T-79.511 Special Course on Cryptology / Zero Knowledge: Etudé in application of the triangle inequality

Mika Kojo

October 10, 2001

1 Introduction

The paper by M. Ber-Or *et al.* "Everything Provable is Provable in Zero-Knowledge" [1] is the subject of this manuscript. We aim to give overview of the main theorem of the paper, namely, as the title of the paper suggests, **IP** = **ZK**. However, many of the proofs in the paper are based on a very simple idea which we explain before the main theorem. Furthermore, to keep this discussion light we insist on leaving out several fundamental results.

The original paper [1] is perhaps not a very easy reading, although its not hard either. Infact, even that paper lacks many of the building blocks necessary for the main theorem. The results in [2] are necessary, and the understanding of Arthur-Merlin protocols (as stated e.g. [4]) are also needed. We suggests that the reader studies the papers by Kaski [4] and Tarkkala [5]. The former in particular is necessary for good appreciation of the main theorem here, and the latter explains something about its strength.

Acknowledgements. Thanks are due to Helger Lipmaa (for the seminar), Petteri Kaski and Lauri Tarkkala (for their elegant surveys). Furthermore, I thank (ahead of time) Petteri Kaski for his efforts in trying to decipher this manuscript.

2 Preliminaries from topology

This section is about elementary topology, but perhaps explains the title of this manuscript. Suppose (X, d) is a metric topological space, then by definition $d : X \times X \to \mathbb{R}_+$ is the distance function over X.

The triangle inequality for a metric states that

$$d(x, z) \leq d(x, y) + d(y, z),$$

for any points $x, y, z \in X$. Furthermore, if (X, n) is a normed topological space, where $n : X \to \mathbb{R}_+$, we can define

$$d(x, y) = n(x - y).$$

Inparticular over $\mathbb{R}$ we have the standard euclidean metric

$$d(x, y) = |x - y|.$$

Observe further, that if $Y \subset X$ then Y inherits the metric of X. Thus we may consider $[0, 1] \subset \mathbb{R}$ as a metric space.

Given the above consideration we have now defined for $[0, 1]$ a metric which we may write $|x - y|$ for $x, y \in [0, 1]$. Hence,

$$|x - z| \leq |x - y| + |y - z|,$$

holds for all $x, y, z \in [0, 1]$. We will use this later in a particular way that needs some elaboration.

Suppose you have a predicate $P : X \to [0, 1]$ and then $x, z \in X$. Thus we can study the distance

$$|P(x) - P(z)|.$$

Furthermore, by triangle inequality in the space $[0, 1]$, we may take *any* point $y \in X$ and see that

$$|P(x) - P(z)| \leq |P(x) - P(y)| + |P(y) - P(z)|.$$

If we take $z_1, \ldots, z_n \in X$ and write

$$|P(z_1) - P(z_n)| \leq \sum_{i=1,\ldots,n-1} |P(z_i) - P(z_{i+1})|,$$

we see that there exists (at least one) i such that

$$\frac{|P(z_1) - P(z_n)|}{n - 1} \leq |P(z_i) - P(z_{i+1})|.$$

This is important for our applications of triangle inequality towards the proof of the main theorem.

More generally we have the following lemma, which is useful in the context of triangle inequality.

Lemma 2.1 *Let $a \in \mathbb{R}$, $b : I \to \mathbb{R}$ and $\lambda : I \to \mathbb{R}_+$. Suppose*

$$a \leq \sum_{i \in I} \lambda(i) b(i),$$

then there exists $i \in I$ such that

$$b(i) \geq \frac{a}{\sum_{i \in I} \lambda(i)}.$$

Armed with this understanding we can now turn to the proof of the main theorem. The discussion of this section is not explicitly used in the following.

3 Main theorem

The purpose of this section is to prove the following theorem.

Theorem 3.1 *Given secure probabilistic encryption scheme every language in* **IP** *is also in* **ZK**. *Namely,* **IP** = **ZK**.

We will walk through the proof without formally defining everything. Reader should consult the mentioned references for formal definitions. However, as the notion of secure probabilistic encryption schemes is central to the theorem we need to give a short definition for it.

Definition 3.2 Let $E : \{0,1\} \times \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell(n)}$ be function with ℓ being a polynomial (we use just one polynomial for simplicity). If E is computable in polynomial time, $E(x,y) = E(x',y') \implies x = x'$ and

$$|\mathbf{Pr}[C_n(v) = 1 : v = E(1,y), y \in_U \{0,1\}^*] - \mathbf{Pr}[C_n(v) = 1 : v = E(0,y), y \in_U \{0,1\}^*]| < \frac{1}{q(n)},$$

for any family $C = \{C_n\}$ of polynomial time algorithms, where $C_n : \{0,1\}^{\ell(n)} \to \{0,1\}$, and q is a polynomial, then we say that E is a secure probabilistic encryption scheme.

This definition leads immediately to the following lemma.

Lemma 3.3 *A secure probabilistic encryption scheme E can be extended to $E' : \{0,1\}^{\ell(n)} \times \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell(n)^2}$ which is also a secure probabilistic encryption scheme.*

Proof. As this proof is a trivial application of the triangle inequality we give it in sufficient detail. Let us suppose there exists a family of algorithms $C = \{C_n\}$ and $\{z_n\}$ a collection of strings, with $|y_n| = \ell(n)$, such that

$$|\mathbf{Pr}[C_n(v) = 1 : v = E(z_n, y), y \in_U \{0,1\}^*] - \mathbf{Pr}[C_n(v) = 1 : v = E(0^{\ell(n)}, y), y \in_U \{0,1\}^*]| \geq \frac{1}{q(n)}.$$

We construct $z_n^i = z_n[1, \ldots, i]0^{\ell(n)-i}$ from the bits of z_n and $0^{\ell(n)}$. Hence, $z_n^0 = 0^{\ell(n)}$ and $z_n^{\ell(n)} = z_n$. This gives us a simple example of application of the triangle inequality and leads to

$$|\mathbf{Pr}[C_n(v) = 1 : v = E(z_n^i, y), y \in_U \{0,1\}^*] - \mathbf{Pr}[C_n(v) = 1 : v = E(z_n^{i+1}, y), y \in_U \{0,1\}^*]| \geq \frac{1}{\ell(n)q(n)}.$$

So we see that z_n^i and z_n^{i+1} disagree only at the $i+1$'th bit. It is hence natural to construct a family of algorithms $C' = \{C'_n\}$ such that have agreements "hardwired" and then they utilize the algorithms C_n to check the $i+1$'th place. Hence, we have constructed a distinguisher for the one-bit case which is a contradiction. ■

The above proof is a model for the proof of the main theorem, and hence reader may find it useful preliminary step. At any rate, when we speak about (secure) probabilistic encryption schemes we mean schemes that have this "widened" message domain. The above lemma states that we can do that from any one-bit probabilistic encryption scheme.

Let us continue to prove the main theorem. Reader is again adviced to recall the definitions from [4], as they are necessary for the following theorem.

Theorem 3.4 *Given an interactive proof system (V, P) we may construct an Arthur-Merlin protocol (A, M) such that it is* one-sided*, M is deterministic and (A, M) has error probability less than $1/5$. Furthermore, (A, M) can be assumed to take $r(n)$ rounds and each message may be assumed to be of length $\ell(n)$, where the common input x has length n and r, ℓ are polynomials.*

We do not give a proof for the theorem, as most of the details are given in [4]. However, it is necessary to understand what it means for an interactive proof system to be one-sided. Exactly, the verifier V accepts, if it accepts, independent of its random string. Namely, the predicate for acceptance for V is deterministic polynomial time (as V is polynomial time) algorithm. This is vital for the proof of the main theorem.

However, as the above theorem states these are not restrictions. There exists a set of transformations such that the conditions become satisfied.

Indeed, we may thus assume that we are given an Arthur-Merlin protocol (A, M). Arthur, that is A, uses $\ell(n)r(n)$ bits of his random string, denoted by $x_1, \ldots, x_{r(n)}$, which he just sends to Merlin, that is M. Merlin responds with messages $y_1, \ldots, y_{r(n)}$. Furthermore, the predicate indicate above is denoted by

$$\mathrm{accept}(x, x_1, \ldots, x_{r(n)}, y_1, \ldots, y_{y(n)}).$$

So Arthur accepts by evaluating this predicate.

We may suppose that (A, M) is our given interactive proof system and, by slight abuse of notation, we may attempt to build an equivalent interactive proofs system (V, P), where P is zero-knowledge. To do so we simply wrap, or compile, (A, M) into (V, P).

However, one may ask; what is zero-knowledge? This will become quite clear during this proof[1], however, let us explain it intuitively at this stage. Namely, (V, P) is zero-knowledge interactive proof system if the distribution of the tuples $(x, x_1, \ldots, x_{r(n)}, y_1, \ldots, y_{r(n)})$ (that is, public histories of the protocol execution) is computationally indistinguishable from a distribution that V alone could produce.

[1] Unfortunately, from this proof it will *not* become clear that P can prove anything to V, only that V cannot find anything else but what P wants to prove.

Let us describe the transformation from (A, M) to (V, P). Let $L \subset U$ be the language which (A, M) attempts to prove. Let $x \in U$ be a common input. Now consider what P does; if $x \notin L$ then P responds always with #, say. If $x \in L$ and P receives a message x_i from V then P responds with # in case that x_i was invalid (in some sense, e.g. $|x_i| \neq \ell(n)$) or P responds with $y_i = E_n(z_i, d_i)$ otherwise. Here $z_i = M(x, x_1, \ldots, x_i)$ and $d_i \in_U \{0,1\}^{\ell(n)}$. This continues for exactly $r(n)$ rounds, and thus P always responds with either y_i or #.

The same $r(n)$ first rounds the verifier V acts exactly as A. Namely, V sends to P the successive random bits from her random string. In total V uses the $r(n)\ell(n)$ random bits in this phase.

One should observe that all of this is possible, because V is not required to understand the messages y_i. Thus the operation can continue for the $r(n)$ steps even if V obtains practically random messages from P.

When the $r(n)$ steps are exhausted then P must convince V that with large probability V would have accepted if the y_i's would not have been encrypted. Namely, P wishes to convince V of the validity of the assertion

$$\begin{aligned}&\exists d_1, \ldots, d_{r(n)}, z_1, \ldots, z_{r(n)}\\ &\quad [(\forall i : E_n(z_i, d_i) = y_i) \text{ and}\\ &\quad \mathrm{accept}(x, x_1, \ldots, x_{r(n)}, y_1, \ldots, y_{r(n)}) = 1].\end{aligned}$$

It is important that this statement is in **NP** and thus one may attempt to use "weaker" zero-knowledge proof system to prove it. Indeed, we can use the following results.

Theorem 3.5 *Given secure probabilistic encryption scheme graph 3-coloring admits a zero-knowledge proof. Hence, every language in* **NP** *admits a zero-knowledge proof.*

This theorem is quite hard to prove (mainly the zero-knowledge part, as in our main theorem), but the protocol is in itself simple (see [2]). Infact, O. Goldreich [3] states

> "... We believe that this [the main theorem wrt. zero-knowledge for languages in **NP**] extension does not have much practical significance."

Shortly; let G be a graph, and P wishes to prove to V that she knows the 3-coloring of this graph (and that it exists). So P knows a labeling map $\lambda : G \to \{0,1,2\}$ that satisfies, if $(a \to b) \in G$ then $\lambda(a) \neq \lambda(b)$. Each round i of the interactive proof system (V,P) the prover P sends an encrypted graph $G' = E_n(\pi(\lambda(G)), d_i)$ to the verifier, where $\pi : \{0,1,2\} \to \{0,1,2\}$ is a permutation, and verifier is allowed to ask whether an edge $(a' \to b') \in G'$ is properly colorable. This query is responded by prover by revealing the appropriate $(\pi(\lambda(a)), \pi(\lambda(b)), d_{i,a}, d_{i,b})$ to the verifier. These steps may be repeated (as a distinct round) until verifier is satisfied. See [2] and [3] for more details.

Instead of proceeding further with the 3-coloring or languages in **NP**, we continue with our proof of the main theorem.

There is now a gap in our construction. Namely, we first run a interactive proof system for $r(n)$ rounds and then use another for the rest, but finite number of rounds. The problem is that even if both parts would be zero-knowledge in themselves, is the composition of these protocols zero-knowledge? The following theorem states that indeed this holds.

Theorem 3.6 *Let us have an interactive proof system $(V_1,P_1);(V_2,P_2)$ for L where P_1 and P_2 are zero-knowledge. Here P_1 is zero-knowledge over L and if P_2 is zero-knowledge over $L' = (V_1,P_1)(L)$, which is the set of all histories for the protocol (V_1,P_1) for all $x \in L$, then $P_1;P_2$ is zero-knowledge over L.*

We note that $(V,P)(x)$ is just the set of all histories produced by the protocol (V,P) given the input x.

Proof of this theorem may be found in [1]. As the proof is just an application of triangle inequality along the lines of what we will see below it is not given here. Indeed, the idea is to observe that any algorithms that produces distributions similar to (V_1,P_1) and (V_2,P_2) may be composed as well. This provides a triangle inequality argument which leads to a reduction from P to P_1 (or to P_2).

The above theorem thus suggests that it suffices to give a zero-knowledge proof for our construction (V,P) for the first $r(n)$ rounds. We can then apply the composition theorem and get a proof for the full protocol. Namely, we assume that the zero-knowledge system for the langauge in **NP** is a black box satisfying the above conditions. Furthermore, we require this black box zero-knowledge protocol accepts in finite time with error probability less than 1/5. We also require it to be one-sided.

It is now possible to prove that (V,P) is an interactive proof system (complete and sound). As we have one-sidedness throughout it implies that if A would accept then V would accept. Thus (V,P) is complete interactive proof system.

If P' is a corrupted prover then we need to show that V accepts with small error probability (indeed, with an error probability that can be made negligible). We see that by probability at most $1/5$ V would accept when A would not over unencrypted messages. However, we know that A has error probability at most $1/5$ and thus V accepts with error probability at most $2/5$. Hence, (V,P) is a sound interactive proof system.

We are now left with the question whether (V,P) is zero-knowledge interactive proof system. To be zero-knowledge is clearly property of the prover. As noted before we need to prove this only for the first $r(n)$ rounds after which the full proof comes from the composition theorem.

Again we attempt to apply the triangle inequality. This time the goal is to reduce the notion of zero-knowledge to that of the security of the secure probabilistic encryption scheme.

Let $\bar{P}_j$ be a prover algorithm that for $r(n)$ rounds sends $\bar{y}_i = E_n(z_i, d_i)$, where $z_i = M(x, x_1, \ldots, x_i)$ if $i \leq j$ and $z_i = 0^{\ell(n)}$ otherwise. For rounds beyond $r(n)$ then prover $\bar{P}_j$ responds with $\#$, say. Observe that $\bar{P}_{r(n)}$ is P restricted to $r(n)$ rounds, which we wish to prove to be zero-knowledge, and $\bar{P}_1$ is the prover that produces random output. We wish to use this to reduce down to the probabilistic encryption scheme using triangle inequality. Reader may already see how this is going to happen.

For simplifying some notation let $S_j = (\bar{V}, \bar{P}_j)(x)$ be a set of views for some fixed polynomial time algorithm $\bar{V}$. We assume that $\bar{V}$ tosses $t(n)$ coins for each round. Furthermore, we always take $|x| = n$. Also, let $C = \{C_n\}$ be a polynomial sized family of algorithms (with $C_n : \{0,1\}^{\omega(n)} \to \{0,1\}$, for some polynomial ω).

To say that $\bar{P}_{r(n)}$ is *not* zero-knowledge can thus be formalized as

$$|\mathbf{Pr}[C_n(v) = 1 : v \in_U S_1(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_{r(n)}(x)]| \geq \frac{1}{h(n)},$$

for some polynomial h. Now applying the triangle inequality we get

$$\epsilon_n = |\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_{i+1}(x)]| \geq \frac{1}{r(n)h(n)}.$$

Our goal is to construct a family of algorithm $C' = \{C'_n\}$ such that

$$|\mathbf{Pr}[C'_n(v) = 1 : v = E_n(0^{\ell(n)}, d), d \in_U \{0,1\}^{\ell(n)}] - \mathbf{Pr}[C'_n(v) = 1 : v = E_n(y, d), d \in_U \{0,1\}^{\ell(n)}]| \geq \epsilon_n,$$

for some $y \in \{0,1\}^{\ell(n)}$, say. That is, we would then contradict the security of the probabilistic encryption scheme E_n that we assume.

We consider the first $i+1$ rounds and set $f(n) = (i+1)t(n)$. Thus $\bar{V}$ uses $f(n)$ randoms bits from its given random string. Infact, the $f(n)$ bit prefix σ of $\bar{V}$'s random string and the messages $y_1, \ldots, y_i$ determine; $\bar{V}$'s messages $x_1, \ldots, x_{i+1}$, $\bar{V}$'s state after $i+1$ rounds, the string z_{i+1} that the prover will encrypt next. This comes from the requirement that $\bar{P}_j$ are deterministic (which comes from the fact that Merlin, M, is deterministic).

Suppose now that $\bar{V}$'s random string is prefixed with σ. Let us write

$$\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i(x)] = \sum_{\sigma \in \{0,1\}^{f(n)}} 2^{-f(n)} \mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^\sigma(x)],$$

which defines S_i^σ as the set of histories where $\bar{V}$ has prefix σ for its random string. We may thus write due to triangle inequality

$$\epsilon_n = |\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_{i+1}(x)]| \leq \sum_{\sigma \in \{0,1\}^{f(n)}} 2^{-f(n)} |\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^\sigma(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_{i+1}^\sigma(x)]|,$$

which implies that there exists σ such that

$$|\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^\sigma(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_{i+1}^\sigma(x)]| \geq \epsilon_n.$$

We fix such a σ and do the same for $y = y_1, \ldots, y_i$. Namely, write $g(n) = |y| = i\ell(n)q(n)$, where $q(n)$ is the size of the output of E_n. Let $\lambda(y)$ be the probability that the provers first i messages are $y_1, \ldots, y_i$. Now we may write

$$\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^\sigma(x)] = \sum_{y \in \{0,1\}^{g(n)}} \lambda(y) \mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^{\sigma,y}(x)],$$

where $S_i^{\sigma,y}$ is the set of histories with σ being the prefix of $\bar{V}$'s random string and y the first i messages send by $\bar{P}_i$. Indeed, we get

$$\epsilon_n \leq |\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^\sigma(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^\sigma(x)]| \leq \sum_{y \in \{0,1\}^{g(n)}} \lambda(y) |\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^{\sigma,y}(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_{i+1}^{\sigma,y}(x)]|,$$

which in turn leads to (by triangle inequality as has become a habbit for us)

$$|\mathbf{Pr}[C_n(v) = 1 : v \in_U S_i^{\sigma,y}(x)] - \mathbf{Pr}[C_n(v) = 1 : v \in_U S_{i+1}^{\sigma,y}(x)]| \geq \epsilon_n,$$

for some y.

From our discussion before it is now clear that given σ and y we have determined $x_1, \ldots, x_{i+1}$ and z_i such that $E_n(z_{i+1}, d_{i+1})$ or $E_n(0^{\ell(n)}, d_{i+1})$ is send form $\bar{P}_i$ to $\bar{V}$ and if one then continues as

$(\bar{P}_1, \bar{V})$ for the rest $r(n) - i - 1$ rounds, one has a distinguisher for the produced histories. Let us use this to distinguish the probabilistic encryption scheme.

Let $C' = \{C'_n\}$ be a family of algorithm which is constructed as follows. We "hardwire" into C'_n the fixed values σ and y that were deduced by the above discussion. The algorithm C'_n takes $\bar{V}$ and runs it until one exhausts σ, that is, for the first $i + 1$ rounds. The messages $x_1, \ldots, x_{i+1}$ can be recorded during this procedure. The algorithm C'_n uses its argument y_{i+1} then o continue the process for one step. For the rest of the rounds it uses the algorithm $\bar{P}_1$ to interact with $\bar{V}$. After all this C'_n calls C_n with the constructed history and outputs whatever C_n outputs.

The constructed algorithm C'_n runs in time polynomial as all the lengths are polynomially bounded. Naturally, we also have

$$\begin{aligned} &|\mathbf{Pr}[C'_n(v) = 1 : v = E_n(0^{\ell(n)}, d), d \in_U \{0,1\}^{\ell n}] - \\ &\mathbf{Pr}[C'_n(v) = 1 : v = E_n(x, d), d \in_U \{0,1\}^{\ell n}]| \\ &\geq \epsilon_n, \end{aligned}$$

for any $x \in \{0,1\}^{\ell(n)}$. As $\epsilon_n \geq \dfrac{1}{r(n)h(n)}$ we thus have a distinguisher for E_n. This is a contradiction to our hypothesis.

Indeed, we have thus shown that (V, P), as here constructed, is a zero-knowledge interactive proof system assuming that E_n is a family of secure probabilistic encryption schemes.

References

[1] Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Hstad, Joe Kilian, Silvio Micali and Phillip Rogaway, Everything Provable is Provable in Zero-Knowledge.

[2] Oded Goldreich, Silvio Micali and Avi Wigderson, How to Prove All NP Statements in Zero-Knowledge and a Methodology of Cryptograhic Protocol design (Extended Abstract).

[3] Oded Goldreich, *Foundations of Cryptography: Fragments of a book*, Chapter 6, Zero-knowledge proof systems. Weizmann Institute o Science, February 1995.

[4] Petteri Kaski, T-79.511 Special Course on Cryptology / Zero Knowledge: Rudiments of Interactive Proof Systems, September 26, 2001.

[5] Lauri Tarkkala, T-79.511 Special Course on Cryptology / Zero Knowledge: IP=PSPACE, October 1, 2001.