Zero Knowledge, We know everything…!

Participants: Viking Mann, Pawandeep Gill, Ariel Machado, Amanda Wong
Professor: Charles Lam Assistant: Frank Madrid

Abstract

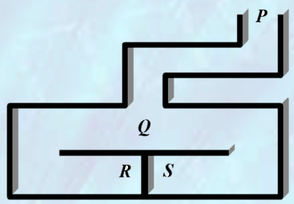
A zero knowledge proof is a method utilized in order to prove one party's identification to another party. Various proofs are used to prove to the "verifier" that the "claimant" knows a secret without disclosing the secret to a third party that might be spying

Background

Cryptography has been used over the centuries in an effort to send secret messages without the threat that an eavesdropping third party will discover the message. Prior to the digital age, couriers delivered secret keys and codes, allowing for the threat of a compromised courier or one that loses the secret to the enemy. The rise of the digital age has allowed for mass communication instantly, however an increase in technology allows for various attackers to record and store transmitted information. Thus, cryptography is as essential as ever, especially cryptography that allows one party to verify the identity of another party. Zero knowledge allows one to do just that, to verify a "Claimant" to a "Verifier".

General Idea Behind Zero Knowledge Proofs

This picture demonstrates the problem Zero Knowledge Proofs solve: How to show a Verifier that you know a secret without disclosing the secret.



In this example, Alice has the code to open the door between R and S, but she does not want Bob to know that she has the code. Furthermore, Bob wants to be certain that Alice has the code and another person is not impersonating Alice.

A single proof occurs such as this:
Bob stands at point P while Alice walks to either R or S. Then, Bob goes to point Q and shouts at Alice to appear from the side of his choice, R or S. Now there is a 50% chance that Alice is at the chosen point already. If Alice truly has the code, she unlocks the door to appear from whichever side Bob directs her to appear from. A mathematical demonstration of this concept is shown the Fiat-Shamir Identification Protocol.

.

Fiat-Shamir Identification Protocol

A simple protocol that effectively demonstrates Zero Knowledge proofs is the Fiat-Shamir identification protocol. The protocol establishes the three levels of correspondence between the verifier and the claimant, specifically, the witness, challenge, and response. This protocol is shown below in a bank-to-client verification:

A trusted source will generate a number, p*q to elicit n, where "p" and "q" are both primes. Generally, they are large primes, an RSA minimum of 1024 to 2048 bits (1024 represents a number roughly 308 digits). In this case, smaller numbers are used for simplicity. Let "p" be 101 and "q" be 103 resulting in an "n" of 10403. This "n" is made public therefore the bank, the client, and the adversary all know the public key.

The client, in order to begin, will first generate a personal public key "v" using a secret coprime number "s" and register that number with the trusted source for all to know. Say a coprime "s" of 23 is chosen by the client based upon the algorithm $1 \leq s \leq n-1$. Next the client must register their personal public key "v" by computing $v=s^2 \pmod n$. This would result in $529=23^2 \pmod{10403}$. This key is returned to the trusted source and made public.

Now the actual verification may begin. Say the client desires to view the amount of funds in their bank account and requests this privilege from the bank. The client first generates a random number "r" based upon $1 \leq r \leq n-1$, say in this case 21. That number is inputted into the equation $x=r^2 \pmod n$ to generate the witness. Thus $441=21^2 \pmod{10403}$.

Next the bank chooses either e=0 or e=1 and sends this challenge to the client. If the bank chooses e=0, then the client must use the algorithm y=r to compute a response, however if the bank challenges with e=1, then the client must compute using $y=r*s \pmod n$. The bank will then check the validity of the response utilizing the equation $y^2 \equiv x*v^e \pmod n$ Let the bank choose e=0 as a challenge. The client responds with y=r, 21. The bank verifies with $21 \equiv 441*529^0 \pmod{10403}$

However, if the bank chooses e=1 then the client must respond with y=rs (mod n). So, $483=21*23 \pmod{10403}$. The bank checks with $483^2 \equiv 441*529^1 \pmod{10403}$.

Reference: (s=23 v=529 N= 10403 r=21 x=441)

This protocol is executed 2^t with "t" being a large number to allow for identity certainty, because the claimant will only be admitted if all the rounds are correct.

Cheating The Fiat-Shamir Protocol

If the bank selected e=0, then the attacker simply has to give the bank the "r" they themselves created, forcing the bank to accept the statement.
If the bank selects e=1, then the attackers can bypass the security by obtaining "v" (a public knowledge key) in order to generate a special "x" by using $x=r^2/v \pmod n$ instead of computing the real equation $x=r^2 \pmod n$.
Therefore in the example, the attacker will send $7808 =21^2/529 \pmod{10403}$.
When the bank challenges with e=1, the attacker, instead of computing using $y=r*s \pmod n$, will simply send y=r (21). Then the bank will compute using the default $y^2 \equiv x*v^e \pmod n$. $21^2 \equiv 7808 *529^1 \pmod{10403}$. In both cases, the attacker has to guess the banks question, the equivalent of giving a response before one of two questions are asked and hoping the answer is correct. Thus if the protocol was executed as standard, it could be initiated $½^{80}$ times, resulting in a cheating success rate of $8.2718061x10^{-25}$

If one knows the primes that compose the public key, they may generate "s" through advanced number theory. However, the primes chosen as standard are very large in order to prevent factorization.

Guillou-Quisquater(GQ) Protocol

The Guillou-Quisquater(GQ) Protocol requires approximately three times the computational power of the Fiat-Shamir protocol.

To execute this protocol, the "Claimant" must obtain a sequence of numbers such as an ID card number to represent "J". "J" is the public key. The Claimant tries to prove to the "Verifier" that the credentials are theirs.
A random exponent "v" and a modulus "n" will also be made public. The modulus "n" will be a product of two large primes.
The private key B is computed so that $J*B^V=1 \pmod n$.

- The Claimant begins by sending the Verifier their credentials, "J". She then picks a random r so that $1 < r < n$.
- The Claimant sends the equation $T = r^v \pmod n$ to the Verifier.
- The Verifier also sends a random number "d" so that $0 < d < v$.
- The Claimant computes $D = r*B^d \pmod n$ and send it to the Verifier.
- The Verifier computes $T' = D^v*J^d \pmod n$.
- If $T = T' \pmod n$, then the authentication succeeds.

Therefore, it is more complex than the Fiat-Shamir Protocol, making it harder to break.

Future Research

As computer hardware shrinks and processing power rises, the ability to break systems increases. This is directly related to the fact that greater processing power allows for even larger numbers to be factored, resulting in security protocols being compromised. This calls for newer algorithms or larger parameters that reduce vulnerability.

Works Cited
"2.1.8 What Are Interactive Proofs and Zero-knowledge Proofs?" *RSA Security*. RSA Labs, n.d. Web. 06 Aug. 2013.

Aronsson, Hannu A. "Zero Knowledge Protocols and Small Systems." *Network Security: Zero Knowledge and Small Systems*. Helsinki University of Technology, n.d. Web. 06 Aug. 2013.

Menezes, A. J., Oorschot Paul C. Van, and Scott A. Vanstone. *Handbook of Applied Cryptography*. Boca Raton: CRC, 1997. Print.

This material is based upon work supported by the National Science Foundation under Grant No. 1241636. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

CSU Bakersfield REVS UP Research Experience Vitalizing Science University Program

CSU Bakersfield Research Experience Vitalizing Science — University Program

Chevron

National Science Foundation