V is a polynomial-time algorithm, we define the following properties:

Perfect completeness: For all $x \in L$ and witnesses w of x, V always accepts the common input x after interacting with P whose auxiliary input is w.

Zero-knowledge proofs and arguments: soundness

Soundness for proofs: For all $x \notin L$ and all P^*, the probability that V accepts the common input x after interacting with P^* is negligible.

Soundness for arguments: For all $x \notin L$ and all P^* that can be implemented by *non-uniform* polynomial-size circuits, the probability that V accepts the common input x after interacting with P^* in negligible.

For one-message (that is, non-interactive) systems, proofs and arguments are equivalent: if there is some prover strategy that makes the verifier accept, the message can be hard-coded into the non-uniform circuit.

Zero-knowledge proofs and arguments: simulation

Simulation in polynomial-time: The system (P, V) is simulatable in time $T(n) = \mathrm{poly}(n)$ if there for all probabilistic polynomial-time V^* exists a probabilistic $T(n)^{O(1)}$-time simulator S such that for all $x \in L$, $y \in R_L(x)$ and z, the view of V^* after interacting with P when the common input is x, the auxiliary input of P is y and the auxiliary input of V^* is z, $\langle P(y), V^*(z)\rangle(x)$, is computationally indistinguishable from the output $S(x, z)$ of the simulator.

That is, for all probabilistic algorithms D whose running time is polynomial in the first argument, all $x \in L$, $y \in R_L(x)$ and z,

$$|\mathrm{Pr}[D(x, z, \langle P(y), V^*(z)\rangle(x)) = 1] - \mathrm{Pr}[D(x, z, S(x, z)) = 1]|$$

is a negligible function of $|x|$, where the probability is over the coin tosses of P, V^*, S and D.

Main result

Under reasonable, but non-standard, complexity assumptions, Barak and Pass shows that every language $L \in \mathbf{NP}$ has a non-interactive system (P, V), where V is a deterministic polynomial algorithm, with the following properties:

Perfect completeness: For all $x \in L$ and $w \in R_L(x)$, $V(x, P(x, w)) = 1$.

Soundness against uniform provers: For every *uniform* probabilistic polynomial-time P^*, the probability that P^* outputs an $x \notin L$ and proof π such that $V(x, \pi) = 1$ is negligible.

This is a relaxation, since the standard definition requires soundness against *non-uniform* polynomial-size circuits.

Main result (cont.)

Quasi-polynomial-time simulation: There is a $n^{\mathsf{poly}(\log n)}$-time simulator S such that for all $x \in L \cap \{0,1\}^n$ and witnesses w for x, $S(x)$ and $P(x,w)$ are computationally indistinguishable by polynomial-size circuits.

This is a relaxation of the standard zero-knowledge property that requires a polynomial-time simulator.

The function $n^{\mathsf{poly}(\log n)}$ can be replaced with any super-polynomial function. The important thing is that the simulator is allowed to use longer running time than the cheating prover.

Note also that the zero-knowledge property is uniform—that is, non-auxiliary input.

Cryptographic assumptions (1)

The protocol relies on 3 non-standard (but reasonable) assumptions.

We assume that there is an one-message (that is, non-interactive) witness indistinguishable proof system for every language in $\mathbf{NP}$.

A witness indistinguishable proof system is simply a proof system where verifiers cannot tell the difference between the witnesses used.

More precisely, an one-message witness indistinguishable proof system (P, V) for L is a proof system such that for all $x \in L$ and $w, w' \in R_L(x)$, $P(x, w)$ and $P(x, w')$ are computationally indistinguishable.

Cryptographic assumptions (1): validity

In Barak, Ong and Vadhan, Derandomization in cryptography (Crypto 2003), it was shown that such a witness indistinguishable proof system exists, if there exist trapdoor permutations and $\mathbf{E} = \mathrm{DTIME}(2^{O(n)})$ contains a function of non-deterministic circuit complexity $2^{\Omega(n)}$.

The basic idea in the protocol is to take a two-round public-coin witness indistinguishable proof system for $\mathbf{NP}$ (such a system exist based on trapdoor permutations by Dwork and Naor, Zaps and their applications (41st FOCS, 2000)) and derandomise it.

Cryptographic assumptions (1): validity (cont.)

If $\mathbf{E}$ contains a function of non-deterministic circuit complexity $2^{\Omega(n)}$, there are (good enough) hitting set generators. Instead of sending random bits to the prover, the interactive protocol is simulated on all the elements in the hitting set as verifier messages.

Since this protocol was presented at the T-79.300 Postgraduate Course in Theoretical Computer Science seminar last autumn, we skip the details.

Cryptographic assumptions (2)

We assume that there is a non-interactive perfectly binding and computationally hiding commitment scheme that is extractable in quasi-polynomial time.

More precisely, there is an algorithm running in time $n^{\log^c n}$, where n is the security parameter and c is a constant, that given a commitment $C(x, r)$ to x recovers the message x.

Note that we assume that the hiding property holds against polynomial-time algorithms but can be broken using a quasi-polynomial time algorithm.

Cryptographic assumptions (2): validity

If there a one-way *permutation* with sub-exponential hardness, such a commitment scheme exists: simply take Blum's well-known commitment scheme with a scaled-down security parameter (see Pass, Simulation in quasi-polynomial time, and its application to protocol composition (Eurocrypt 2003) for details).

Alternatively, if there is a sub-exponentially hard one-way *function* and $\mathbf{E}$ contains a function of non-deterministic circuit complexity $2^{\Omega(n)}$, such a commitment scheme exists [Barak, Ong and Vadhan, 2003]: take Naor's well-known commitment scheme and derandomise it using a hitting-set generator.

Again, we omit the details.

Cryptographic assumptions (3)

We assume that there is a language $\Delta \in \mathbf{P}$ and constants $c_1 < c_2$ such that the following holds.

The language Δ is hard to sample (that is, generate an element of) in time $n^{\log^{c_1} n}$: for every probabilistic $n^{\log^{c_1} n}$-time algorithm A, the probability that $A(1^n) \in \Delta \cap \{0,1\}^n$ is negligible.

The language Δ is easy to sample in time $n^{\log^{c_2} n}$: there is a probabilistic $n^{\log^{c_2} n}$-time algorithm S such that the probability that $S(1^n) \in \Delta \cap \{0,1\}^n$ is greater than $1 - \mu(n)$ for some negligible function μ.

We will discuss the validity of this (new) assumption later.

The protocol

Let $L \in \mathbf{NP}$ be a language with witness relation R_L.

Let $\Delta \in \mathbf{P}$ be a language that is hard to sample in time $n^{\log^{c_1} n}$ but easy to sample in time $n^{\log^{c_2} n}$ (Assumption (3)).

Let C be a perfectly binding and computationally hiding commitment scheme that is extractable in time $n^{\log^{c_0} n}$ (Assumption (2)). By scaling the parameters, we can assume that $c_0 < c_1$.

The protocol will furthermore use a non-interactive witness indistinguishable proof system for $\mathbf{NP}$ (Assumption (1)).

The protocol (cont.)

The common input is $x \in L$ and a security parameter 1^n. By padding, we can assume that the length of x and all witnesses is n.

The prover computes a commitment $\sigma = C(0^n, r)$ to 0^n and a one-message witnesses indistinguishable proof z of the statement that $x \in L$ or there exist y, r' such that $\sigma = C(y, r')$ and $y \in \Delta$. The prover sends (σ, z) to the verifier.

The verifier simply verifies the witnesses indistinguishable proof in deterministic polynomial time.

Main theorem

Under assumptions (1)–(3), the protocol is a one-message weak zero-knowledge argument with perfect completeness and uniform (polynomial-time) soundness for $\mathbf{NP}$.

Here, weak zero-knowledge means that the protocol satisfies the uniform (that is, non-auxiliary input) zero-knowledge property under quasi-polynomial time simulation.

Proof: soundness

Suppose that there is a uniform probabilistic polynomial-time algorithm P^* that produces an accepting proof (σ, z) for some $x \notin L$.

Let y be the (unique, by perfect binding) value committed to by σ. By the perfect soundness of the witness indistinguishable proof, either $x \in L$ or $y \in \Delta$.

Since the commitment is extractable, there is an uniform algorithm E running in time $n^{\log^{c_0} n}$ that extracts y from σ.

Since $c_0 < c_1$, we get by combining P^* and E an uniform algorithm with running time bounded by $n^{\log^{c_1} n}$ that samples Δ—a contradiction.

Proof: simulation

On input x, the simulator samples $y \in \Delta$ in time $n^{\log^{c_2} n}$, computes a commitment $\sigma = C(y, r)$ to y and then computes a witness indistinguishable proof of the true statement that either $x \in L$ or $y \in \Delta$ and σ is a commitment to y.

For all $(x, w) \in R_L$, let $H = (C(y), z)$ be the hybrid distribution where $y \in \Delta$ is the value computed by the simulator and z is the witness indistinguishable proof computed by the real prover on input x using w as a witness.

The hybrid H is computationally indistinguishable from the output of the simulator by the hiding property of the commitment and by the witness indistinguishability.

By the same reason, the hybrid H is computationally indistinguishable from the output of the real prover.

Validity of Assumption (3): uniform hash functions

We will give two examples of reasonable assumptions that implies Assumption (3)—that is, that there exists a language $\Delta \in \mathrm{P}$ that is hard to sample in time $n^{\log^{c_1} n}$ but easy to sample in time $n^{\log^{c_2} n}$.

Suppose that there exists a hash function h (that is computable in polynomial time) and a constant $\epsilon > 0$ such that $|h(x)| = |x|/2$, and for every *uniform* 2^{k^ϵ}-time algorithm A, the probability that A outputs distinct $x, y \in \{0,1\}^k$ such that $h(x) = h(y)$ is negligible.

Let $\Delta = \{(1^n, x, y) : x, y \in \{0,1\}^k, x \neq y, h(x) = h(y)\}$, where $k = \log^{2/\epsilon} n$.

Note that an algorithm that runs in time less than $2^{k^\epsilon} = n^{\log n}$ cannot sample Δ, while it is trivial to sample Δ by trying all the $2^k = n^{\mathrm{poly}\log n}$ values.

Validity of Assumption (3): hardness of $\mathbf{NP} \cap \mathbf{coNP}$

Suppose that there is a unary language $L \in \mathbf{NP} \cap \mathbf{coNP}$ and a constant $\epsilon > 0$ such that for every 2^{k^ϵ}-time probabilistic algorithm A, there is for almost all i a $2^i < k \leq 2^{i+1}$ such that $A(1^k) \neq L(1^k)$.

By padding, we assume that the witnesses have the same length as the input. The language Δ consists of all tuples $(1^m, 1^i, w_{2^i+1}, b_{2^i+1}, \ldots, w_{2^{i+1}}, b_{2^{i+1}})$ such that $i = \log\log^{3/\epsilon} m$ and for all $2^i < k \leq 2^{i+1}$, w_k is a witness that $L(1^k) = b_k$.

Note that Δ is in $\mathbf{P}$, and that Δ can be sampled by searching exhaustively through all the possible witnesses in time $m^{\text{poly}\log m}$.

Validity of Assumption (3): hardness of $\mathrm{NP} \cap \mathrm{coNP}$

Suppose that A is an algorithm that on input 1^m outputs a member of Δ starting with 1^m. We construct an algorithm B for L as follows.

On input 1^k, B finds i such that $2^i < k \leq 2^{i+1}$ and m such that $i = \log\log^{3/\epsilon} m$. It then runs $A(1^m)$ to obtain $(1^m, 1^i, w_{2^i+1}, b_{2^i+1}, \ldots, w_{2^{i+1}}, b_{2^{i+1}}) \in \Delta$ and outputs b_k. This takes at most $m^{\log m} = 2^{\log^2 m}$ steps, and thus the running time of B is less than 2^{k^ϵ}.

Necessity of Assumption (3)

Suppose that there exists one-way injections hard against quasi-polynomial time algorithms and that there is a one-message weak zero-knowledge argument with uniform soundness for $\mathbf{NP}$. Then there is a language Δ that is hard to sample by polynomial-time algorithms but can be sampled by a quasi-polynomial time algorithm.

Note that this is a weakening of Assumption (3) only with respect to the hardness of sampling.

Necessity of Assumption (3): proof

Let f be a one-way function as in the statement, and let h be its hard-core bit. Let $L = \{(f(x), h(x)) : x \in \{0,1\}^*\} \in \mathbf{NP}$. Let V be the verifier algorithm for the weak zero-knowledge argument system for L.

Define $\Delta = \{(y, b, \pi, x) : y = f(x), b \neq h(x), V(y, b, \pi) = 1\}$. That is, Δ is the languages of "false proofs".

By the uniform soundness of the zero-knowledge system, Δ cannot be sampled by uniform polynomial-time algorithms.

Let A be an algorithm that on input 1^n picks $x \in \{0,1\}^n$ and $b \in \{0,1\}$ at random, and outputs $(f(x), b, \pi, x)$, where π is obtained by applying the zero-knowledge simulator to the statement $(f(x), b)$. The running time is clearly $n^{\mathrm{poly}\log n}$.

Necessity of Assumption (3): proof (cont.)

Note that the probability that $V(f(x), b, \pi) = 1$ is very close to 1: otherwise, the simulator and verifier forms a distinguisher for $(f(x), b)$ and $(f(x), h(x))$ contradicting the hard-coreness of h for f.

Note furthermore that the probability that $b \neq h(x)$ is $1/2$, since b is chosen independently.

Thus, A outputs a member of $\triangle$ with probability very close to $1/2$.

Since membership in $\triangle$ can be verified, this probability can be amplified to be negligibly close to 1.