Zero Knowledge Proofs

Comp 4109

Bill Ewanick

March 31, 2011

Zero Knowledge Proofs (ZKP)

- An interactive method for one party to prove that a statement is true

Zero Knowledge Proofs

- An interactive method for one party to prove that a statement is true
- This is done without revealing anything other than the statement is true

Zero Knowledge Proofs

- An interactive method for one party to prove that a statement is true
- This is done without revealing anything other than the statement is true
- Quick Example:
  Prove to the prof that you deserve an A+ without taking the exam.

Quick History

- ZKPs were first conceived in 1985 by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in the draft paper:
  - The Knowledge Complexity of Interactive Proof-Systems
- They also gave the first zero-knowledge proof for a concrete problem, that of deciding quadratic nonresidues mod m.

ZKP: So Easy a Child Can Get It

- Jean-Jacques Quisquater published a paper entitled

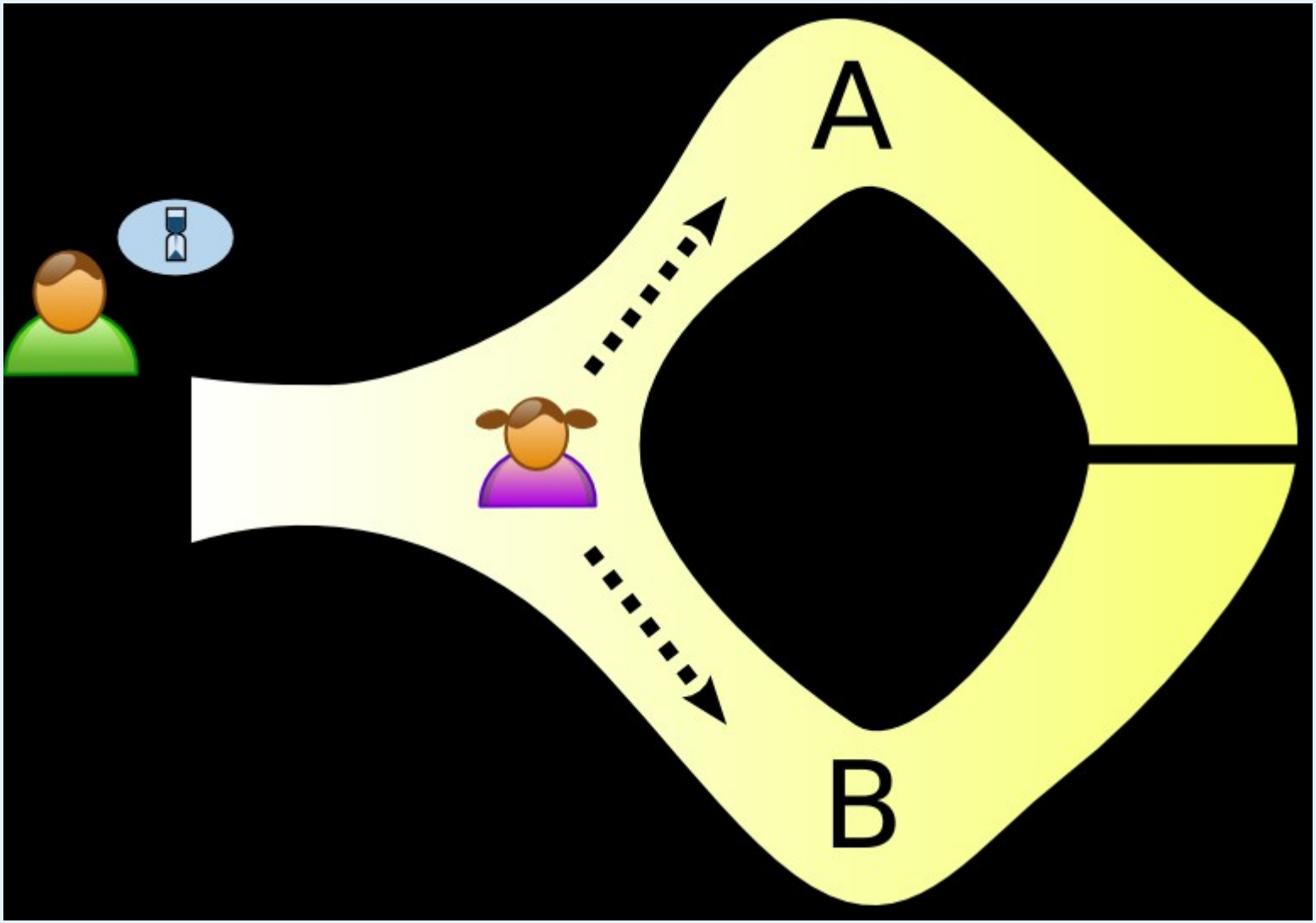
  “How to Explain Zero-Knowledge Protocols to Your Children”

  which contains several simple examples of ZKPs.

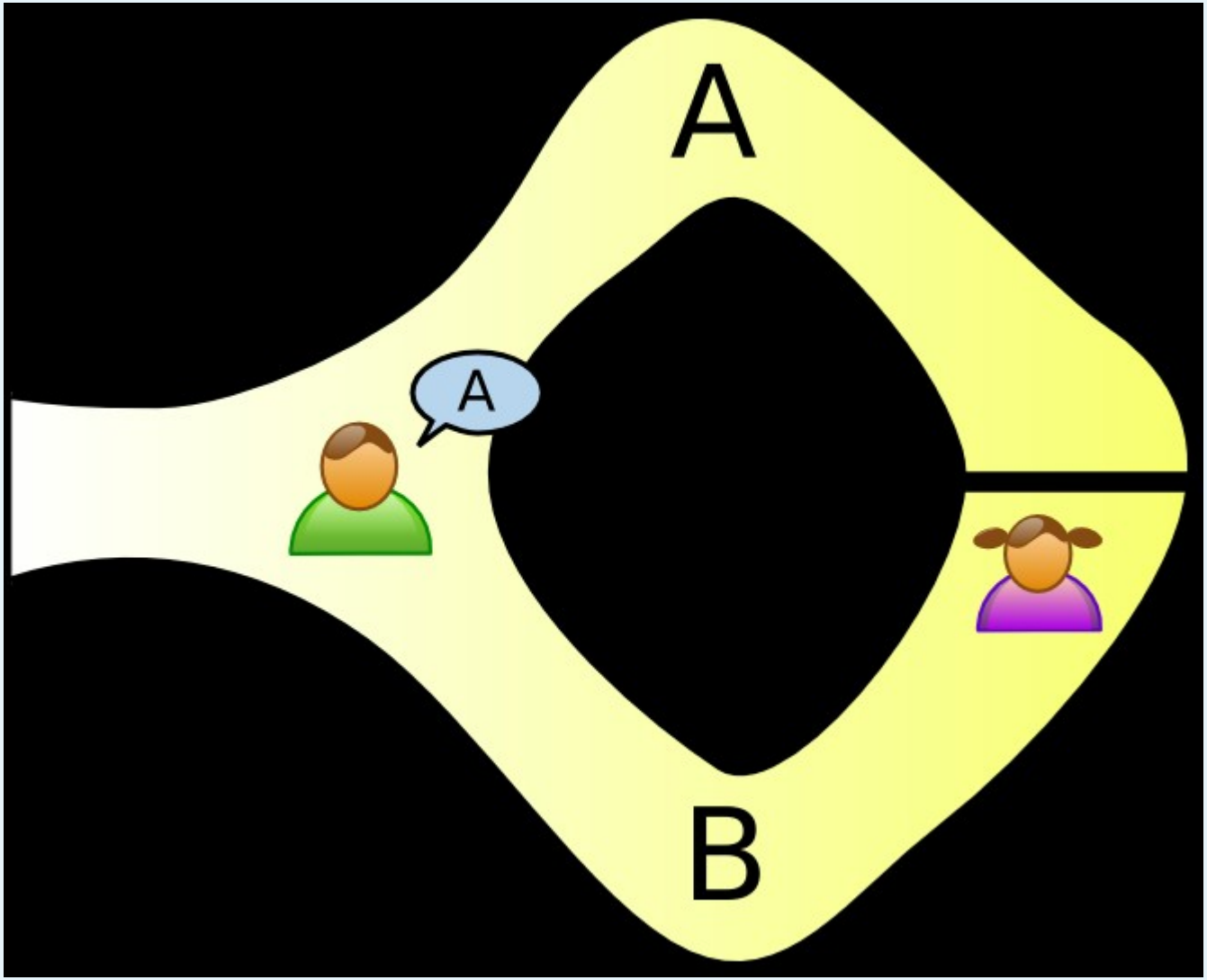
Peggy and Victor

- Instead of Alice and Bob, we have
- Peggy, the prover
- Victor, the verifier

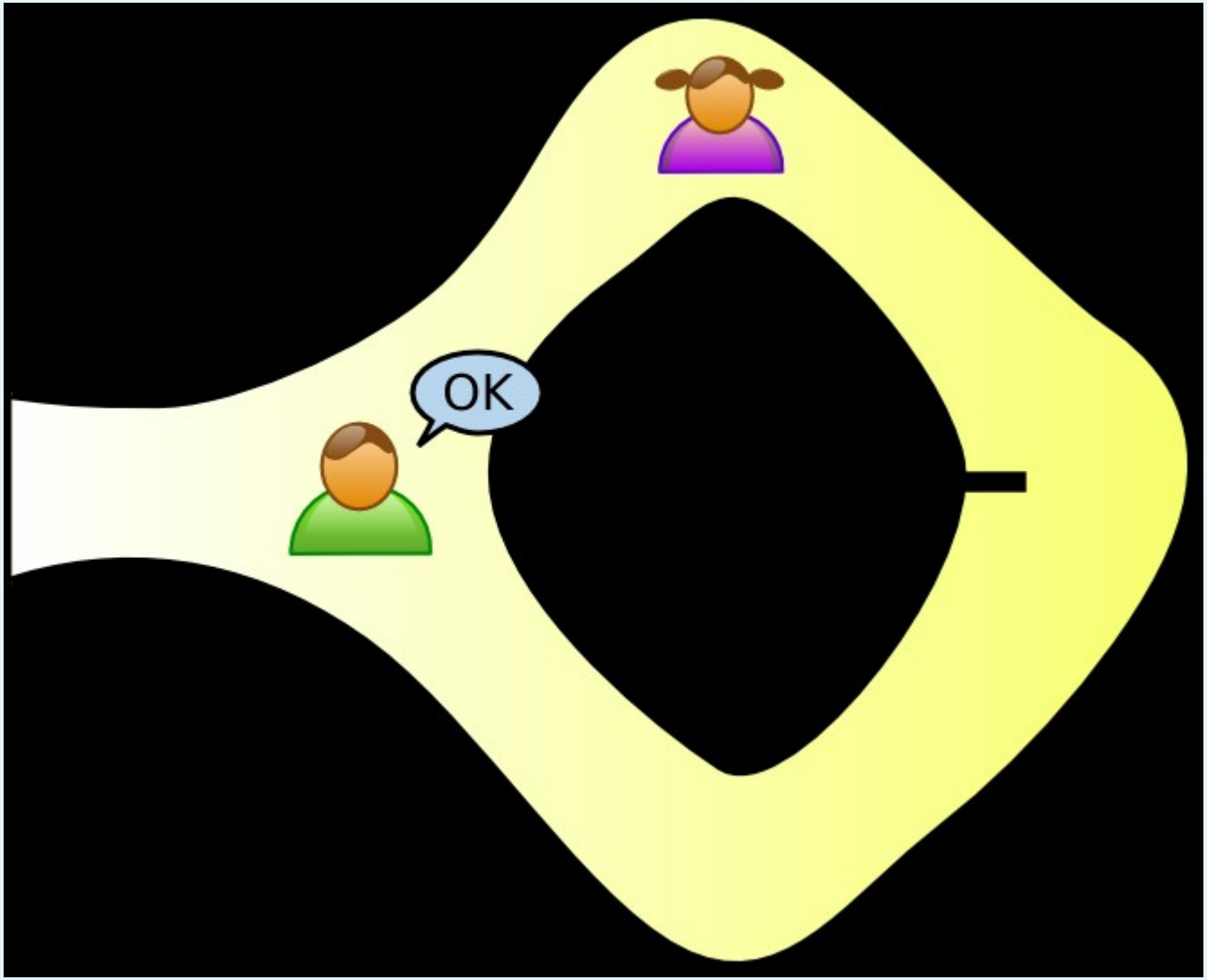
Simple Example



Simple Example



Simple Example



Real Definition

- A Zero Knowledge Proof must satisfy three properties.

Real Definition

1) Completeness:

- If the statement is true, the honest verifier (who follows the protocol properly) will be convinced of this fact by an honest prover.

Real Definition

1) Completeness
2) Soundness:
   - If the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.

Soundness Property in Detail

- An interactive proof is *sound* if there exists an expected polynomial-time algorithm M with the following property:

Soundness Property in Detail

- An interactive proof is *sound* if there exists an expected polynomial-time algorithm M with the following property:
- if a dishonest prover can with non-negligible probability successfully execute the proof, then M can be used to extract from the prover knowledge which with overwhelming probability allows successful subsequent proof executions.

Soundness

- Alternate explanation:
- The prover's secret, s, together with public data satisfies some polynomial-time predicate can be extracted, allowing successful execution of subsequent protocol instances.

What Does It Mean?

- A proof is sound if it is not possible for a dishonest prover to convince an honest verifier, due to the honest prover's secret knowledge.
- This says nothing about the ease of acquiring the prover's secret knowledge.

Real Definition

1) Completeness
2) Soundness

These two properties constitute a more general “Interactive Proof System”.

The last attribute is needed to for the proof to be zero-knowledge.

Real Definition

1) Completeness
2) Soundness
3) Zero-Knowledge
   - If the statement is true, no cheating verifier learns anything other than this fact.

In Depth Zero-Knowledge

- A protocol which is a proof of knowledge has the zero-knowledge property if it is simulatable in the following sense:

In Depth Zero-Knowledge

- There exists an expected polynomial-time algorithm (*simulator*) which can produce, upon input of the assertions to be proven but without interacting with the real prover, transcripts indistinguishable from those resulting from interaction with the real prover.

Real Definition

1) Completeness
2) Soundness
3) Zero-Knowledge

Computational Vs. Perfect ZK

- A protocol is *computationally* zero-knowledge if an observer restricted to probabilistic polynomial-time tests cannot distinguish real from simulated transcripts.

Computational Vs. Perfect ZK

- A protocol is *perfectly* zero-knowledge if the probability distributions of the transcripts are identical.

ZK Property and Soundness Vs. Security

- Neither the Zero-Knowledge property nor the Soundess property guarantee that a protocol is secure (ie. The probability of it being easily defeated is negligible)

ZK Property and Soundness Vs. Security

- These two properties have little value unless the underlying problem faced by an opponent is computationally hard.

More Realistic Example

- Based off the Fiat-Shamir identification protocol, which is beyond the scope of this talk.
- Peggy is going to prove that she knows a secret number *s* to Victor without revealing what *s* is.

One-Time Setup

- Let T be a trusted third party.

One-Time Setup

- Let T be a trusted third party.
- T selects and publishes an RSA-like modulus n = pq, but keeps p and q secret.

One-Time Setup

- Let T be a trusted third party.
- T selects and publishes an RSA-like modulus n = pq, but keeps p and q secret.
- Peggy selects a secret *s* coprime to *n*, such that $1 \leq s \leq n-1$.

One-Time Setup

- Let T be a trusted third party.
- T selects and publishes an RSA-like modulus n = pq, but keeps p and q secret.
- Peggy selects a secret *s* coprime to *n*, such that $1 \leq s \leq n-1$.
- Peggy computes $v = s^2 \bmod n$.

One-Time Setup

- Let T be a trusted third party.
- T selects and publishes an RSA-like modulus n = pq, but keeps p and q secret.
- Peggy selects a secret *s* coprime to *n*, such that $1 \leq s \leq n-1$.
- Peggy computes $v = s^2 \bmod n$.
- Peggy registers v with T as her public key.

Protocol Messages

- Each of the *t* rounds has three messages with the following form:
  - Peggy → Victor: $x = r^2 \bmod n$
  - Victor → Peggy: $e \in \{0, 1\}$
  - Peggy → Victor: $y = r \cdot s^e \bmod n$
- The variables e and r will be explained in the next few slides.

Protocol Actions

- The following steps are iterated *t* times, sequentially and independently.
- Victor accepts the proof that Peggy knows *s* if all *t* rounds succeed.

Protocol Actions

1) Peggy chooses a random number *r* such that $1 \leq r \leq n-1$ and sends $x = r^2 \bmod n$ to Victor.

Protocol Actions

1) Peggy chooses a random number *r* such that $1 \leq r \leq n-1$ and sends $x = r^2 \bmod n$ to Victor.

2) Victor randomly selects a challenge bit $e = 0$ or $e = 1$, and sends *e* to Peggy.

Protocol Actions

1) Peggy chooses a random number *r* such that
   $1 \le r \le n-1$ and sends $x = r^2 \bmod n$ to Victor.
2) Victor randomly selects a challenge bit
   $e = 0$ or $e = 1$, and sends e to Peggy.
3) Peggy computes and sends to Victor
   $y = r \cdot s^e \bmod n$.

Protocol Actions

1) Peggy chooses a random number *r* such that
   $1 \le r \le n-1$ and sends $x = r^2 \bmod n$ to Victor.
2) Victor randomly selects a challenge bit
   $e = 0$ or $e = 1$, and sends *e* to Peggy.
3) Peggy computes and sends to Victor
   $y = r \cdot s^e \bmod n$.
   - Either $y = r$ (if $e = 0$), or $y = rs \bmod n$ (if $e = 1$)

Protocol Actions

1) Peggy chooses a random number *r* such that
   $1 \le r \le n-1$ and sends $x = r^2 \bmod n$ to Victor.
2) Victor randomly selects a challenge bit
   $e = 0$ or $e = 1$, and sends e to Peggy.
3) Peggy computes and sends to Victor
   $y = r \cdot s^e \bmod n$.
   - Either $y = r$ (if $e = 0$), or $y = rs \bmod n$ (if $e = 1$)
4) Victor rejects the proof if $y = 0$, and otherwise
   accepts upon verifying $y^2 \equiv x \cdot v^e \pmod n$

Protocol Actions

- The previous steps are iterated *t* times, sequentially and independently.
- Victor accepts the proof that Peggy knows *s* if all *t* rounds succeed.

Justifications

- The challenge *e* requires that Peggy be able to answer two questions.

Justifications

- The challenge *e* requires that Peggy be able to answer two questions.
- One question demonstrates her knowledge of the secret s.

Justifications

- The challenge *e* requires that Peggy be able to answer two questions.
- One question demonstrates her knowledge of the secret s.
- The other question is an easy one (for honest provers) to prevent cheating.

Justifications

- An opponent impersonating Peggy might try to cheat in the following way:

Justifications

- An opponent impersonating Peggy might try to cheat in the following way:
- By choosing any *r* and setting $x = r^2/v$, the imposter could answer the challenge e=1 with a 'correct' answer.

Justifications

- An opponent impersonating Peggy might try to cheat in the following way:
- By choosing any *r* and setting $x = r^2/v$, instead of r^2 mod n, the imposter could answer the challenge e=1 'correctly'.
- However, they would still be unable to answer the challenge e=0.

Justifications

- An opponent impersonating Peggy might try to cheat in the following way:
- By choosing any *r* and setting $x = r^2/v$, instead of r^2 mod n, the imposter could answer the challenge e=1 'correctly'.
- However, they would still be unable to answer the challenge e=0.
- This is due to the required knowledge of a square root of *x* mod *n*, ie. $\sqrt{\frac{r^2}{v}}$

Example

- Cheating Peggy sets $x = r^2/v$

Example

- Cheating Peggy sets $x = r^2/v$
- Victor requests $y = rs$

Example

- Cheating Peggy sets $x = r^2/v$
- Victor requests $y = rs$
- Cheating Peggy sends $y = r$

Example

- Cheating Peggy sets $x = r^2/v$
- Victor requests $y = rs$
- Cheating Peggy sends $y = r$
- Victor attempts to verify - $y^2 \equiv x \cdot v^e$
  - $x \cdot v^e \equiv r^2/v \cdot v \equiv r^2 \equiv y^2 \pmod{n}$

Example

- Cheating Peggy sets $x = r^2/v$
- Victor requests $y = rs$
- Cheating Peggy sends $y = r$
- Victor attempts to verify - $y^2 \equiv x \cdot v^e$
  - $x \cdot v^e \equiv r^2/v \cdot v \equiv r^2 \equiv y^2 \pmod{n}$
- Victor believes Peggy knows s

Example

- If Cheating Peggy sends $x = r^2/v$, Victor expects her to send $x = r^2$

Example

- If Cheating Peggy sends $x = r^2/v$, Victor expects her to send $x = r^2$
- If Victor asks for $y = r$, he'll try to compute
  - $x \equiv y^2 \rightarrow r^2/v \equiv y^2 \rightarrow r/\sqrt{v} \neq y$

Example

- If Cheating Peggy sends $x = r^2/v$, Victor expects her to send $x = r^2$
- If Victor asks for $y = r$, he'll try to compute
  - $x \equiv y^2 \rightarrow r^2/v \equiv y^2 \rightarrow r/\sqrt{v} \neq y$
- This fails, so Victor knows Peggy is cheating.

Chance of Detection

- An opponent who doesn't know *s* can only answer one of the two challenges.
- This gives a probability of ½ of escaping detection.
- We execute the protocol *t* times to reduce the likelihood of cheating to 2^{-t}.

Zero Knowledge Property

- In the previous example, Peggy reveals she knows *s*, but not any information about *s*.

Zero Knowledge Property

- In the previous example, Peggy reveals she knows *s*, but not any information about *s*.
- The response $y = r$ is totally independent from *s*.

Zero Knowledge Property

- In the previous example, Peggy reveals she knows s, but not any information about s.
- The response $y = r$ is totally independent from s.
- The response $y = rs \bmod n$ also provides no information about s because the random r is unknown to Victor.

Zero Knowledge Property

- Information pairs (x, y) extracted from Peggy could be simulated by Victor alone by choosing y randomly, then defining $x = y^2$ or $y^2/v \bmod n$.

Zero Knowledge Property

- Information pairs (x, y) extracted from Peggy could be simulated by Victor alone by choosing y randomly, then defining $x = y^2$ or $y^2/v \bmod n$.
- Peggy wouldn't create pairs this way, but they have a probability distribution indistinguishable from the ones Peggy would create.

Zero Knowledge Property

- Information pairs (x, y) extracted from Peggy could be simulated by Victor alone by choosing y randomly, then defining $x = y^2$ or $y^2/v \bmod n$.
- Peggy wouldn't create pairs this way, but they have a probability distribution indistinguishable from the ones Peggy would create.
- This establishes the Zero-Knowledge Property.

Example With Numbers

- Here is the previous example with numbers.
- Let p = 5, q = 7. Then, n = pq = 35.
- Peggy secretly chooses s = 16, which is coprime to n. She publishes $v = s^2 \bmod n = 11$ to T.

Example With Numbers

- We'll assume Victor only needs 2 successful tests to be proven.

Example With Numbers

- We'll assume Victor only needs 2 successful tests to be proven.
- Peggy randomly selects r = 10. She sends $x = r^2 \bmod n = 10^2 \bmod 35 = 30$ to Victor.

Example With Numbers

- We'll assume Victor only needs 2 successful tests to be proven.
- Peggy randomly selects r = 10. She sends $x = r^2 \bmod n = 10^2 \bmod 35 = 30$ to Victor.
- Victor randomly selects e = 0, sends to Peggy.

Example With Numbers

- Recall the challenge formula:
  - $y = r \cdot s^e \bmod n$

Example With Numbers

- Recall the challenge formula:
  - $y = r \cdot s^e \bmod n$
- Since e = 0, Peggy computes y = r = 10, and sends it to Victor.

Example With Numbers

- Recall the challenge formula:
  - $y = r \cdot s^e \bmod n$
- Since e = 0, Peggy computes y = r = 10, and sends it to Victor.
- Victor verifies that $y^2 = 10^2 \equiv 30 \pmod{35}$

Example With Numbers

- Recall the challenge formula:
  - $y = r \cdot s^e \bmod n$
- Since e = 0, Peggy computes y = r = 10, and sends it to Victor.
- Victor verifies that $y^2 = 10^2 \equiv 30 \pmod{35}$
- First challenge successfully proven.

Second Challenge

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

Second Challenge

- Peggy randomly selects r = 20. She sends $x = r^2 \bmod n = 20^2 \bmod 35 = 15$ to Victor.
- Victor randomly sends e = 1 to Peggy.

Second Challenge

- Peggy randomly selects r = 20. She sends
  $x = r^2 \bmod n = 20^2 \bmod 35 = 15$ to Victor.
- Victor randomly sends e = 1 to Peggy.
- Peggy computes $y = r \cdot s^e \bmod n$, or
  $y = 20 \cdot 16 \bmod 35 = 5$, sends to Victor.

Second Challenge

- Peggy randomly selects r = 20. She sends
  $x = r^2 \bmod n = 20^2 \bmod 35 = 15$ to Victor.
- Victor randomly sends e = 1 to Peggy.
- Peggy computes $y = r \cdot s^e \bmod n$, or
  $y = 20 \cdot 16 \bmod 35 = 5$, sends to Victor.
- Victor verifies that $y^2 = 25 \equiv 15 \cdot 11 \pmod{35}$

End of Challenges

- Peggy successfully completed t = 2 rounds, so Victor accepts.

Use in Cryptography

- Very useful in authentication schemes.
- Blocks eavesdroppers from discovering secret information.
- Able to enforce honest behavior while maintaining privacy.

Use in Cryptography

- Other problems used include:
  - Graph Isomorphism
  - Graph Three-Colorability
  - Every NP-Complete Problem
- Fiat-Shamir Identification Protocl is not normally implemented in modern cryptosystem.
- However, it is the basis of existing zero-knowledge entity authentication schemes.

ZKP and NP Completeness

- The original ZKP on quadratic nonresidue, along with the graph coloring problem, are all NP-Complete.
- Since every problem in NP can be reduced to every other one, there is a ZKP for all problems in NP.

References

- Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography. 5th Ed. CRC Press, 2001.
- Jean-Jacques Quisquater, Louis C. Guillou, Thomas A. Berson. How to Explain Zero-Knowledge Protocols to Your Children (http://pages.cs.wisc.edu/~mkowalcz/628.pdf). Advances in Cryptology - CRYPTO '89: Proceedings, v.435 p.628-631, 1990.
- Austin Mohr, "A Survey of Zero-Knowledge Proofs with Applications to Cryptography". http://www.austinmohr.com/work/files/zkp.pdf
- "Zero-knowledge proof." Wikipedia, The Free Encyclopedia.
- Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. 1989. http://crypto.cs.mcgill.ca/~crepeau/COMP647/2007/TOPIC02/GMR89.pdf

Quiz

1) Who authored the paper “How to Explain Zero-Knowledge Protocols to Your Children”?

2) What three properties must a Zero-Knowledge Proof satisfy?

3) What does it mean for a protocol to be *perfectly* zero-knowledge?

4) Instead of Alice and Bob, what names are traditionally used in Zero-Knowledge Proofs?

5) When were Zero-Knowledge Proofs first conceived?

Bonus) Name one of the authors from that paper.