bob

On The Existance of 3-Round Zero-Knowledge Proof Systems

Matthew Lepinski and Silvio Micali

April 20, 2001

Abstract

We provide a proof of knowledge assumption that allows us to construct a three round zero-knowledge proof system for any language in NP.

1 Introduction

Goldwasser, Micali and Rackoff[5] defined a Zero-Knowledge Proof System. Brassard, Bhaum and Crèpeau[3] later defined a Zero-Knowledge argument which differs from a Zero-Knowledge proof in that the prover is assumed to be computationally bounded. Goldreich and Krawczyk[4] proved that any language with a 3-round Black Box Zero-Knowledge proof or argument is in BPP. At the time of Goldreich and Krawczyk's paper all known Zero-Knowledge proofs and arguments achieved Black Box Zero-Knowledge. Hada and Tanaka[6] provided a 3-round Zero-Knowledge argument for every language in NP under a very strong version of the Diffie-Hellman assumption.

We present a different assumption that can be used to prove the existence of 3-round Zero-Knowledge proofs for every language in NP. Our work is based on the concept of an oblivious transfer channel proposed by Micali and Bellare[1].

2 The Assumption

A proof of knowledge similar to the following is a commonly used in Zero-Knowledge Proofs.

- PROVER: Sends (p, g, R, H) to VERIFIER where p is a prime of the form $2q-1$ and q is prime, g is a generator of Z_p^*, R is a random element of Z_p^* and H is a hash function whose range is $\{0,1\}^k$.
- VERIFIER: Selects a random x and y in Z_p^*. Flips a coin. If the coin comes up heads, he chooses the pair $X = (g^x, Rg^y)$ if the coin comes up tails, he chooses the pair $X = (Rg^x, g^y)$.
- VERIFIER: Selects k pairs A_i in the following manner. First select x_i and y_i from Z_p^*, then flip a coin to choose between $A_i = (g^{x_i}, Rg^{y_i})$ and $A_i = (Rg^{x_i}, g^{y_i})$. We say A_i is constructed in the same manner as X if $X = (g^x, Rg^y)$ and $A_i = (g^{x_i}, Rg^{y_i})$ or if $X = (Rg^x, g^y)$ and $X = (Rg^x, g^y)$.
- VERIFIER: Let $b_1 \dots b_k = H(X, A_1, \dots, A_k)$. If $b_i = 0$ then set $B_i = (x_i, y_i)$. If $b_i = 1$ and A_i is constructed in the same manner as X then set $B_i = (x + x_i, y + y_i)$. Otherwise set $B_i = (x + y_i, y + x_i)$.
- VERIFIER: Send $(X, A_1, \dots, A_k, B_1, \dots, B_k)$ to PROVER.
- PROVER: Compute $b_1 \dots b_k = H(X, A_1, \dots, A_k)$. Let $X = (W, Z)$, $A_i = (C_i, D_i)$ and $B_i = (E_i, F_i)$. Accept if for each i either $b_i = 0$ and $A_i = (g^{E_i}, Rg^{F_i})$, $b_i = 0$ and $A_i = (Rg^{E_i}, R^{F_i})$, $b_i = 1$ and $(WC_i, ZD_i) = (Rg^{E_i}, Rg^{F_i})$ or $b_i = 1$ and $(WD_i, ZC_i) = (Rg^{E_i}, Rg^{F_i})$.

Assumption 1 (Proof of Knowledge) *For any polynomial time verifier, V, that outputs $(X, A_1, \dots, A_k, B_1, \dots, B_k)$ such that the prover accepts in the above protocol, there exists a polynomial time verifier, V', who with probability*

$1-\epsilon$ *outputs* $(X, A_1, \ldots, A_k, B_1, \ldots, B_k, x, y)$ *such that* $X = (g^x, Rg^y)$ *or* $X = (Rg^x, g^y)$ *where* ϵ *is a negligible function of* k.

3 The Protocol

Our protocol is based on Blum's protocol[2] for Hamiltonian Path.

- PROVER: Sends (p, g, R, H) to VERIFIER where p is a prime of the form $2q-1$ and q is prime, g is a generator of Z_p^*, r is a random element of Z_p^*, $R = g^r$ and H is a hash function whose range is $\{0,1\}^k$.
- VERIFIER: Selects a random x and y in Z_p^*. Flips a coin. If the coin comes up heads, he chooses the pair $X = (g^x, Rg^y)$ if the coin comes up tails, he chooses the pair $X = (Rg^x, g^y)$.
- VERIFIER: Selects k pairs A_i in the following manner. First select x_i and y_i from Z_p^*, then flip a coin to choose between $A_i = (g^{x_i}, Rg^{y_i})$ and $A_i = (Rg^{x_i}, g^{y_i})$. We say A_i is constructed in the same manner as X if $X = (g^x, Rg^y)$ and $A_i = (g^{x_i}, Rg^{y_i})$ or if $X = (Rg^x, g^y)$ and $X = (Rg^x, g^y)$.
- VERIFIER: Let $b_1 \ldots b_k = H(X, A_1, \ldots, A_k)$. If $b_i = 0$ then set $B_i = (x_i, y_i)$. If $b_i = 1$ and A_i is constructed in the same manner as X then set $B_i = (x + x_i, y + y_i)$. Otherwise set $B_i = (x + y_i, y + x_i)$.
- VERIFIER: Send $(X, A_1, \ldots, A_k, B_1, \ldots, B_k)$ to PROVER.
- PROVER: Compute $b_1 \ldots b_k = H(X, A_1, \ldots, A_k)$. Let $X = (U, V)$, $A_i = (C_i, D_i)$ and $B_i = (E_i, F_i)$. Reject unless for each i either $b_i = 0$ and $A_i = (g^{E_i}, Rg^{F_i})$, $b_i = 0$ and $A_i = (Rg^{E_i}, R^{F_i})$, $b_i = 1$ and $(UC_i, VD_i) = (Rg^{E_i}, Rg^{F_i})$ or $b_i = 1$ and $(UD_i, VC_i) = (Rg^{E_i}, Rg^{F_i})$.
- PROVER: Pick a random $z \in Z_p^*$. Let N_0 be the response to challenge 0 in Blum's protocol. Let N_1 be the response to challenge 1 in Blum's protocol. Encrypt N_0 using a secure private-key encryption scheme with key U^z. Encrypt N_1 using a secure private key encryption scheme with key V^z. Send g^z and both encryptions to VERIFIER.
- VERIFIER: If $X = (g^x, Rg^y)$ decrypt the first encryption with key $(g^z)^x$ and accept if it is a proper response to challenge 0 in the Blum protocol. If $X = (Rg^x, g^y)$ decrypt the second encryption with key $(g^z)^y$ and accept if it is a proper response to challenge 1 in the Blum protocol.

Theorem 1 *The above protocol is a Zero-Knowledge Proof System for Hamiltonian Path*

4 A Protocol Based on a Different Proof of Knowledge

This protocol is also based on Blum's protocol[2] for Hamiltonian Path. It differs from the previous protocol in that it is based on the hardness of factoring instead of the hardness of discrete log.

- PROVER: Sends (n, H) to VERIFIER where n is the product of two randomly chosen prime numbers and H is a hash function whose range is $\{0,1\}^k$.
- VERIFIER: Selects a random x in Z_n^*. Let $X = x^2 \mod n$.
- VERIFIER: Selects k random numbers w_i in Z_n^*. Let $W_i = w_i^2 \mod n$.
- VERIFIER: Let $b_1 \dots b_k = H(X, W_1, \dots, W_k)$. Let $B_i = w_i x_i^{b_i}$.
- VERIFIER: Send $(X, W_1, \dots, W_k, B_1, \dots, B_k, \bar{R})$ to PROVER, where $\bar{R}$ is a randomly chosen string.
- PROVER: Compute $b_1 \dots b_k = H(X, W_1, \dots, W_k)$. Reject unless for each i, $B_i^2 = W_i X^{b_i}$.
- PROVER: Let y and z be the two square roots of X in Z_n^*. Pick a sequence of k random strings R_i. Let K_y be the k-bit string whose i^{th} bit is $< R_i, y >$[1]. Similarly, let K_z be the k-bit string whose i^{th} bit is $< R_i, z >$.
- PROVER: Let N_0 be the response to challenge 0 in Blum's protocol. Let N_1 be the response to challenge 1 in Blum's protocol. Encrypt $N_{<\bar{R},y>}$ using a secure private-key encryption scheme with key K_y. Encrypt $N_{<\bar{R},z>}$ using a secure private key encryption scheme with key K_z. Send $(R_1, \dots, R_k)$ and both encryptions to VERIFIER.
- VERIFIER: Let K_x be the k-bit string whose i^{th} bit is $< R_i, y >$. Attempt to decrypt both encryptions with key K_x. Accept only if one of the decryptions is a correct response to challenge $< \bar{R}, x >$ in Blum's protocol.

5 Conclusion

We believe that this protocol is an improvement over the Hada Tanaka protocol for the following reasons:

1. We feel that our assumption is more believable than the Strong Diffie-Hellman assumption used in the Hada Tanaka protocol because our assumption is based on a widely used Proof of Knowledge.

[1] Where $< R_i, y >$ is the inner product of R_i and y

2. We also prefer our Proof of Knowledge Assumption to the Strong Diffie-Hellman assumption because our assumption is really a class of assumptions. Instead of starting with a proof of knowledge for discrete log, a protocol similar to ours could be created based on a different Proof of Knowledge.

3. We believe that the proof that our protocol is a Zero-Knowledge Proof System is much simpler than the proof required for the Hada Tanaka protocol.

4. The protocol that we present is a Zero-Knowledge Proof System. That is, it is sound even if the prover is computationally unbounded.

5. In addition to the Strong Diffie Hellman assumption, the Hada Tanaka protocol required an assumption that Discrete Log is hard for all primes, p, of the form $2q+1$. Our protocol requires us to assume only that Discrete Log is hard for a randomly chosen prime.

References

[1] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In *Proceedings of Crypto'89*, 1989.

[2] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, 1986.

[3] G. Brassard, D. Chaum, and C. Crèpeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2), 1988.

[4] O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal of Computing*, 25(1), 1996.

[5] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal of Computing*, 18(1), 1989.

[6] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In *Proceedings of Crypto'98*, 1998.