How many rounds can Random Selection handle?[*]

Shengyu Zhang[†]

Abstract

The construction of zero-knowledge proofs can be greatly simplified if the protocol is only required be secure against the honest verifier. Damgård, Goldreich and Wigderson invented a *Random Selection* technique to transform a public-coin honest verifier zero-knowledge protocol to a general zero-knowledge protocol. Though all the three variants of Zero Knowledge, *i.e.* Perfect Zero-Knowledge(PZK), Statistical Zero-Knowledge(SZK), Computational Zero-Knowledge(CZK) can be handled, their method, however, only applies to constant-round protocols. Later, Goldreich, Sahai and Vadhan extended the transformation result to being able to handle the general SZK and CZK protocols of a polynomial number of rounds, but their extension fails on the PZK case. In this paper we show that applying the original DGW transformation technique twice in a nested way enables the transformation handle $O(\log(n)/\log\log(n))$ rounds.

Key words: cryptography, zero knowledge

1 Introduction

Zero-knowledge (ZK) proofs are interactive proofs yielding nothing beyond the validity of the statement being proven. Since their invention [8, 6], zero-knowledge proofs have been extensively studied for more than two decades, and have reinforced their central position of the modern cryptography. More formally, a proof is zero-knowledge if there is an efficient procedure, called the *simulator*, to simulate the transcript of the real prover-verifier interaction. Three variants are defined — Perfect Zero-Knowledge (PZK), Statistical Zero-Knowledge(SZK) and Computational Zero-Knowledge (CZK), according to different requirements of the simulation quality.

To make the zero-knowledge proofs robust to be useful in many cryptography settings, it is required that the verifier cannot learn anything even if it deviates its behavior from the specified protocol in an arbitrary way. If we only require a protocol to be zero-knowledge against the *honest verifier*, *i.e.* the verifier who behaves exactly as specified in the protocol, then the construction is usually much easier. Therefore, it is desirable to have a generic procedure transforming an arbitrary honest verifier zero-knowledge protocol to a general zero-knowledge protocol.

Considerable attention has been paid to these transformations [1, 9, 2, 3, 7]. Damgård, Goldreich and Wigderson [3] invented a technique to transform a constant-round public-coin honest

[*] This work was supported by the National Science Foundation under grant PHY-0456720 and the Army Research Office under grant W911NF-05-1-0294 through the Institute for Quantum Information at the California Institute of Technology.

[†] Computer Science Department and Institute for Quantum Information, California Institute of Technology, 1200 E California Bl, Pasadena, CA 91125, USA. Email: shengyu@caltech.edu

verifier PZK/SZK/CZK protocol to a general PZK/SZK/CZK protocol (against arbitrary verifiers). The technique they used was referred to as *Random Selection*. Their results were later extended by Goldreich, Sahai and Vahdan [7] to the general polynomial-round protocols for SZK and CZK cases by introducing other hashing structures. This structure, however, introduces new deviation in the simulation, so it does not apply to the PZK case.

One shortcoming of the DGW Random Selection technique is that it seems only to work for a constant number of rounds by itself because of the competing requirements of the zero-knowledge and soundness properties. In this paper, we show that applying it twice in a nested way can boost the success probability of the zero-knowledge simulator, allowing us to handle PZK protocols of $c \log n / \log\log n$ rounds, where c is any constant.

2 Preliminaries

In this section we give the definition of zero-knowledge protocols and review the Random Selection transformation. For basic notions like interactive proofs, statistical difference and computational indistinguishibility, we refer readers to, for example, [4].

Definition 1 *An interactive proof* (P, V) *is* zero-knowledge *if for any probabilistic polynomial time machine* V^**, called the cheating verifier, there exists a probabilistic polynomial time machine* S^**, called the simulator,* s.t. *for all* $x \in L$

$$\langle P, V^* \rangle (x) \sim S^*(x). \tag{1}$$

where $\langle P, V^* \rangle (x)$ *is the* view *of* V^* *on* x*, consisting of three parts:* V^**'s random string, all messages from* P *and the final acceptance/rejection decision.* $S^*(x)$ *is the (random) output of* S^* *on* x*, and the symbol* $\sim$ *has different meanings for different settings as follows:*

1. Perfect Zero-Knowledge *(PZK):* $S^*(x)$ *may output "Fail" with probability less than 1/2, but conditioned on the output is not "Fail", it is identically distributed as* $\langle P, V^* \rangle (x)$*.*

2. Statistical Zero-Knowledge *(SZK): The statistical difference between* $S^*(x)$ *and* $\langle P, V^* \rangle (x)$ *is negligible,* i.e. $o(1/poly(n))$*.*

3. Computational Zero-Knowledge *(CZK): The ensembles* $\{\langle P, V^* \rangle (x)\}_{x \in L}$ *and* $\{S^*(x)\}_{x \in L}$ *are computationally indistinguishable.*

The constant 1/2 in the PZK definition is not crucial; any constant gives the same class of functions.

Though the above definition does not deal with the more general auxiliary inputs case, it is easy to see that results in both previous work [3, 7] and the current paper hold for ZK with auxiliary inputs too. For more detailed introductions of the area, see [4, 5] for excellent surveys.

Next we review the DGW Random Selection procedure [3]. First, for any integers $0 < t < s$, there is a family $\mathcal{F}^d_{s,t}$ of almost d-wise independent hash functions from $\{0,1\}^s$ to $\{0,1\}^t$ *s.t.*

1. Every function $f \in \mathcal{F}^d_{s,t}$ can be described by an sd-bit string;

2. $\forall y \in \{0,1\}^t$, the *cell* y, *i.e.* the set $f^{-1}(y)$, has size $1 \le |f^{-1}(y)| \le d2^{s-t}$;

3. $\forall y \in \{0,1\}^t$, the whole cell y can be computed in time polynomial in $d2^{s-t}$. (Note that we will later set $d = s = poly(n)$ and $s - t = O(\log n)$, so 2^{s-t} is still $poly(n)$.)

4. $\mathcal{F}^d_{s,t}$ is a family of almost d-wise independent hashing functions in the following sense: for every d distinct images, $x_1, ..., x_d \in \{0,1\}^s - \{0,1\}^t 0^{s-t}$, for a uniformly chosen $f \in \mathcal{F}^d_{s,t}$, the random variables $f(x_1), ..., f(x_d)$ are independently and uniformly distributed in $\{0,1\}^t$.

Now suppose that in some round, the honest verifier V sends its public coin random string $\alpha \in_R \{0,1\}^s$ and the prover P responds with β. Here the notation $a \in_R A$ means to pick an element a uniformly at random from the set A. Then the Random Selection transforms this one-round protocol to the following two-round protocol (P_1, V_1) as follows.

1. V_1 chooses $f \in_R \mathcal{F}^s_{s,t}$ and sends f to P_1.
2. P_1 chooses $y \in_R \{0,1\}^t$ and sends y to V_1.
3. V_1 chooses $\alpha \in_R f^{-1}(y)$ and sends α to P_1.
4. P_1 checks whether $f(\alpha) = y$ and terminates the protocol if not. Otherwise, P_1 responds with a message β in the same way as P does when seeing α.

It is proved in [3] that the DGW protocol satisfies the following simulation deviation property.

Fact 1 *For the honest prover P_1, no matter what strategy V_1 uses, as long as $\alpha \in f^{-1}(y)$, the α is at most $2s2^{-(s-t)/4} + 2^{-s}$ away from the uniform distribution in $\{0,1\}^s$ (under the total variance distance).*

We denote the original r-round public-coin protocol by $(\alpha_1, \beta_1, ..., \alpha_r, \beta_r)$, which means that in the round i, the verifier sends its random string $\alpha_i \in_R \{0,1\}^s$ and the prover responds with message β_i. Then for this general r-round protocol, the DGW transformation uses the above procedure to transform each (α_i, β_i) and gets a new protocol $(f_1, y_1, \alpha_1, \beta_1, ..., f_r, y_r, \alpha_r, \beta_r)$. Here β_i is chosen the same way by P_1 as by P on seeing the previous history $\alpha_1, ..., \alpha_i$. By the above fact, it is immediate that the DGW protocol satisfies the following soundness property.

Fact 2 *Suppose the original r-round public coin protocol has the soundness error ϵ_0, then the DGW transformation gives a new protocol with the soundness error*

$$\epsilon_1 = \epsilon_0 + r \cdot (2s2^{-(s-t)/4} + 2^{-s}). \tag{2}$$

The soundness is proved by the following hashing lemma.

Fact 3 *For any fixed subset $A \subseteq \{0,1\}^s$ of size at least 2^{s-1}, we have*

$$\mathbf{Pr}_{f \in_R \mathcal{F}^s_{s,t}} \left[\left| |f^{-1}(y) \cap A| - |A|/2^t \right| > \epsilon |A|/2^t, \ \forall y \in \{0,1\}^t \right] \leq \left(\frac{2s}{\epsilon 2^{(s-t)/2}} \right)^s \tag{3}$$

For the zero-knowledge property, one can define the following simulator S_1^* for any cheating verifier V_1^* as follows [3]. Suppose the random string used by V_1^* is of length l.

1. Run the original honest verifier simulator S for (P, V) and get a transcript $(\alpha_1, \beta_1, ..., \alpha_r, \beta_r)$. For the PZK case, if S fails, then S_1^* also outputs "Fail" and terminates the simulation.
2. Pick $rand \in_R \{0,1\}^l$ and use $rand$ as V_1^*'s random string throughout the following simulation.
3. for $i = 1$ to r

(a) Get the $f_i \in \mathcal{F}_{s,t}^d$ from V_1^* after feeding β_{i-1} to V_1^* in the last round. (For the first round, V_1^* initializes the protocol).

(b) Let $y_i = f_i(\alpha_i)$ and feed y_i to V_1^*.

(c) Get the α_i' from V_1^*

(d) If $\alpha_i' = \alpha_i$, feed β_i to V_1^*; else, output "Fail" and terminate the simulation.

4. Output $(rand; y_1, \beta_1, ..., y_r, \beta_r; decision)$ to simulate V_1^*'s view, where $decision \in \{0,1\}$ is the acceptance/rejection bit of V_1^* in the last step.

Note that after its randomness fixed, the verifier V_1^* is a deterministic machine. In particular, it will output one particular α_i', which we will call *representative*, for each cell y_i. We say S_1^* *passes* round i if it does not output "Fail"' in Step 3d. It is not hard to see that S_1^* passes round i if and only if α_i is one of these 2^t representatives. For the PZK case which we will mainly focus on, since the honest verifier simulator S outputs all α_i uniformly and independently at random, we have the following fact.

Fact 4 *For the originally r-round PZK protocol (P, V), the DGW simulator S_1^* for the new protocol (P_1, V_1^*) passes all rounds with probability $2^{-(s-t)r}$, for any possibly cheating verifier V^*. In addition, conditioned on the event that S_1^* passes all r rounds, its output has exactly the same distribution as V_1^*'s view.*

When $(s-t)r = O(\log(n))$, the success probability of S^* is $1/poly(n)$, thus repeating it $poly(n)$ times gives a simulator which outputs "Fail" only with probability no more than 1/2. This upper bound for $(s-t)r$ is necessary to let the success probability of S^* be non-negligible. On the other hand, note that to make the soundness error smaller than 1, we need $s-t = \Omega(\log(rs)) = \Omega(\log(n))$. These two requirements together force $r = O(1)$.

In the next section, we will show how to increase the success probability of zero-knowledge simulator to relax this tension and handle more rounds.

3 A new transformation: construction and properties

Note that in the simulator, the success probability in each round is 2^{t-s}, which is polynomially small if $s - t = \Theta(\log(n))$ as in the original paper [3]. Actually the success probability is the number of representatives (2^t) divided by the total number of points (2^s). Since each cell only has one representative, the probability is 2^{t-s}. A basic idea for boosting the success probability is to increase the number of representatives. Simply letting the verifier send many random points does not work (not because of the soundness, but because of rewinding for some reason.) However, it turns out that we can use nested Random Selections to achieve any $O(\log(n)/\log\log(n))$ rounds.

Theorem 1 *For any honest verifier PZK protocol with $r = O(\log(n)/\log\log(n))$ rounds, we can transform it to a $3r$-round general PZK protocol.*

Construction

Suppose the original protocol is $(P, V) : (\alpha_1, \beta_1, ..., \alpha_r, \beta_r)$. The new transformation is specified as follows for each round (α_i, β_i), with parameters set as

$$s_1 = s, \quad t_1 = s_1 - 8\log(rs_1), \quad s_2 = s_1 - t_1, \quad t_2 = s_2 - 8\log(rs_2), \quad \epsilon_1 = 1/2. \tag{4}$$

1. V_2 chooses $f_{i1} \in_R \mathcal{F}^{s_1}_{s_1,t_1}$ and sends it to P_2;
2. P_2 chooses a random cell index $y_{i1} \in_R \{0,1\}^{t_1}$ and sends it to V_2;
3. V_2 checks whether $|f_{i1}^{-1}(y_{i1})| \in [2^{s_1-t_1} - \epsilon_1, 2^{s_1-t_1} + \epsilon_1]$. If not, then V_2 accepts and terminate the whole protocol; else V_2 chooses $f_{i2} \in_R \mathcal{F}^{s_2}_{s_2,t_2}$ and sends it to P_2;
4. P_2 chooses a random subcell index $y_{i2} \in_R \{0,1\}^{t_2}$ and sends it to V_2;
5. V_2 chooses a random α_i in the subcell indexed by (y_1, y_2) and sends it to P_2;
6. P_2 checks whether $f(\alpha) = y$ and terminates the protocol if not. Otherwise, P_2 sends β_i in the same way as P does on seeing $\alpha_1, ..., \alpha_i$.

Analysis of the Zero-Knowledge property

As before, since the messages α_i outputted by S is uniformly at random in $\{0,1\}^s$, independent of all previous messages, the simulator S_2^* passes a round i if and only if the α_i is one of the representatives. But now we have $2^{t_1+t_2}$ representatives, thus S_2^* passes the round with probability $2^{-(s_1-t_1-t_2)}$. Thus it passes all r rounds with probability $2^{-(s_1-t_1-t_2)r}$, which is $O(\log n)$ with the current setting of parameters and for $r = O(\log n / \log\log n)$. Also similar as before, conditioned on S_2^* passes all rounds, the output is equally distributed as V_2^*'s view. Thus the ZK property holds.

Analysis of the soundness error

Note that the inner Random Selection produces an α_i deviating from being uniform in $f_{i1}^{-1}(y_{i1})$ by at most $2s_2 2^{-(s_2-t_2)/4} + 2^{-s_2}$. Thus finally the α_i deviates from uniform in $\{0,1\}^s$ by at most

$$2s_1 2^{-(s_1-t_1)/4} + 2^{-s_1} + 2s_2 2^{-(s_2-t_2)/4} + 2^{-s_2} \tag{5}$$

Another part of soundness error probability introduced by this transformation is due to the termination in Step 3. Applying the hashing lemma (Fact 3) with $A = \{0,1\}^s$, we can upper bound this additional error probability by

$$\left(\frac{2s_1}{\epsilon_1 \sqrt{2^{s_1-t_1}}} \right)^{s_1} \tag{6}$$

for each round. Thus the soundness error of (P_1, V_1) is at most

$$\epsilon_0 + r \left[2s_1 2^{-(s_1-t_1)/4} + 2^{-s_1} + 2s_2 2^{-(s_2-t_2)/4} + 2^{-s_2} + \left(\frac{2s_1}{\epsilon_1 \sqrt{2^{s_1-t_1}}} \right)^{s_1} \right] \tag{7}$$

where ϵ_0 is the soundness error probability of the original protocol (P_0, V_0).

By either the padding argument or parallel composition, we can assume that $s \geq n$. For any $r = c\log(n)/\log\log(n)$ where c is a constant, with the current setting of parameters, the soundness error increases by at most $2/rs_1 + r2^{-s} + 2/rs_2 + 1/r^8 s_1^8 + (2/r^4 s_1^3)^{s_1} = o(1)$. This completes the proof.

Similar to the arguments also apply to SZK and CZK, but we will omit the details since it is already known how to deal with polynomial-round protocols in those two cases [7].

4 Concluding discussions

The approach used in this paper seems not enough to break the $\log(n)$ barrier by itself. The reason is that even if one can let $s-t$ be a constant — and thus the probability of the simulator succeeds in each round is a constant p — the probability of all rounds succeed is still p^r, which is negligible if $r = \omega(\log(n))$. So it seems that some new structure is needed, like as done in the work [7], with the further requirement that the new structure does not introduce any further simulation deviation.

References

[1] M. Bellare, S. Micali, and R. Ostrovsky. The (true) complexity of statistical zero knowledge. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing (STOC)*, pages 494–502, 1990.

[2] I. Damgård. Interactive hashing can simplify zero-knowledge protocol design without computational assumptions (extended abstract). In *Proceedings of the 13th Annual International Cryptology Conference (CRYPTO)*, pages 100–109, 1993.

[3] I. Damgård, O. Goldreich, and A. Wigderson. Hashing functions can simplify zero-knowledge protocol design (too). Technical Report RS-94-39, BRICS, 1994.

[4] O. Goldreich. *Foundations of cryptography*, volume 1. Cambridge University Press, 2001.

[5] O. Goldreich. Zero-knowledge twenty years after its invention. *Electronic Colloquium on Computational Complexity (ECCC)*, (063), 2002.

[6] O. Goldreich, S. Micali, and A. Widgerson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991. Preliminary version in *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, 1986.

[7] O. Goldreich, A. Sahai, and S. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 399–408, 1998.

[8] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18:186–208, 1989. Preliminary version in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC)*, 1985.

[9] R. Ostrovsky, R. Venkatesan, and M. Yung. Interactive hashing simplifies zero-knowledge protocol design. In *Proceedings of the 12th Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT)*, pages 267–273, 1993.