Chapter 14 Zero-Knowledge Technique

What Is ZKP?

This is more commonly referred to as Zero-Knowledge Proof.

What Is ZKP?

This is more commonly referred to as Zero-Knowledge Proof.

Definition

A Zero-Knowledge Proof (ZKP) is an interactive method for one party to prove to another that a statement is true without anything other than the verity of the statement.

What Is ZKP?

This is more commonly referred to as Zero-Knowledge Proof.

Definition

A Zero-Knowledge Proof (ZKP) is an interactive method for one party to prove to another that a statement is true without anything other than the verity of the statement.

A ZKP must satisfy three properties:

1. Completeness: if the statement is true, the honest verifier will be convinced of this fact by an honest prover

What Is ZKP?

This is more commonly referred to as Zero-Knowledge Proof.

Definition

A Zero-Knowledge Proof (ZKP) is an interactive method for one party to prove to another that a statement is true without anything other than the verity of the statement.

A ZKP must satisfy three properties:

1. Completeness: if the statement is true, the honest verifier will be convinced of this fact by an honest prover
2. Soundness: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability

What Is ZKP?

This is more commonly referred to as Zero-Knowledge Proof.

Definition

A Zero-Knowledge Proof (ZKP) is an interactive method for one party to prove to another that a statement is true without anything other than the verity of the statement.

A ZKP must satisfy three properties:

1. Completeness: if the statement is true, the honest verifier will be convinced of this fact by an honest prover
2. Soundness: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability
3. Zero-Knowledgeness: if the statement is true, no cheating verifier learns anything other than the statement is true

More Basics

The first two are properties of more general interactive proof systems. The most common use of a ZKP is in authentication systems where one party wants to be able to prove its identity to a second party via some secret information (such as a password) but doesn't want the second party to learn anything about the secret.

More Basics

The first two are properties of more general interactive proof systems. The most common use of a ZKP is in authentication systems where one party wants to be able to prove its identity to a second party via some secret information (such as a password) but doesn't want the second party to learn anything about the secret.

Note: ZKPs are not proofs in the mathematical sense of the term because there is some small probability (called the soundness error) that a cheating prover will be able to convince the verifier of a false statement. However, there are standard techniques to decrease the soundness error to any arbitrarily small value.

More Basics

The first two are properties of more general interactive proof systems. The most common use of a ZKP is in authentication systems where one party wants to be able to prove its identity to a second party via some secret information (such as a password) but doesn't want the second party to learn anything about the secret.

Note: ZKPs are not proofs in the mathematical sense of the term because there is some small probability (called the soundness error) that a cheating prover will be able to convince the verifier of a false statement. However, there are standard techniques to decrease the soundness error to any arbitrarily small value.

Commonly, we refer to the prover as Peggy and the verifier as Victor (this is not just in our textbook). Also, sometimes, we refer to the parties as Alice and Bob.

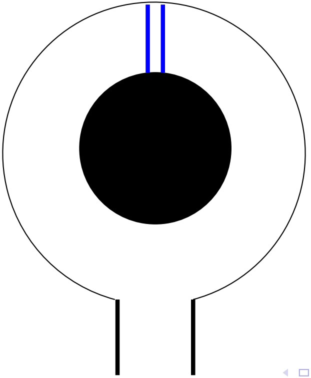
A Basic Example Protocol

This is a well-knows story presenting ZKP that was first published by Jean-Jacques Quisquatal in ‘How to Explain Zero-Knowledge Protocols to Your Children’.

A Basic Example Protocol

This is a well-knows story presenting ZKP that was first published by Jean-Jacques Quisquatal in 'How to Explain Zero-Knowledge Protocols to Your Children'.

In the story, Peggy has uncovered the secret word that opens a magic door in the back of a cave. The cave is shaped like a circle with the entrance on one side and the magic door blocking the opposite side.



A Basic Example Protocol

Victor says he will pay her for the secret word but not until he is sure she really knows it. Peggy says she will tell him the secret word but not until she gets the money. They devise a scheme by which Peggy can prove she knows the secret word without telling it to Victor.

A Basic Example Protocol

Victor says he will pay her for the secret word but not until he is sure she really knows it. Peggy says she will tell him the secret word but not until she gets the money. They devise a scheme by which Peggy can prove she knows the secret word without telling it to Victor.

First, Victor waits outside the cave as Peggy goes in. She randomly enters either the path on the left or the one on the right. Then Victor enters the cave and shouts the name of the path he wants her to use to return, either the left or the right at random. Providing she really does know the secret word, this is easy; she opens the door, if necessary, as returns along the desired path. Note that Victor does now know which path Peggy has gone down.

A Basic Example Protocol

However, suppose she did not know the secret word. Then, she would only be able to return by the named path if Victor were to name the path that she had just entered. Since Victor would want to choose the path at random, she would have a 50% chance of guessing correctly. If they were to repeat this trick many times, say 20 times in a row, her chance of anticipating all of Victor's requests becomes very small (2^{-20}). So, if Peggy reliably appears at the exit Victor names, he can conclude she very likely has the secret word.

However, suppose she did not know the secret word. Then, she would only be able to return by the named path if Victor were to name the path that she had just entered. Since Victor would want to choose the path at random, she would have a 50% chance of guessing correctly. If they were to repeat this trick many times, say 20 times in a row, her chance of anticipating all of Victor's requests becomes very small (2^{-20}). So, if Peggy reliably appears at the exit Victor names, he can conclude she very likely has the secret word.

If there was another observer, there is no way to convince them that Peggy knows the secret word since she and Victor could have planned the sequence.

Color Example

The classic example is the following: Imagine your friend is color-blind. You have two billiard balls; one is red, one is green, but they are otherwise identical. To your friend they seem completely identical, and he is skeptical that they are actually distinguishable. You want to prove to him that they are in fact differently-colored. On the other hand, you do not want him to learn which is red and which is green.

Color Example

The classic example is the following: Imagine your friend is color-blind. You have two billiard balls; one is red, one is green, but they are otherwise identical. To your friend they seem completely identical, and he is skeptical that they are actually distinguishable. You want to prove to him that they are in fact differently-colored. On the other hand, you do not want him to learn which is red and which is green.

Here is the proof system. You give the two balls to your friend so that he is holding one in each hand. You can see the balls at this point, but you don't tell him which is which. Your friend then puts both hands behind his back. Next, he either switches the balls between his hands, or leaves them be, with probability $\frac{1}{2}$ each. Finally, he brings them out from behind his back. You now have to 'guess' whether or not he switched the balls.

Color Example

By looking at their colors, you can of course say with certainty whether or not he switched them. On the other hand, if they were the same color and hence indistinguishable, there is no way you could guess correctly with probability higher than $\frac{1}{2}$.

Color Example

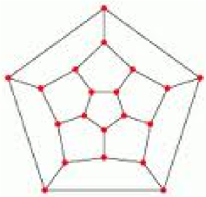
By looking at their colors, you can of course say with certainty whether or not he switched them. On the other hand, if they were the same color and hence indistinguishable, there is no way you could guess correctly with probability higher than $\frac{1}{2}$.

If you and your friend repeat this 'proof' t times (for large t), your friend should become convinced that the balls are indeed differently colored; otherwise, the probability that you would have succeeded at identifying all the switches and non-switches is at most $\left(\frac{1}{2}\right)^t$. Furthermore, the proof is 'zero-knowledge' because your friend never learns which ball is green and which is red; indeed, he gains no knowledge about how to distinguish the balls.

A More Practical Example

Peggy knows a Hamiltonian cycle for a large graph G, which is Peggy's public key. Victor knows G, since it is public, but not the cycle. Peggy will prove she knows the cycle without revealing it. However, Peggy does not want to simply reveal the Hamiltonian cycle or any other information to Victor. Maybe she is the only one who knows the information? Maybe he wants to buy the information but wants verification first? Here is why the Hamiltonian cycle creates a problem:

Sir William Rowan Hamilton (1805-1865) was an Irish physicist, astronomer and mathematician. His major work was in reformulating Newtonian mechanics, which was renamed Hamiltonian mechanics. As a result of his work was his invention of a puzzle known as the Icosian Game in 1857. One question the game posed is whether it was possible to start at a vertex of the above graph and return to that vertex by visiting every other vertex once each. This idea became known as a Hamiltonian circuit.



Terms

Definition

A Hamiltonian path is a path that visits each vertex once.

Terms

Definition

A Hamiltonian path is a path that visits each vertex once.

Definition

A Hamiltonian cycle (or circuit) is a closed path that visits each vertex once.

Terms

Definition

A Hamiltonian path is a path that visits each vertex once.

Definition

A Hamiltonian cycle (or circuit) is a closed path that visits each vertex once.

Definition

A graph that has a Hamiltonian cycle is called Hamiltonian.

Terms

Definition

A Hamiltonian path is a path that visits each vertex once.

Definition

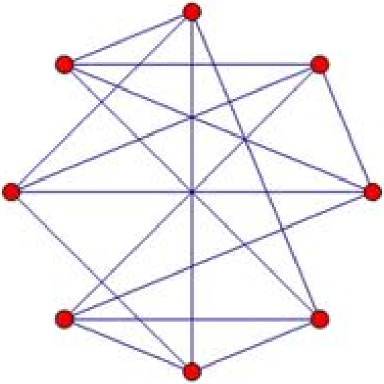
A Hamiltonian cycle (or circuit) is a closed path that visits each vertex once.

Definition

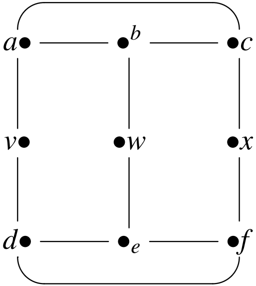
A graph that has a Hamiltonian cycle is called Hamiltonian.

The question is, when does a graph have a Hamiltonian cycle? a Hamiltonian path?

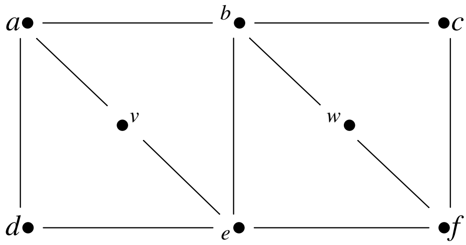
Is This Graph Hamiltonian?



Is This Graph Hamiltonian?



Is This Graph Hamiltonian?



When Is A Graph Hamiltonian?

Theorem

Ore's Theorem(1960) *Suppose that G is a graph with $n \geq 3$ vertices and for all distinct nonadjacent vertices x and y,*

$$deg(x) + deg(y) \geq n$$

The G has a Hamiltonian circuit.

When Is A Graph Hamiltonian?

Theorem

Ore's Theorem(1960) *Suppose that G is a graph with $n \geq 3$ vertices and for all distinct nonadjacent vertices x and y,*

$$deg(x) + deg(y) \geq n$$

The G has a Hamiltonian circuit.

Suppose that G has no Hamiltonian circuit. We will show that for some nonadjacent vertices $x, y \in V(G)$,

$$deg_G(x) + deg_G(y) < n \qquad (*)$$

where $deg_G(a)$ means the degree of a in G.

When Is A Graph Hamiltonian?

Theorem

Ore's Theorem(1960) *Suppose that G is a graph with $n \geq 3$ vertices and for all distinct nonadjacent vertices x and y,*

$$deg(x) + deg(y) \geq n$$

The G has a Hamiltonian circuit.

Suppose that G has no Hamiltonian circuit. We will show that for some nonadjacent vertices $x, y \in V(G)$,

$$deg_G(x) + deg_G(y) < n \qquad (*)$$

where $deg_G(a)$ means the degree of a in G.

If we add edges to G, we eventually obtain a complete graph, which has a Hamiltonian circuit.

Ore's Theorem

Thus, in the process of adding edges, we must eventually hit a graph H with the property that H has no Hamiltonian circuit but adding any more edges to H gives us a graph with a Hamiltonian circuit.

Thus, in the process of adding edges, we must eventually hit a graph H with the property that H has no Hamiltonian circuit but adding any more edges to H gives us a graph with a Hamiltonian circuit.

We will show that in H, there are nonadjacent x and y so that

$$deg_H(x) + deg_H(y) < n \qquad (**)$$

Ore's Theorem

Thus, in the process of adding edges, we must eventually hit a graph H with the property that H has no Hamiltonian circuit but adding any more edges to H gives us a graph with a Hamiltonian circuit.

We will show that in H, there are nonadjacent x and y so that

$$deg_H(x) + deg_H(y) < n \qquad (**)$$

But $deg_G(a) \leq deg_H(a)$ for all a, so $(**)$ implies $(*)$.

Ore's Theorem

Pick any nonadjacent vertices x and y in H. Then H plus the edge $\{x, y\}$ has a Hamiltonian circuit.

Ore's Theorem

Pick any nonadjacent vertices x and y in H. Then H plus the edge $\{x, y\}$ has a Hamiltonian circuit.

Since H does not, this circuit must use the edge $\{x, y\}$. Hence, it can be written as

$$x, y, a_1, a_2, \ldots, a_{n-2}, x$$

Ore's Theorem

Pick any nonadjacent vertices x and y in H. Then H plus the edge $\{x, y\}$ has a Hamiltonian circuit.

Since H does not, this circuit must use the edge $\{x, y\}$. Hence, it can be written as

$$x, y, a_1, a_2, \ldots, a_{n-2}, x$$

$$\bullet_y \;—\; \bullet_{a_1} \;—\; \bullet_{a_2} \;—\; \ldots \;—\; \bullet_{a_{i-1}} \;—\; \bullet_{a_i} \;—\; \bullet_{a_{i+1}} \;—\; \ldots \;—\; \bullet_{a_{n-2}} \;—\; \bullet_x$$

Ore's Theorem

Now, $V(H) = \{x, y, a_1, a_2, \ldots, a_{n-2}\}$. Moreover, we note that for $i > 1$,

$$\{y, a_i\} \in E(H) \Rightarrow \{x, a_{i-1}\} \notin E(H) \qquad (***)$$

Ore's Theorem

Now, $V(H) = \{x, y, a_1, a_2, \ldots, a_{n-2}\}$. Moreover, we note that for $i > 1$,

$$\{y, a_i\} \in E(H) \Rightarrow \{x, a_{i-1}\} \notin E(H) \qquad (***)$$

For if not, then

$$y, a_i, a_{i+1}, \ldots, a_{n-2}, x, a_{i-1}, a_{i-2}, \ldots, a_1, y$$

is a Hamiltonian circuit in H, which is a contradiction. So, $(***)$ and $\{x, y\} \notin E(H)$ imply $(**)$.

Ore's Theorem

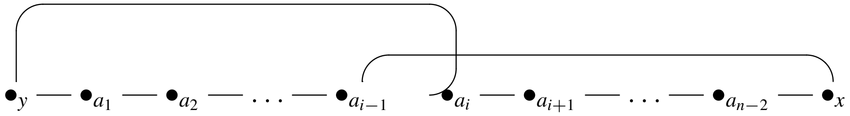
Now, $V(H) = \{x, y, a_1, a_2, \ldots, a_{n-2}\}$. Moreover, we note that for $i > 1$,

$$\{y, a_i\} \in E(H) \Rightarrow \{x, a_{i-1}\} \notin E(H) \qquad (***)$$

For if not, then

$$y, a_i, a_{i+1}, \ldots, a_{n-2}, x, a_{i-1}, a_{i-2}, \ldots, a_1, y$$

is a Hamiltonian circuit in H, which is a contradiction. So, $(***)$ and $\{x, y\} \notin E(H)$ imply $(**)$.

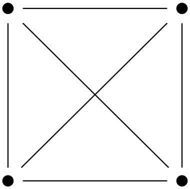


Problem With Ore’s Theorem

Not all Hamiltonian graphs have this degree property. Further, this theorem tells if such a cycle exists but not how to find the cycle.

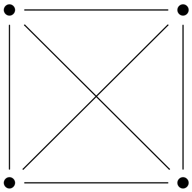
Problem With Ore's Theorem

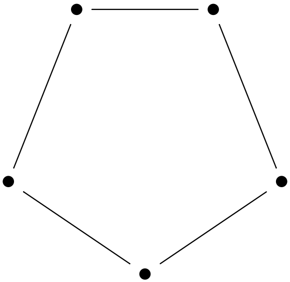
Not all Hamiltonian graphs have this degree property. Further, this theorem tells if such a cycle exists but not how to find the cycle.



Problem With Ore's Theorem

Not all Hamiltonian graphs have this degree property. Further, this theorem tells if such a cycle exists but not how to find the cycle.



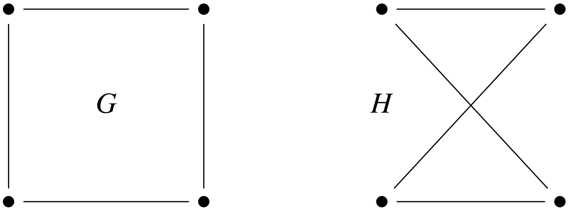


Back To Our Example

To show that Peggy knows this Hamiltonian cycle, she and Victor play several rounds of a game:

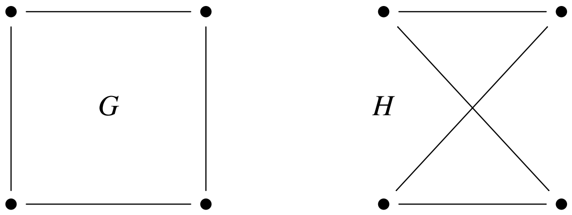
Back To Our Example

To show that Peggy knows this Hamiltonian cycle, she and Victor play several rounds of a game:
At the beginning of each round, Peggy creates H, an isomorphic graph to G.



Back To Our Example

To show that Peggy knows this Hamiltonian cycle, she and Victor play several rounds of a game:
At the beginning of each round, Peggy creates H, an isomorphic graph to G.



Or, H differs from G in that the vertices have different names.
Since it is trivial to translate a Hamiltonian cycle between isomorphic graphs with known isomorphism, if Peggy knows a Hamiltonian cycle in G, she must also know one for H.

Peggy labels the vertices of H with random numbers and then for each edge of H she writes on a small piece of paper the two vertices incident to the edge and then puts these pieces of paper upside down on a table. The purpose of this commitment is that Peggy is not able to change H while at the same time Victor has no information about H.

Hamiltonian Protocol

Peggy labels the vertices of H with random numbers and then for each edge of H she writes on a small piece of paper the two vertices incident to the edge and then puts these pieces of paper upside down on a table. The purpose of this commitment is that Peggy is not able to change H while at the same time Victor has no information about H.

Victor then chooses one of two questions to ask Peggy. He can either ask her to show the isomorphism between H and G, or he can ask her to show a Hamiltonian cycle in H.

If Peggy is asked to show that the two graphs are isomorphic, she first uncovers all of H (by turning over all of the pieces of paper she put on the table) and then provides the vertex translations that map H to G. Victor can verify that they are isomorphic.

Hamiltonian Protocol

If Peggy is asked to show that the two graphs are isomorphic, she first uncovers all of H (by turning over all of the pieces of paper she put on the table) and then provides the vertex translations that map H to G. Victor can verify that they are isomorphic.

If Peggy is asked to prove she knows a Hamiltonian cycle in H, she translates her Hamiltonian cycle in G onto H and only uncovers the edges on the Hamiltonian cycle. This is enough for Victor to verify that H is Hamiltonian.

Why This Is ZKP: Completeness

During each round, Peggy does not know which question will be asked until after giving Victor H. Therefore, in order to be able to answer both, H must be isomorphic to G and she must have a Hamiltonian cycle in H. Because only someone who knows a Hamiltonian cycle n G would have been able to answer both questions, Victor (after a sufficient number of rounds) becomes convinced that Peggy does know this information.

Why This Is ZKP: Zero-Knowledgeness

Peggy's answers do not reveal the original Hamiltonian cycle in G. Each round, Victor will only learn H isomorphic to G or a Hamiltonian cycle in H. He would need both answers for a single H to discover the cycle in G, so the information remains unknown as long as Peggy can generate a unique H every round.

Why This Is ZKP: Zero-Knowledgeness

Peggy's answers do not reveal the original Hamiltonian cycle in G. Each round, Victor will only learn H isomorphic to G or a Hamiltonian cycle in H. He would need both answers for a single H to discover the cycle in G, so the information remains unknown as long as Peggy can generate a unique H every round.

If Peggy does not know a Hamiltonian cycle in G, but somehow knew in advance what Victor would ask to see in each round then she could cheat.

Why This Is ZKP: Zero-Knowledgeness

Peggy's answers do not reveal the original Hamiltonian cycle in G. Each round, Victor will only learn H isomorphic to G or a Hamiltonian cycle in H. He would need both answers for a single H to discover the cycle in G, so the information remains unknown as long as Peggy can generate a unique H every round.

If Peggy does not know a Hamiltonian cycle in G, but somehow knew in advance what Victor would ask to see in each round then she could cheat.

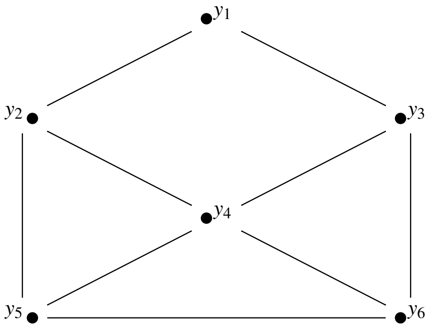
For example, if Peggy knew ahead of time that Victor would ask to see a Hamiltonian cycle in H, she could generate a Hamiltonian cycle in an unrelated graph. Similarly, if Peggy knew in advance that Victor would ask to see an isomorphism, then she would simply generate an isomorphic graph H (in which she also does not know a Hamiltonian cycle). Victor could simulate the protocol by himself (without Peggy) because she knows what he will ask to see. Therefore, Victor gains no information about the Hamiltonian cycle in G from the information revealed in each round.

Why Is This ZKP: Soundness

If Peggy does not know the information, she can guess which question Victor will ask and generate either a graph isomorphic to G or a Hamiltonian cycle for an unrelated graph, but since she cannot do both without this guesswork, her chance of fooling Victor is 2^{-n}, where n is the number of rounds. For all realistic purposes, it is infeasibly difficult to defeat a zero-knowledge proof with a reasonable number of rounds this way,

Graph Coloring Example

Peggy wants to convince Victor that a particular graph G, known to both of them, is 3-colorable and that Peggy knows such a coloring, without revealing to Victor any information about how the coloring looks.



Graph Coloring Protocol

Peggy colors the graph $G = (V, E)$ with colors (red, blue, green) and she performs with Victor $|E|^2$ times the following interactions, where $v_1, \ldots, v_n \in V(G)$.

Graph Coloring Protocol

Peggy colors the graph $G = (V, E)$ with colors (red, blue, green) and she performs with Victor $|E|^2$ times the following interactions, where $v_1, \ldots, v_n \in V(G)$.

Peggy chooses a random permutation of colors, recolors G, and encrypts for $i = 1, 2, \ldots, n$, the color c_i of vertex v_i by the encryption procedure e_i for each different i.

Graph Coloring Protocol

Peggy colors the graph $G = (V, E)$ with colors (red, blue, green) and she performs with Victor $|E|^2$ times the following interactions, where $v_1, \dots, v_n \in V(G)$.

Peggy chooses a random permutation of colors, recolors G, and encrypts for $i = 1, 2, \dots, n$, the color c_i of vertex v_i by the encryption procedure e_i for each different i.

Peggy then removes colors from the vertices, labels the i^{th} vertex of G with the cryptotext $y_i = e_i(c_i)$, and designs the following table.

1	red	e_1	$e_1(red) = y_1$
2	green	e_2	$e_2(green) = y_2$
3	blue	e_3	$e_3(blue) = y_3$
4	red	e_4	$e_4(red) = y_4$
5	blue	e_5	$e_5(blue) = y_5$
6	green	e_6	$e_6(green) = y_6$

Graph Coloring Protocol

Peggy colors the graph $G = (V, E)$ with colors (red, blue, green) and she performs with Victor $|E|^2$ times the following interactions, where $v_1, \ldots, v_n \in V(G)$.

Peggy chooses a random permutation of colors, recolors G, and encrypts for $i = 1, 2, \ldots, n$, the color c_i of vertex v_i by the encryption procedure e_i for each different i.

Peggy then removes colors from the vertices, labels the i^{th} vertex of G with the cryptotext $y_i = e_i(c_i)$, and designs the following table.

1	red	e_1	$e_1(red) = y_1$
2	green	e_2	$e_2(green) = y_2$
3	blue	e_3	$e_3(blue) = y_3$
4	red	e_4	$e_4(red) = y_4$
5	blue	e_5	$e_5(blue) = y_5$
6	green	e_6	$e_6(green) = y_6$

Peggy finally shows Victor the graph with vertices labeled by cryptotexts.

Graph Coloring Protocol

Victor chooses and edge and asks Peggy to show him the coloring of the corresponding vertices.

Graph Coloring Protocol

Victor chooses and edge and asks Peggy to show him the coloring of the corresponding vertices.

Peggy shows Victor entries of the table corresponding to the vertices of the chosen edge.

Graph Coloring Protocol

Victor chooses and edge and asks Peggy to show him the coloring of the corresponding vertices.

Peggy shows Victor entries of the table corresponding to the vertices of the chosen edge.

Victor performs encryptions to verify that the nodes have the colors as shown.

Coin Flipping By Phone

Alice and Bob got divorced and they do not trust each other any longer. They want to decide, communicating by phone only, who gets the car.

Alice and Bob got divorced and they do not trust each other any longer. They want to decide, communicating by phone only, who gets the car.

Protocol 1: Alice sends Bob messages *head* and *tail* encrypted by a one-way function f. Bob guesses which one of them is the encryption of *head*. Alice tells Bob whether his guess was correct. If Bob does not believe her, Alice sends f to Bob.

Coin Flipping By Phone

Protocol 2: Alice chooses two large primes p, q and sends $n = pq$ to Bob, keeping p and q secret. Bob chooses randomly an integer $x \in \{1, 2, \ldots, \frac{n-1}{2}\}$, sends Alice $y \equiv x^2 \pmod{n}$ and tells Alice 'If you guess x correctly, you get the car'.

Coin Flipping By Phone

Protocol 2: Alice chooses two large primes p, q and sends $n = pq$ to Bob, keeping p and q secret. Bob chooses randomly an integer $x \in \{1, 2, \ldots, \frac{n-1}{2}\}$, sends Alice $y \equiv x^2 \pmod{n}$ and tells Alice 'If you guess x correctly, you get the car'.

Alice computes four square roots $(x_1, n - x_1)$ and $(x_2, n - x_2)$ of x.

Coin Flipping By Phone

Protocol 2: Alice chooses two large primes p, q and sends $n = pq$ to Bob, keeping p and q secret. Bob chooses randomly an integer $x \in \{1, 2, \ldots, \frac{n-1}{2}\}$, sends Alice $y \equiv x^2 \pmod{n}$ and tells Alice 'If you guess x correctly, you get the car'.

Alice computes four square roots $(x_1, n - x_1)$ and $(x_2, n - x_2)$ of x.

Let

$$x_1' = min\{x_1, n - x_1\},\ x_2' = min\{x_2, n - x_2\}$$

Since $x \in \{1, \ldots, \frac{n-1}{2}\}$, either $x = x_1'$ or $x = x_2'$.

Coin Flipping By Phone

Protocol 2: Alice chooses two large primes p, q and sends $n = pq$ to Bob, keeping p and q secret. Bob chooses randomly an integer $x \in \{1, 2, \ldots, \frac{n-1}{2}\}$, sends Alice $y \equiv x^2 \pmod{n}$ and tells Alice 'If you guess x correctly, you get the car'.

Alice computes four square roots $(x_1, n - x_1)$ and $(x_2, n - x_2)$ of x.

Let

$$x_1' = min\{x_1, n - x_1\},\; x_2' = min\{x_2, n - x_2\}$$

Since $x \in \{1, \ldots, \frac{n-1}{2}\}$, either $x = x_1'$ or $x = x_2'$.

Alice then guesses $x = x_1'$ or $x = x_2'$ and tells Bob of her choice (for example by reporting the position and value of the leftmost bit in which x_1' and x_2' differ).

Coin Flipping By Phone

Protocol 2: Alice chooses two large primes p, q and sends $n = pq$ to Bob, keeping p and q secret. Bob chooses randomly an integer $x \in \{1, 2, \ldots, \frac{n-1}{2}\}$, sends Alice $y \equiv x^2 \pmod{n}$ and tells Alice 'If you guess x correctly, you get the car'.

Alice computes four square roots $(x_1, n - x_1)$ and $(x_2, n - x_2)$ of x.

Let

$$x_1' = min\{x_1, n - x_1\},\ x_2' = min\{x_2, n - x_2\}$$

Since $x \in \{1, \ldots, \frac{n-1}{2}\}$, either $x = x_1'$ or $x = x_2'$.

Alice then guesses $x = x_1'$ or $x = x_2'$ and tells Bob of her choice (for example by reporting the position and value of the leftmost bit in which x_1' and x_2' differ).

Bob tells Alice whether her guess was correct, and then later if necessary, Alice reveals p and q and Bob reveals x.

A More Mathematical Example

Let $n = pq$ for large primes p and q. Assume $p, q \equiv 3 \pmod 4$.

A More Mathematical Example

Let $n = pq$ for large primes p and q. Assume $p, q \equiv 3 \pmod 4$.

Suppose Peggy knows the factorization $n = pq$.

- Victor chooses a random integer x and sends $x^4 \pmod n$ to Peggy.

A More Mathematical Example

Let $n = pq$ for large primes p and q. Assume $p, q \equiv 3 \pmod 4$.

Suppose Peggy knows the factorization $n = pq$.

- Victor chooses a random integer x and sends $x^4 \pmod n$ to Peggy.
- Peggy computes the principal square root

$$y_1 = (x^4)^{\frac{p+1}{2}}$$

of $x^4 \pmod p$, and principal square root

$$y_2 = (x^4)^{\frac{q+1}{2}}$$

of $x^4 \pmod q$, and uses the Chinese Remainder Theorem (and the Euclidean Algorithm) to compute y so that $y \equiv y_1 \pmod p$ and $y \equiv y_2 \pmod q$. Peggy sends this value back to Victor.

A More Mathematical Example

This formula for square root modulo primes congruent to $3 \pmod 4$ returns the principal square root, which is itself a square. For a prime $p \equiv 3 \pmod 4$ and for $a \equiv b^2 \pmod 4$, the two square roots are $\pm b$, and exactly one of $\pm b$ is itself a square since -1 is a nonsquare and $\mathbb{Z}/p^*$ is cyclic.

A More Mathematical Example

This formula for square root modulo primes congruent to $3 \pmod 4$ returns the principal square root, which is itself a square. For a prime $p \equiv 3 \pmod 4$ and for $a \equiv b^2 \pmod 4$, the two square roots are $\pm b$, and exactly one of $\pm b$ is itself a square since -1 is a nonsquare and $\mathbb{Z}/p^*$ is cyclic.

Since Victor can already compute x^2, Peggy has certainly imported no information to Victor.

A More Mathematical Example

Victor should be convinced that there is no other way for Peggy to have found the square root than by knowing the factors p and q because in any case, being able to take a square roots $\pmod{n}$ gives a probabilistic algorithm for factoring n (when n is in the special form $n = pq$ with distinct primes p and q) as following:

A More Mathematical Example

Victor should be convinced that there is no other way for Peggy to have found the square root than by knowing the factors p and q because in any case, being able to take a square roots $\pmod{n}$ gives a probabilistic algorithm for factoring n (when n is in the special form $n = pq$ with distinct primes p and q) as following:

If we have an <u>oracle</u> (meaning, some otherwise unexplained mechanism) which computes square roots modulo n), we repeatedly do the following: pick a random number x, compute $x^2 \pmod{n}$ and feed the result to the oracle, which returns a square root of $x^2 \pmod{n}$. Since these are exactly two square roots of any nonzero square modulo a prime (by the Chinese Remainder Theorem), there are exactly 4 square roots of any square modulo $n = pq$ and $\pm x$ is just two of them. Let the other two be $\pm x'$.

A More Mathematical Example

Assuming that the original x was really chosen randomly, the probability is $\frac{1}{2}$ that the oracle will return x' as y. If so, then n does not divide either of $x \pm y$ but nevertheless n divides $x^2 = y^2$ (since $x^2 \equiv y^2 \pmod{n}$). So, p divides one of $x \pm y$ and q divides the other one. therefore, $((x - y), n)$ is either p or q which could easily be computed. Since the oracle can be called repeatedly, at each invocation there is probability $\frac{1}{2}$ that a factorization will be obtained. So the probability that after l invocations we don't obtain s factorization is $\left(\frac{1}{2}\right)^l$. This goes to 0 quickly as l goes to ∞, which we construe as an indication that we will obtain a factorization in reasonable time.

Fiat and Shamir

The first publication on the practical application of ZKP was by Fiat and Shamir in 1986. It described a protocol for the identification of one entity by another and a procedure for generating digital signatures.

Fiat and Shamir

The first publication on the practical application of ZKP was by Fiat and Shamir in 1986. It described a protocol for the identification of one entity by another and a procedure for generating digital signatures.

A system is organized around a central body/entity, which provides each member of a certain group with secret personal information. This secret information is generated from a large number n, which is the product of two large numbers p and q, as in the RSA system.

Fiat and Shamir

The first publication on the practical application of ZKP was by Fiat and Shamir in 1986. It described a protocol for the identification of one entity by another and a procedure for generating digital signatures.

A system is organized around a central body/entity, which provides each member of a certain group with secret personal information. This secret information is generated from a large number n, which is the product of two large numbers p and q, as in the RSA system.

The value of n is made public, but the two numbers p and q are only known to a centrally located third party. This central entity will generate an identification sequence I for each member of the group, which contains all relevant information, such as name, address, etc.

Fiat and Shamir

The first publication on the practical application of ZKP was by Fiat and Shamir in 1986. It described a protocol for the identification of one entity by another and a procedure for generating digital signatures.

A system is organized around a central body/entity, which provides each member of a certain group with secret personal information. This secret information is generated from a large number n, which is the product of two large numbers p and q, as in the RSA system.

The value of n is made public, but the two numbers p and q are only known to a centrally located third party. This central entity will generate an identification sequence I for each member of the group, which contains all relevant information, such as name, address, etc.

The central entity can calculate k different integers $v_j = f(I, c_j)$, where c_j is an integer and v_j has integer value for which $u_j^2 \equiv v_j \pmod{n}$ with u_j between 0 and $n-1$.

Fiat and Shamir

The function $f(.,.)$, which is public, can be realized with, for example, the triple-DES. The k value can be obtained by repeatedly calculating different values of $f(I, c_j)$, until k values for v_j have been found, which satisfy the above condition.

Fiat and Shamir

The function $f(.,.)$, which is public, can be realized with, for example, the triple-DES. The k value can be obtained by repeatedly calculating different values of $f(I, c_j)$, until k values for v_j have been found, which satisfy the above condition.

The central entity will then calculate the smallest roots of $v_j^{-1} \pmod{n}$, for each of the k values of v_j. These are denoted by s_j.

$$s_j^2 v_j \equiv 1 \pmod{n}$$

The function $f(.,.)$, which is public, can be realized with, for example, the triple-DES. The k value can be obtained by repeatedly calculating different values of $f(I, c_j)$, until k values for v_j have been found, which satisfy the above condition.

The central entity will then calculate the smallest roots of $v_j^{-1} \pmod{n}$, for each of the k values of v_j. These are denoted by s_j.

$$s_j^2 v_j \equiv 1 \pmod{n}$$

We should note that the calculation of these roots requires information about the factors p and q. Since we may assume the factorization of a large number is computationally infeasible, as in RSA, no one other than the central entity can calculate the values of s_j. Thus, these values are used as the secret values with which others can ascertain another person's identity. The actual values of s_j will provide no information with respect to p and q. Therefore there is nothing to prevent n being shared by more than one member.

The function $f(.,.)$, which is public, can be realized with, for example, the triple-DES. The k value can be obtained by repeatedly calculating different values of $f(I, c_j)$, until k values for v_j have been found, which satisfy the above condition.

The central entity will then calculate the smallest roots of $v_j^{-1} \pmod{n}$, for each of the k values of v_j. These are denoted by s_j.

$$s_j^2 v_j \equiv 1 \pmod{n}$$

We should note that the calculation of these roots requires information about the factors p and q. Since we may assume the factorization of a large number is computationally infeasible, as in RSA, no one other than the central entity can calculate the values of s_j. Thus, these values are used as the secret values with which others can ascertain another person's identity. The actual values of s_j will provide no information with respect to p and q. Therefore there is nothing to prevent n being shared by more than one member.

After this initialization phase, the ZKP can commence.

Using This ZKP

Let us suppose Bob wishes to ascertain the identity of Alice. Alice must therefore prove in some manner that she has access to the secret values $s_1, \ldots, s_k$ without actually revealing any of these values. The protocol requires the following:

Using This ZKP

Let us suppose Bob wishes to ascertain the identity of Alice. Alice must therefore prove in some manner that she has access to the secret values $s_1, \ldots, s_k$ without actually revealing any of these values. The protocol requires the following:

Alice sends I and the values $c_1, \ldots, c_k$ to Bob.

Let us suppose Bob wishes to ascertain the identity of Alice. Alice must therefore prove in some manner that she has access to the secret values $s_1, \ldots, s_k$ without actually revealing any of these values. The protocol requires the following:

Alice sends I and the values $c_1, \ldots, c_k$ to Bob.

Bob generates $v_j = f(I, c_j)$ for $j = 1, \ldots, k$.

Using This ZKP

Let us suppose Bob wishes to ascertain the identity of Alice. Alice must therefore prove in some manner that she has access to the secret values $s_1, \ldots, s_k$ without actually revealing any of these values. The protocol requires the following:

Alice sends I and the values $c_1, \ldots, c_k$ to Bob.

Bob generates $v_j = f(I, c_j)$ for $j = 1, \ldots, k$.

Alice selects a random number r_i between 0 and $n - 1$ and sends

$$x_i \equiv r_i^2 \pmod{n}$$

to Bob.

Using This ZKP

Let us suppose Bob wishes to ascertain the identity of Alice. Alice must therefore prove in some manner that she has access to the secret values $s_1, \ldots, s_k$ without actually revealing any of these values. The protocol requires the following:

Alice sends I and the values $c_1, \ldots, c_k$ to Bob.

Bob generates $v_j = f(I, c_j)$ for $j = 1, \ldots, k$.

Alice selects a random number r_i between 0 and $n - 1$ and sends

$$x_i \equiv r_i^2 \pmod{n}$$

to Bob.

Bob generates a binary random vector $(t_{i1}, \ldots, t_{ik})$ and sends this to Alice.

Using This ZKP

Alice sends y_i to Bob for which

$$y_i \equiv r_i \prod_j s_j^{t_{ij}} \pmod{n}$$

Using This ZKP

Alice sends y_i to Bob for which

$$y_i \equiv r_i \prod_j s_j^{t_{ij}} \pmod{n}$$

Bob computes

$$z_i \equiv y_i^2 \prod_j v_j^{t_{ij}} \pmod{n}$$

and checks if $z_1 = x_i$.

Using This ZKP

Alice sends y_i to Bob for which

$$y_i \equiv r_i \prod_j s_j^{t_{ij}} \pmod{n}$$

Bob computes

$$z_i \equiv y_i^2 \prod_j v_j^{t_{ij}} \pmod{n}$$

and checks if $z_1 = x_i$.

Steps 3-6 are repeated for $i = 1 \ldots, t$.

Using This ZKP

Alice sends y_i to Bob for which

$$y_i \equiv r_i \prod_j s_j^{t_{ij}} \pmod{n}$$

Bob computes

$$z_i \equiv y_i^2 \prod_j v_j^{t_{ij}} \pmod{n}$$

and checks if $z_1 = x_i$.

Steps 3-6 are repeated for $i = 1 \ldots, t$.

Bob will accept that the person claiming to be Alice is really Alice if all t checks out successfully.

Using This ZKP

If all proceeds according to plan, then for all i, $z_i = x_i$. By combining the equations in steps 3,4 and 6, it follows that

$$z_i \equiv r_i^2 \prod_j s_j^{2t_{ij}} \prod_j v_j^{t_{ij}} \pmod{n}$$

Using This ZKP

If all proceeds according to plan, then for all i, $z_i = x_i$. By combining the equations in steps 3,4 and 6, it follows that

$$\begin{aligned} z_i &\equiv r_i^2 \prod_j s_j^{2t_{ij}} \prod_j v_j^{t_{ij}} \pmod{n} \\ &\equiv r_i^2 \prod_j \left(s_j^2 \cdot v_j\right)^{t_{ij}} \pmod{n} \end{aligned}$$

Using This ZKP

If all proceeds according to plan, then for all i, $z_i = x_i$. By combining the equations in steps 3,4 and 6, it follows that

$$\begin{aligned} z_i &\equiv r_i^2 \prod_j s_j^{2t_{ij}} \prod_j v_j^{t_{ij}} \pmod{n} \\ &\equiv r_i^2 \prod_j \left(s_j^2 \cdot v_j\right)^{t_{ij}} \pmod{n} \\ &\equiv r_i^2 \pmod{n} \end{aligned}$$

Using This ZKP

If all proceeds according to plan, then for all i, $z_i = x_i$. By combining the equations in steps 3,4 and 6, it follows that

$$\begin{aligned} z_i &\equiv r_i^2 \prod_j s_j^{2t_{ij}} \prod_j v_j^{t_{ij}} \pmod{n} \\ &\equiv r_i^2 \prod_j \left(s_j^2 \cdot v_j\right)^{t_{ij}} \pmod{n} \\ &\equiv r_i^2 \pmod{n} \\ &\equiv x_i \end{aligned}$$

Beware Of Charles ...

Suppose Charles wishes to pretend he is Alice. Since he knows neither the values of s_j, not the values of r_i, he cannot calculate y_i in step 5.

Beware Of Charles ...

Suppose Charles wishes to pretend he is Alice. Since he knows neither the values of s_j, not the values of r_i, he cannot calculate y_i in step 5.

Although Charles knows the values of v_j and n, this alone is still not sufficient for calculating s_j. The reason for this is that the square of the value can be calculated without any problem for modulo calculations, but the calculation of the root is computationally infeasible for n sufficiently large.

Beware Of Charles ...

Suppose Charles wishes to pretend he is Alice. Since he knows neither the values of s_j, not the values of r_i, he cannot calculate y_i in step 5.

Although Charles knows the values of v_j and n, this alone is still not sufficient for calculating s_j. The reason for this is that the square of the value can be calculated without any problem for modulo calculations, but the calculation of the root is computationally infeasible for n sufficiently large.

This also implies to the computation of r_i from x_i.

Beware Of Charles ...

Suppose Charles wishes to pretend he is Alice. Since he knows neither the values of s_j, not the values of r_i, he cannot calculate y_i in step 5.

Although Charles knows the values of v_j and n, this alone is still not sufficient for calculating s_j. The reason for this is that the square of the value can be calculated without any problem for modulo calculations, but the calculation of the root is computationally infeasible for n sufficiently large.

This also implies to the computation of r_i from x_i.

Furthermore, Charles will also find it computationally infeasible to deduce the values of y_i from the equation in step 6; Charles knows the values of $z_i s_i$ since these must be equal to those of x_i, and the values of v_j and the elements t_{ij} of the random vector, but he will still be defeated by the problem of finding a square root of a modulo value.

Why Is This A ZKP?

Consider the simplest case where $k = t = 1$. Alice has one secret key s and must demonstrate this to Bob.

Why Is This A ZKP?

Consider the simplest case where $k = t = 1$. Alice has one secret key s and must demonstrate this to Bob.

According to protocol, she generates a random number r, calculates x and sends this value to Bob. Bob will return exactly one bit: 0 or 1.

Why Is This A ZKP?

Consider the simplest case where $k = t = 1$. Alice has one secret key s and must demonstrate this to Bob.

According to protocol, she generates a random number r, calculates x and sends this value to Bob. Bob will return exactly one bit: 0 or 1.

If he returns a zero, then Alice will respond by sending r. If he returns a 1, Alice will respond by sending the product rs.

Why Is This A ZKP?

Consider the simplest case where $k = t = 1$. Alice has one secret key s and must demonstrate this to Bob.

According to protocol, she generates a random number r, calculates x and sends this value to Bob. Bob will return exactly one bit: 0 or 1.

If he returns a zero, then Alice will respond by sending r. If he returns a 1, Alice will respond by sending the product rs.

Bob can verify that Alice has responded correctly by using his knowledge of the values of I and x. If Alice returns r, Bob will not learn anything of s, since r is a random number. And if Alice returns rs, he will still learn nothing of s since in this scenario, r is not known.