A Survey of Zero-Knowledge Proofs with Applications to Cryptography

Austin Mohr
Southern Illinois University at Carbondale
Carbondale, IL 62901
E-mail: austinmohr@gmail.com

Abstract

Zero-knowledge proofs are proofs that show a statement to be true without revealing anything other than the veracity of the statement to be proven. After a formal definition of zero-knowledge proof schemes and a simple example, zero-knowledge proofs for Graph Isomorphism and Graph 3-colorability are presented, the latter being the basis for the proof that all languages in NP have zero-knowledge proofs. The paper concludes with an application of zero-knowledge proofs in cryptography, the Fiat-Shamir Indentification Protocol, which is the basis for modern zero-knowledge entity authentication schemes.

1 Introduction

A zero-knowledge proof (ZKP) is a proof of some statement which reveals nothing other than the veracity of the statement. The word "proof" here is not used in the traditional mathematical since (e.g. one may prove the Pythagorean Theorem). Rather, a "proof", or equivalently a "proof system", is a randomized protocol by which one party (called the *prover*) wishes to convince another party (called the *verifier*) that a given statement is true.

Definition 1.1 (Interactive Proof systems and the class $\mathcal{IP}$ [2])**.** An interactive proof system for a set S is a two-party game between a *verifier* executing a probabilistic polynomial-time strategy and a *prover* which executes a computationally unbounded strategy satisfying:

- *Completeness*: For every $x \in S$, the verifier always accepts after interacting with the prover on common input x.
- *Soundness*: For some polynomial p, it holds that for every $x \notin S$ and every potential strategy P^*, the verifier rejects with probability at least $\frac{1}{p(|x|)}$ after interacting with P^* on common input x.

The class of problems having interactive proof systems is denoted $\mathcal{IP}$.

In other words, a proof is complete if an honest verifier will always be convinced of a true statement by an honest prover. A proof is sound if a cheating prover can convince an honest verifier that some false statement is actually true with only a small probability. A proof is further considered to be zero-knowledge if it satisfies the following definition.

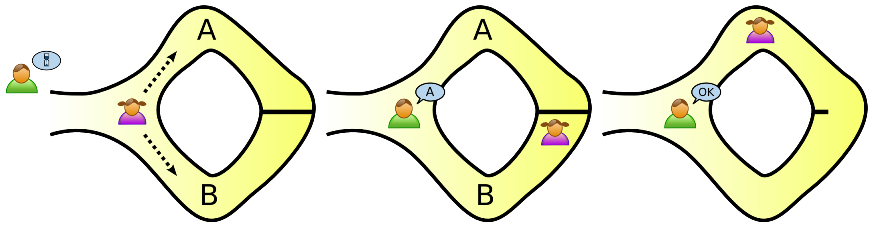
Definition 1.2 (Zero-knowledge [2])**.** A strategy A is zero-knowledge on (inputs from) the set S if, for every feasible strategy B^* there exists a feasible computation C^* so that the following two probability ensembles are computationally indistinguishable:

- the output of B^* after interacting with A on common input $x \in S$
- the output of C^* on input $x \in S$

In the previous definition, the first ensemble represents the output of an actual execution of the proof system protocol, while the second ensemble (called the "simulation") is the output of a stand-alone procedure which is not a part of any interactive system. A proof is called zero-knowledge if the output of any strategy B^* used by a cheating verifier could also be produced by the non-interactive computation C^*. In other words, whatever information can be learned by interacting with A on some input x can also be extracted from x *without* interacting with A.

Remark 1.3. ZKPs exist *only* if one-way functions exist, as a cheating verifier may be able to extract additional information after interacting with a prover by essentially "hacking" the prover. The notion of "one-way functions" is often generalized to represent any means of "commiting to a secret bit" (i.e., information hiding).

2 A Very Simple ZKP



A magical cave [7]

(The following example was taken from [7], which was itself adapted from [5].) Peggy has found a magical cave! The cave has a magic door deep inside it that opens only upon uttering the secret word, a secret which Peggy has uncovered. Victor hears about this and wishes to also know the secret. Peggy agrees to sell Victor the secret word for \$1,000,000, but Victor wants to be certain that Peggy, indeed, knows the secret word before he pays. How can Peggy (the prover) prove to Victor (the verifier) that she knows the secret without actually conveying the secret to Victor?

Peggy and Victor devise the following scheme. Victor will wait outside the cave while Peggy enters. She chooses either path A or path B at random. Victor does not know which path she has chosen. Then, Victor will enter the cave as far as the fork and announce the path along which he wants Peggy to return.

Suppose Peggy knows the secret word. Then, she will be able to return along either path A or path B regardless of the path she chose initially. If Victor announces the same path through which Peggy chose to enter, she simply exits the cave along that path. If Victor announces the path that Peggy did not choose, she whispers the secret word (Victor is presumably too far away to hear it), thus opening the door and allowing her to return along the desired path.

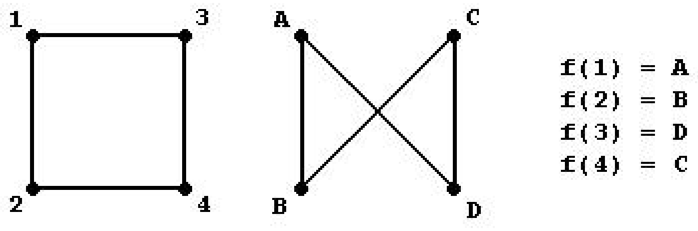
Suppose Peggy does not know the secret word. Then, she will only be able to return along the appropriate path if Victor announces the same path that she chose. This will occur with probability $\frac{1}{2}$. If Peggy is tested many times, the probability that the path announced by Victor is the same chosen by Peggy for every test becomes negligible. That is, Victor will eventually discover that Peggy is a liar.

One might wonder why Victor simply does not tell Peggy to enter through a known path (say, path A) and then require her to return along the other path. Clearly, this would force Peggy to use her knowledge of the secret word to return appropriately. However, such a scheme also allows Victor to eavesdrop by following her down the prespecified path. By randomizing the initial path, the probability that Victor can successfully eavesdrop is reduced.

3 Zero-Knowledge Proof for Graph Isomorphism

Definition 3.1 (Graph Isomorphism and the language GI [1]). Two graphs $G(V, E)$ and $H(V, F)$ are isomorphic if and only if there exists a permutation $\pi \in S_{|V|}$ (the symmetric group of $|V|$ elements) such that $(u, v) \in E$ iff $(\pi(u), \pi(v)) \in F$. We then write $H = \pi G$. We say that the graph $H(V, F)$ is a random isomorphic copy of the graph $G(V, E)$ if H is obtained from G by picking $\pi \in_R S_{|V|}$ and letting $H = \pi G$.

The language GI (graph isomorphism) consists of all the pairs of isomorphic graphs (i.e. $GI = \{(G, H) : \exists \pi \text{ such that } H = \pi G\}$).



A pair of isomorphic graphs with isomorphism f

Protocol 3.2. [A zero-knowledge proof system for GI [1]]
Common Input: Two graphs $G_0(V, E_0)$ and $G_1(V, E_1)$

Let ϕ denote the isomorphism between G_0 and G_1 (i.e., $G_1 = \phi G_0$). The following four steps are executed m times, each time using independent random coin tosses.

(P1) The prover generates a graph H, which is a random isomorphic copy of G_1. This is done by selecting a permutation $\pi \in_R S_{|V|}$, and computing $H = \pi G_1$. The prover sends the graph H to the verifier.

(V1) The verifier chooses at random $\alpha \in_R \{0, 1\}$, and sends α to the prover. (Intuitively, the verifier asks the prover to show him that H and G_α are indeed isomorphic.)

(P2) If $\alpha = 1$, then the prover sends π to the verifier, else (i.e., $\alpha \neq 1$) the prover sends $\pi \cdot \phi$. (Note that the case $\alpha \notin \{0, 1\}$ is handled as $\alpha = 0$.)

(V2) If the permutation (ψ) received from the prover is not an isomorphism between G_α and H (i.e., $H \neq \psi G_\alpha$), then the verifier stops and *rejects*; otherwise, he continues.

If the verifier has completed m iterations of the above steps, then he *accepts*.

Protocol 3.2 works because, if the prover indeed possesses an isomorphism ϕ such that $G_1 = \phi G_0$, then any permutation $\pi \cdot \phi$ has the property that $H = \pi G_1 = \pi \cdot \phi G_0$. Since π is chosen randomly, no knowledge of ϕ is revealed to the verifier. The verifier asks for an isomorphism between H and one of G_0 and G_1 randomly because only a prover with knowledge of ϕ can prove upon request that either of (H, G_0) and (H, G_1) belong to GI. The validity of Protocol 3.2 is made more precise in the following proposition.

Proposition 3.3. Protocol 3.2 constitutes a zero-knowledge proof for GI.

Proof. We show that Protocol 3.2 upholds the properties of completeness, soundness, and zero-knowledge.

Completeness: Let $(G_0, G_1) \in$ GI. Then the random isomorphic copy H of G_1 will always be isomorphic to both G_0 and G_1. Therefore, an honest verifier will complete all m iterations and *accept* with probability 1.

Soundness: Let $(G_0, G_1) \notin$ GI. Then the random isomorphic copy H of G_1 will be isomorphic to only one of G_0 or G_1. (Note that in the protocol as written, the prover always sets $H = \pi G_1$. A cheating prover, however, could possibly set $H = \pi G_0$.) Therefore, an honest verifier will *reject* with probability $\frac{1}{2}$ in each round.

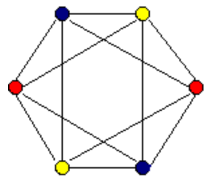
Zero-knowledge: The only information revealed in each round is either π or $\pi \cdot \phi$ where $\pi \in_R S_{|V|}$. Due to the random selection of π, a simulator which simply computes a random isomorphic copy of G_0, G_1, or both is computationally indistinguishable from interaction with the prover. $\square$

Remark 3.4. In the proof of zero-knowledge in [1], the author points out that a cheating verifier may attempt to select a particular sequence of α in order to extract some additional information about the graphs. The author shows (in painful detail) that Protocol 3.2 is resistant to such a strategy. This portion of the proof is exceptionally lengthy and will be omitted.

4 Zero-Knowledge Proofs for All Languages in NP

As shown in [1], there exists a ZKP for any language in NP. The proof of this claim begins by first explicitly constructing a ZKP for the graph three-colorability problem (G3C). Using this scheme and the fact that G3C is NP-Complete [3], the proof that ZKPs exist for all problems in NP follows.

Definition 4.1 (Graph Three-Colorability and the language G3C [1])**.** A graph $G(V, E)$ is said to be <u>3-colorable</u> if there exists a mapping $\phi : V \to \{1, 2, 3\}$ (called a <u>proper coloring</u>) such that every two adjacent vertices are assigned different colors (i.e., each $(u, v) \in E$ satisfies $\phi(u) \neq \phi(v)$). Such 3-coloring induces a partition of the vertex set of the graph to three independent sets. The language <u>graph 3-colorability</u>, denoted G3C, consists of the set of undirected graphs that are 3-colorable.



A properly 3-colored graph [6]

The following "physical" zero-knowledge protocol for G3C uses as its bit-commitment scheme a set of n lockable boxes whose keys are all distinct.

Protocol 4.2. [A zero-knowledge proof system for G3C [1]]
Common Input: A graph $G(V, E)$, $|V| = n$, $|E| = m$

The following four steps are executed m^2 times, each time using independent coin tosses.

(P1) The prover chooses at random an assignment of three colors to the three independent sets induced by (the proper 3-coloring) ϕ, colors the graph using this 3-coloring, and places these colors in n locked boxes each bearing the number of the corresponding vertex. More specifically, the prover chooses a permutation $\pi \in_R S_3$ (the symmetric group of 3 elements), places $\pi(\phi(i))$ in a box marked i ($\forall i \in V$), locks all boxes and sends them (wihtout the keys) to the verifier.

(V1) The verifier chooses at random an edge $e \in_R E$ and sends it to the prover. (Intuitively, the verifier asks to examine the colors of the endpoints of $e \in E$.)

(P2) If $e = (u, v) \in E$, then the prover reveals the colors of u and v (to the verifier) by sending the keys to boxes u and v. Otherwise, the prover does nothing.

(V2) The verifier opens boxes u and v using the keys received and checks whether they contain two different elements of $\{1, 2, 3\}$. If the keys do not match the boxes, or the contents violates the condition, then the verifier *rejects* and stops. Otherwise, the verifier continues to the next iteration.

If the verifier has completed all m^2 iterations, then it *accepts*.

Proposition 4.3. Protocol 4.2 constitutes a zero-knowledge proof for G3C.

Proof. We show that Protocol 4.2 upholds the properties of completeness, soundness, and zero-knowledge.

Completeness: Let $G \in$ G3C. Then any pair of boxes u and v corresponding to some edge of G will certainly be colored differently. Therefore, an honest verifier will complete all m^2 iterations and *accept* with probability 1.

Soundness: Let $G \notin$ G3C. Then at least 1 of the m edges of G is not properly colored. Therefore, an honest verifier will *reject* with probability at least $\frac{1}{m}$ in each round.

Zero-knowledge: The only information revealed in each round is $\pi(\phi(u))$ and $\pi(\phi(v))$ where $(u, v) = e \in_R E$, $\pi \in_R S_3$. Due to the random selection of π, a simulator which simply outputs a pair of integers (i, j), $i, j \in_R \{1, 2, 3\}$ with $i \neq j$, is computationaly indistinguishable from interaction with the prover. $\square$

Remark 4.4. If the permutation π is not applied independently and randomly at each round, then the verifier may be able to learn the values of

$\phi(u)$ and $\phi(v)$ for the selection of vertices u and v in each round. This not only provides the verifier with additional information about ϕ (and thereby violates the condition of zero-knowledge), but, on an edge-dense graph where it may be the case that $m \gg n$, it is possible that m^2 rounds is sufficient for the verifier to have learned all of ϕ!

Theorem 4.5. *[Zero-knowledge proof for all languages in NP [1]] If there exists a nonuniformly secure encryption function, then every language in NP has a zero-knowledge interactive proof system.*

Proof. Let $L \in$ NP, and t be the polynomial-time computable and invertible reduction of L to 3-Colorability (G3C). Namely, t is the composition of the standard reduction of L to 3SAT and the standard reduction of 3SAT to G3C. Recall that $x \in L$ iff $t(x)$ is 3-colorable. A zero-knowledge interactive proof system for L proceeds as follows:

On common input x, each party computes $G \leftarrow t(x)$. The prover uses an (arbitrary) zero-knowledge interactive proof system to prove that G is 3-colorable. The verifier acts according to the result of this subprotocol.

By Proposition 4.3, the above protocol constitutes an interactive proof system for L. To see that the protocol is indeed zero-knowledge, one should note that t is polynomial-time invertible (i.e., there exists a polynomial-time algorithm t^{-1} such that $t^{-1}(t(x)) = x$). $\square$

Remark 4.6. The proof found in [1] further explains why the invertibility of t implies that the protocol Theorem 4.5 is zero-knowledge, but such a proof is beyond the scope of this paper. As a sketch, the verifier V will receive input $t(x)$ and auxilliary input x. However, because of the auxilliary input, V is not of a form to which the standard definition of zero-knowledge can be applied. Instead, another verifier V^* is employed that first computes x from $t(x)$ (which can be done since t is polynomial-time invertible) and then applies V with x as input. From here, the definition of zero-knowledge can be proven first for V^* and then separately for V.

5 Fiat-Shamir Identification Protocol

In cryptogrpahy, ZKPs are primarily used as a means of entity authentication. That is, Peggy possesses some secret S that only she can know. She proves to Victor that she is indeed Peggy (and not an impostor) by proving that she possesses S. Of course, she wants to do so without revealing S to Victor (or any potential eavesdroppers).

The Fiat-Shamir identification protocol, while itself not usually implemented in modern systems, is the basis of zero-knowledge identification protocols currently in use, such as Feige-Fiat-Shamir and Guillou-Quisquater. As such, it serves to illustrate the properties which are important in more sophisticated schemes.

Protocol 5.1 (Fiat-Shamir Identification Protocol [4]).
Initialization

1. A trusted center T selects and publishes an RSA-like modulus $n = pq$ but keeps the primes p and q secret.

2. The prover selects a secret s coprime to n, $1 \leq s \leq n-1$, computes $v = s^2 \bmod n$, and registers v with T as her public key.

Identification Protocol
The following steps are executed t times, each time using independent random coin tosses.

(P1) The prover choses a random r, $1 \leq r \leq n-1$ and sends $x = r^2 \bmod n$ to the verifier.

(V1) The verifier randomly selects a bit $e \in \{0, 1\}$ and sends e to the prover.

(P2) The prover computes and sends to the verifier y, where $y = r$ (if $e = 0$) or $y = rs \bmod n$ (if $e = 1$).

(V2) The verifier *rejects* if $y = 0$ or if $y^2 \not\equiv x \cdot v^e \pmod{n}$. (Depending on e, $y^2 = x$ or $y^2 = xv \bmod n$, since $v = s^2 \bmod n$. Note that checking for $y = 0$ precludes the case $r = 0$.)

If the verifier has completed all t iterations of the above steps, then he *accepts*.

Proposition 5.2. The Fiat-Shamir Identification Protocol constitutes a zero-knowledge entity authentication protocol.

Proof. We show that the Fiat-Shamir Identification Protocol upholds the properties of completeness, soundness, and zero-knowledge.

Completeness: Suppose the prover possesses the secret s. Then she can always correctly provide the verifier with $y = r$ or $y = rs$ upon request. Therefore, an honest verifier will complete all t iterations and *accept* with probability 1.

Soundness: Suppose the prover does not possess the secret s. Then, during any given round, she can provide only one of $y = r$ or $y = rs$ (see remark). Therefore, an honest verifier will reject with probability $\frac{1}{2}$ in each round (which implies an overall probability of 2^{-t} that a cheating prover will *not* be caught).

Zero-knowledge: The only information revealed in each round is $x = r^2 \bmod n$ (in step P1) and either $y = r$ or $y = rs$ (in step P2). Such pairs (x, y) could be simulated by choosing y randomly, then defining $x = y^2$ or $x = \frac{y^2}{v}$. Such pairs, while not generated in the same way as in the protocol, are computationally indistinguishable from them. □

Remark 5.3. As in the cave story, one might wonder why Victor doesn't always request $y = rs$, thus forcing Peggy to make use of s. If this is done, Peggy has a cheating strategy. She can select r randomly and then set $x = \frac{r^2}{v}$. When Victor requests $y = rs$, Peggy will send $y = r$. Now, when Victor attempts to verify Peggys response, he will compute:

$$x \cdot v^e \equiv \frac{r^2}{v} \cdot v \equiv r^2 \equiv y^2 \pmod{n}$$

thus fooling Victor into believing she has possession of s. This strategy becomes ineffective if Victor asks either of the two questions at random. Since Victor believes that $x = r^2$ when he in fact has $x = \frac{r^2}{v}$ (due to Peggy's deception), he will expect Peggy to send $y = \sqrt{\frac{r^2}{v}}$, which she will be unable to compute since this requires her to solve a square root (mod n).

Example 5.4 (An example run of the Fiat-Shamir Identification Protocol)**.**

<u>Initialization</u>

1. Let $p = 5$ and $q = 7$. Then, $n = pq = 35$. n is published to a trusted center.

2. Peggy secretly chooses $s = 16$, which is coprime to n. She publishes $v = s^2 \bmod n = 11$ to the trusted center.

<u>Protocol</u>
Suppose Victor is very easy to convince, and hence requires only 2 successful iterations of the protocol in order to *accept*.

(P1_1) Peggy randomly selects $r = 10$. She sends $x = r^2 \bmod n = 10^2 \bmod 35 = 30$ to Victor.

(V1_1) Victor randomly selects $e = 0$ and sends it to Peggy.

($P2_1$) Since $e = 0$, Peggy computes $y = r = 10$ and sends it to Victor.

($V2_1$) Victor verifies that $y^2 = 10^2 \equiv 30 \pmod{35}$.

($P1_2$) Peggy randomly selects $r = 20$. She sends $x = r^2 \bmod n = 20^2 \bmod 35 = 15$ to Victor.

($V1_2$) Victor randomly selects $e = 1$ and sends it to Peggy.

($P2_2$) Since $e = 1$, Peggy computes $y = sr \bmod n = 16 \cdot 20 \bmod 35 = 5$ and sends it to Victor.

($V2_2$) Victor verifies that $y^2 = 25 \equiv 15 \cdot 11 \pmod{35}$.

Peggy has successfully completed $t = 2$ rounds, so Victor *accepts*

6 Conclusion

Zero-knowledge proofs are of considerable theoretical and practical interest to mathematicians and cryptographers alike. ZKPs acheive the seemingly contradictory goals of proving a statement without revealing anything other than the fact that the statement is indeed true. The zero-knowledge proofs presented for Graph Isomorphism and 3-colorability have further implications on complexity theory which are not fully discussed in this paper. The Fiat-Shamir Identification Protocol, while not normally implemented in modern cryptosystems, is the basis of existing zero-knowledge entity authentication schemes and shows that such schemes can actually be used in practice.

7 Acknowledgements

The author extends thanks to Kenny Fong of Southern Illinois University at Carbondale for the introduction to this fascinating area of study.

References

[1] Oded Goldreich, Silvio Micali, Avi Wigderson. Proofs that yield nothing but their validity (http://portal.acm.org/citation.cfm?id=116852). *Journal of the ACM*, volume 38, issue 3, p.690-728. July 1991.

[2] Oded Goldreich. Zero-knowledge twenty years after its invention. Unpublished manuscript. 2002.

[3] Richard M. Karp. Reducibility Among Combinatorial Problems. In Complexity of Computer Computations, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y.. New York: Plenum, p.85-103. 1972.

[4] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography. 5th Ed. CRC Press, 2001.

[5] Jean-Jacques Quisquater, Louis C. Guillou, Thomas A. Berson. How to Explain Zero-Knowledge Protocols to Your Children (http://www.cs.wisc.edu/∼mkowalcz/628.pdf). *Advances in Cryptology - CRYPTO '89: Proceedings*, v.435 p.628-631, 1990.

[6] "Graph coloring." Wikipedia, The Free Encyclopedia. 24 Apr 2007, 05:32 UTC. Wikimedia Foundation, Inc. 24 Apr 2007 (http://en.wikipedia.org/w/index.php?title=Graph_coloring&oldid=125418313).

[7] "Zero-knowledge proof." Wikipedia, The Free Encyclopedia. 12 Mar 2007, 20:19 UTC. Wikimedia Foundation, Inc. 17 Apr 2007 (http://en.wikipedia.org/w/index.php?title=Zero-knowledge_proof&oldid=114623242).