Interesting Primitives and Applications Of Cryptography

O.S.L.Bhavana

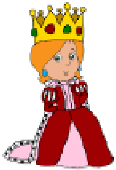
Advisor: Dr. Bhavana Kanukurthi
Cryptography, Security and Privacy lab
CSA, IISC

July 5, 2017

- Bit Commitment Schemes
- Zero knowledge proofs

Coin flipping

Alice



Makes her call x

Reveals her call to Bob

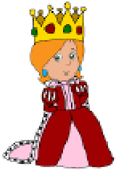
Bob

Tosses a coin in Alice's presence

Declares who is the winner

Coin flipping over distance

Alice



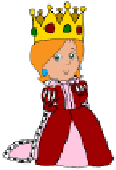
Bob

Makes her call x

Can Alice still reveal her call to Bob before coin toss?

Coin flipping over distance

Alice



Bob

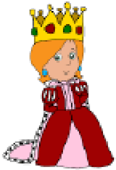
Makes her call x

Can Alice reveal her call to Bob before coin toss? No.

Bob may report the toss wrongly.

Coin flipping over distance

Alice



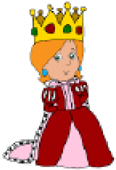
Bob

Tosses a coin

Can Bob reveal the coin toss without knowing Alice's call?

Coin flipping over distance

Alice



Bob

Tosses a coin

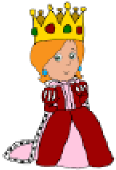
Can Bob reveal the coin toss without knowing Alice's call? No.

Alice may modify her call.

Coin flipping over distance

Trusted party

Alice



Bob

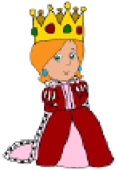
Coin flipping over distance

Trusted party

Alice

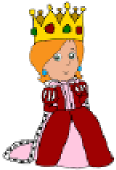
But I don't exist!!
Figure it out yourself!

Bob



Coin flipping over distance by commitment

Alice **Bob**



makes her call x
Locks her call in a box

toss result

tosses a coin

 and x

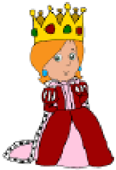
opens the box and
checks Alice's call

If Alice's call is 'x', Accept and
Declares who is the winner

Else Reject

Coin flipping over distance by commitment

Alice



Bob

makes her call x

Locks her call in a box  = *com*

→

toss result

←

tosses a coin

 = *dec*

→

opens the box and checks Alice's call

Declares who is the winner

Mathematically..

Alice

Makes her call x

$(dec, com) = Commit(x)$

$\xrightarrow{com}$

Bob

coin toss results in b

$\xleftarrow{b}$

$\xrightarrow{x, dec}$

If $Decommit(x, dec, com) =$ Accept

Announces who is the winner

Else Reject

Properties of Bit-Commitment schemes

Properties of Bit-Commitment schemes

Sender's security: Hiding

The receiver should not know whether the committed bit is 0 or 1, on seeing the commitment *com*.

Properties of Bit-Commitment schemes

Sender's security: Hiding
The receiver should not know whether the committed bit is 0 or 1, on seeing the commitment *com*.
Receiver's security: Binding
After committing to 0, sender shouldn't be able to generate dec' that decommits *com* to 1 and vice-versa.

Building blocks

Adversary

- Information theoretic: Has unbounded computing power
  - Example: Can easily run exponential time algorithms.

Building blocks

Adversary

- Information theoretic: Has unbounded computing power
  - Example: Can easily run exponential time algorithms.
- Computational: Has limited computing power
  - Example: Can only run polynomial time algorithms.
  - Probabilistic Poly time Adversary is a computational adversary that can flip coins.

Building blocks

Adversary

- Information theoretic: Has unbounded computing power
  - Example: Can easily run exponential time algorithms.
- Computational: Has limited computing power
  - Example: Can only run polynomial time algorithms.
  - Probabilistic Poly time Adversary is a computational adversary that can flip coins.

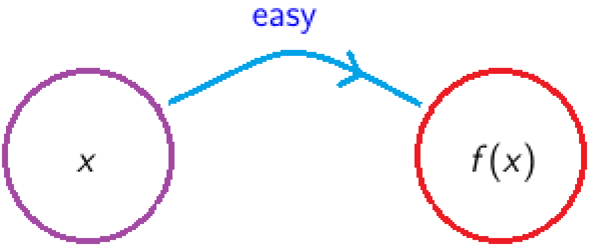
Adversary is an algorithm!!!

- Information theoretic: Has unbounded computing power
  - Example: Can easily run exponential time algorithms.
- Computational: Has limited computing power
  - Example: Can only run polynomial time algorithms.
  - Probabilistic Poly time Adversary is a computational adversary that can flip coins.

Adversary is an algorithm!!!

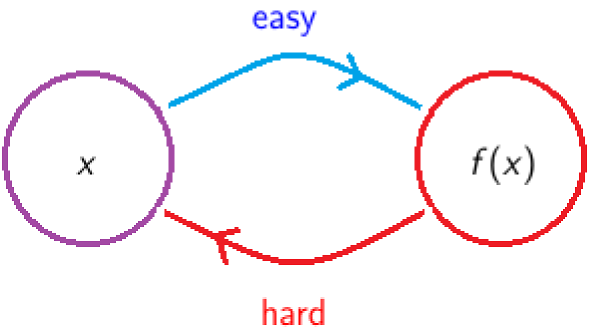
Building blocks

One way function



Building blocks

One way function



Building blocks

One way function

f

Adversary

Challenger

$x \in_R \{0,1\}^n$

$\xleftarrow{\quad f(x) \quad}$

$\xrightarrow{\quad x' \quad}$

If $f(x') = f(x)$

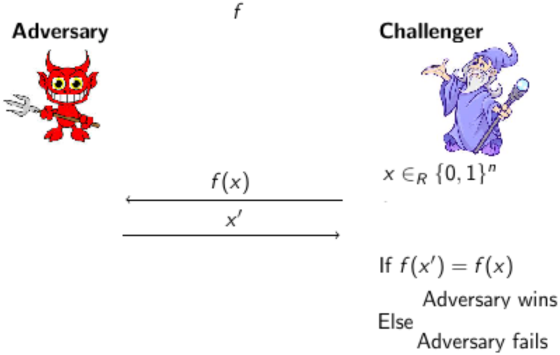
Adversary wins

Else

Adversary fails

Building blocks

One way function



A function $f : \{0,1\}^n \to \{0,1\}^*$ is one-way if

- Given x, f(x) is efficiently computable.
- Pr[Adversary wins in OWF game] is negligible.

- Why randomly chosen x ?

Building blocks

One way function

- Why randomly chosen x ?
- Do OWF's exist information theoretically?

Building blocks

One way function

- Why randomly chosen x ?
- Do OWF's exist information theoretically?

Building blocks

One way function

- Why randomly chosen x ?
- Do OWF's exist information theoretically?No

Building blocks

One way function

- Why randomly chosen x ?
- Do OWF's exist information theoretically?No
- Proving existence of a OWF is an open problem.

Building blocks

Hard core predicate

f , hcp

Adversary

Challenger

$x \in_R \{0,1\}^n$

$\xleftarrow{\quad f(x) \quad}$

$\xrightarrow{\quad \mathrm{b}' \quad}$

If $hcp(x) = \mathrm{b}'$

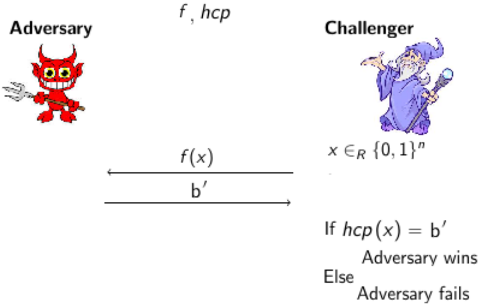
Adversary wins

Else

Adversary fails

Building blocks

Hard core predicate

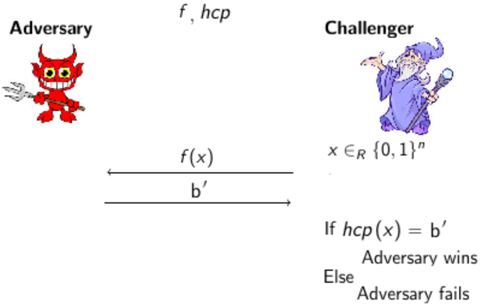


A boolean function $hcp : \{0,1\}^n \rightarrow \{0,1\}$, is hard core predicate of a function $f : \{0,1\}^n \rightarrow \{0,1\}^*$, if

- Given x, hcp(x) is efficiently computable.
- Pr[Adversary wins in HCP game] is $\frac{1}{2}+$ negligible.

Building blocks

Hard core predicate



A boolean function $hcp : \{0,1\}^n \to \{0,1\}$, is hard core predicate of a function $f : \{0,1\}^n \to \{0,1\}^*$, if

- Given x, hcp(x) is efficiently computable.
- Pr[Adversary wins in HCP game] is $\frac{1}{2}+$ negligible.

Every OWF has a HCP.

Building blocks

One way permutation

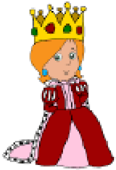
$f : \{0,1\}^n \to \{0,1\}^n$ is a OWP if f is a

- Permutation
- One way function

Constructing commitments from OWP

f is a OWP.

Sender



Receiver

$y \in_R \{0,1\}^n$;

Commit phase $\xrightarrow{com = (f(y), x \oplus hcp(y))}$

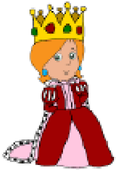
Parse com as (a, b)
If $a == f(\text{dec})$ and
$b \oplus hcp(\text{dec}) = x$
Accept
Else **Reject**

Decommit Phase $\xrightarrow{x,\ \text{dec} = y}$

Constructing commitments from OWP

f is a OWP.

Sender



Receiver

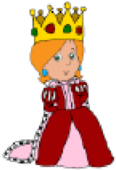
$y \in_R \{0,1\}^n$;

Commit phase $\xrightarrow{com = (f(y), x \oplus hcp(y))}$

Hiding: Having seen com,can Bob know whether x= 0 or x=1?

Constructing commitments from OWP

Sender



$$\xrightarrow{f(y), x \oplus hcp(y)}$$

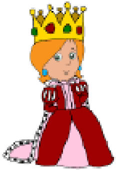
Receiver

Hiding holds as $hcp(y)$ is unpredictable.

Constructing commitments from OWP

f is a OWP.

Sender



Receiver

$y \in_R \{0,1\}^n$;

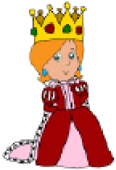
Commit phase $\xrightarrow{com = (f(y), x \oplus hcp(y))}$

Decommit Phase $\xrightarrow{x', dec' = y'}$

Binding: Can Alice commit to x, and send dec' that decommits to 1-x?

Constructing commitments from OWP

Sender



$$f(y), x \oplus hcp(y)$$

$$x', y'$$

Receiver

Binding holds as f is a OWP

If y' ≠ y then f(y') ≠ f(y)
Therefore receiver rejects x'

Bit commitments : Summary

- Motivation
- Building blocks
- An explicit construction

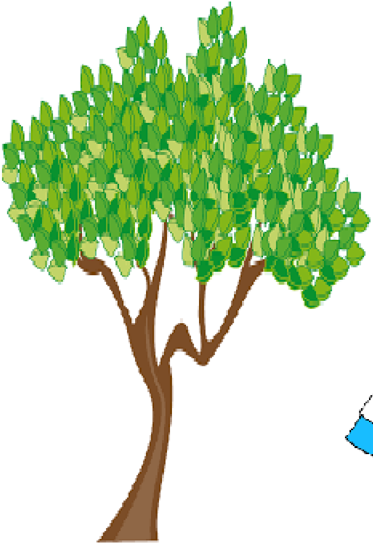
Zero Knowledge Proofs

Zero Knowledge Proofs

- Motivation
- Properties
- ZKP for graph coloring

ZKP: Motivation

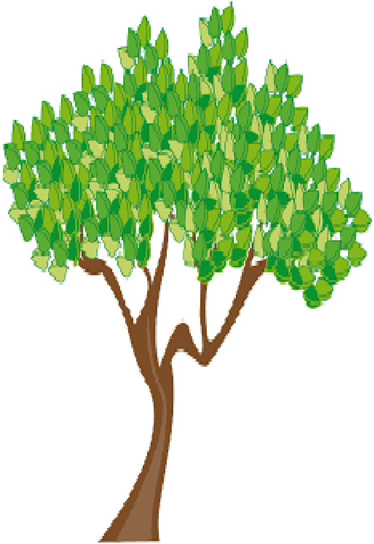
I can count the
number of leaves.



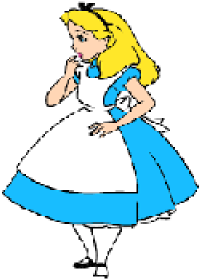
Let me test

ZKP: Motivation

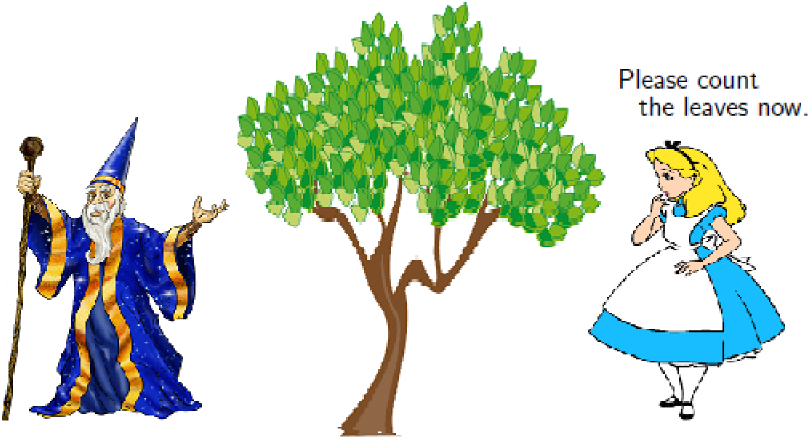
ZKP: Motivation



Tosses a coin.
Plucks a leaf if heads.



ZKP: Motivation



Prob[cheating wizard fails] $= \frac{1}{2}$

Important application of ZKP

- Cloud computation

Properties of ZKP

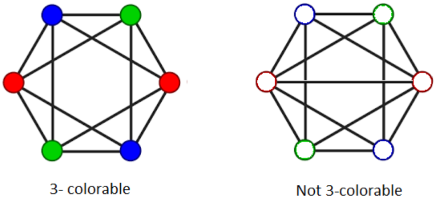
Properties of ZKP

Verifier's security: Soundness
The prover should not be able to prove verifier false statements

Properties of ZKP

Verifier's security: Soundness
The prover should not be able to prove verifier false statements
Prover's security: Zero knowledge
The verifier should not learn any additional information other than the statement being proved.

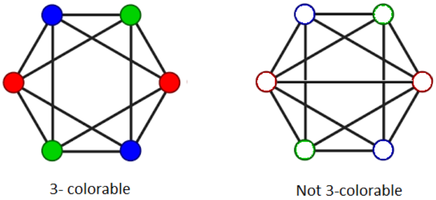
Graph 3-Coloring



Given graph $G = (V, E)$, can we assign each vertex a color (one of the three colors) such that no two adjacent vertices have same color?

- For a graph G that is 3-colorable, the witness is the color assignment.

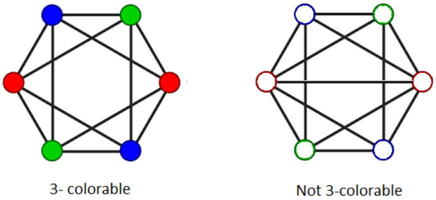
Graph 3-Coloring



Given graph $G = (V, E)$, can we assign each vertex a color (one of the three colors) such that no two adjacent vertices have same color?

- For a graph G that is 3-colorable, the witness is the color assignment.
- Graph 3-Coloring is an NP-Complete problem.

Graph 3-Coloring



Given graph $G = (V, E)$, can we assign each vertex a color (one of the three colors) such that no two adjacent vertices have same color?

- For a graph G that is 3-colorable, the witness is the color assignment.
- Graph 3-Coloring is an NP-Complete problem.
- Therefore, no known algorithm with polynomial running time can decide whether a graph G is 3-colorable or not?

ZKP for Graph 3-Coloring

Prover **G=(V,E) is 3-colorable** **Verifier**

3-coloring C

ZKP for Graph 3-Coloring

Prover **G=(V,E) is 3-colorable** **Verifier**

3-coloring C

Chooses random permutation of 3-colors

Re-assign colors based on permutation

ZKP for Graph 3-Coloring

Prover **G=(V,E) is 3-colorable** **Verifier**

3-coloring C

Chooses random permutation of 3-colors

Re-assign colors based on permutation

Commit to re-assigned colors of all vertices

$(dec_i, com_i) = Commit(color(V_i))$

$\xrightarrow{(com_1, .., com_n)}$

ZKP for Graph 3-Coloring

Prover	G=(V,E) is 3-colorable	Verifier
3-coloring C		
Chooses random permutation of 3-colors		Chooses an edge $(u, v) \in_R E$
Re-assign colors based on permutation		
Commit to re-assigned colors of all vertices		
$(dec_i, com_i) = Commit(color(V_i))$	$\xrightarrow{(com_1, .., com_n)}$	
	$\xleftarrow{\text{reveal colors of } u, v}$	

ZKP for Graph 3-Coloring

Prover | **G=(V,E) is 3-colorable** | **Verifier**

3-coloring C

Chooses random permutation of 3-colors

Re-assign colors based on permutation

Commit to re-assigned colors of all vertices

$(dec_i, com_i) = Commit(color(V_i))$

Chooses an edge$(u, v) \in_R E$

$\xrightarrow{(com_1, .., com_n)}$

$\xleftarrow{\text{reveal colors of } u, v}$

$\xrightarrow{color_u, color_v, dec_u, dec_v}$

If $Decommit(color_u, com_u, dec_u) =$ Accept

and $Decommit(color_v, com_v, dec_v) =$ Accept

and $color_u \neq color_v$;

Accept that G is 3-colorable

Else reject

ZKP for Graph 3-Coloring

Soundness

- If G isn't 3-colorable, there exists an edge (u, v) such that $color(u) = color(v)$

ZKP for Graph 3-Coloring

Soundness

- If G isn't 3-colorable, there exists an edge (u, v) such that $color(u) = color(v)$
- Verifier would reject the graph if he chose edge (u, v)

ZKP for Graph 3-Coloring

Soundness

- If G isn't 3-colorable, there exists an edge (u, v) such that $color(u) = color(v)$
- Verifier would reject the graph if he chose edge (u, v)
- Pr[Verifier rejects the graph] $\geq \frac{1}{n^2}$

ZKP for Graph 3-Coloring

Soundness

- If G isn't 3-colorable, there exists an edge (u, v) such that $color(u) = color(v)$
- Verifier would reject the graph if he chose edge (u, v)
- Pr[Verifier rejects the graph] $\geq \frac{1}{n^2}$
- Pr[Verifier accepts a non 3-colorable graph] $\leq 1 - \frac{1}{n^2}$

ZKP for Graph 3-Coloring

Soundness

- If G isn't 3-colorable, there exists an edge (u, v) such that $color(u) = color(v)$
- Verifier would reject the graph if he chose edge (u, v)
- Pr[Verifier rejects the graph] $\geq \frac{1}{n^2}$
- Pr[Verifier accepts a non 3-colorable graph] $\leq 1 - \frac{1}{n^2}$
- Repeat the experiment n^3 times

ZKP for Graph 3-Coloring

Soundness

- If G isn't 3-colorable, there exists an edge (u, v) such that $color(u) = color(v)$
- Verifier would reject the graph if he chose edge (u, v)
- Pr[Verifier rejects the graph] $\geq \frac{1}{n^2}$
- Pr[Verifier accepts a non 3-colorable graph] $\leq 1 - \frac{1}{n^2}$
- Repeat the experiment n^3 times
- Pr[Verifier accepts the non 3-colorable graph in all runs]

$$\leq (1 - \frac{1}{n^2})^{n^3} \leq e^{-n}$$

ZKP for Graph 3-Coloring

Zero knowledge

- In a single run, verifier would only know colors of two vertices.

ZKP for Graph 3-Coloring

Zero knowledge

- In a single run, verifier would only know colors of two vertices.
- Colors of other vertices are hidden by commitments

ZKP for Graph 3-Coloring

Zero knowledge

- In a single run, verifier would only know colors of two vertices.
- Colors of other vertices are hidden by commitments
- In the next run the colors are randomly permuted, so the information of colors about previous run would not help

Reference

Foundations of Cryptography, Volume 1, Oded Goldreich.

Thank you

ZKP for Graph 3-coloring

G=(V,E) is 3-colorable

Prover

3-coloring C

$$\xrightarrow{(com_1, .., com_n)}$$

Verifier

Chooses an edge$(u, v) \in E$uniformly

$$\xleftarrow{\text{reveal colors of } u, v}$$

$$\xrightarrow{color_u, color_v, dec_u, dec_v}$$

If $Decommit(color_u, com_u, dec_u) = 1$

and $Decommit(color_v, com_v, dec_v) = 1$

Accept that G is 3-colorable Else reject

Building blocks

Computational hardness

- Why bother about Computational adversary?

Theorem

- *IT secure schemes are costly*

construction from OWP

Sender Receiver Hiding holds as $hcp(y)$ is unpredictable. Binding holds as f is a OWP. $Decommit(com, dec)$
$\{$Parse com as (a, b)
Parse dec as y
If $a == f(y)$
Output $b \oplus hcp(y)$
Else Output $\perp\}$

Coin flipping over distance

I can count the number of leaves.
Let me test.
Tosses a coin.
Plucks a leaf if heads.
Please count the leaves now.
Prob[cheating wizard fails] $= \frac{1}{2}$ Alice makes her call
Locks her call in a box
sends this box to Bob without key
Bob tosses a coin
Bob reveals the toss result
Alice reveals her call and sends the box key
Bob opens the box and crosschecks Alice's call
Winner is declared according to the toss result

Building blocks

Algorithms

- Deterministic: For a given input, output of the algorithm is fixed
  - Example: $f(x, y) = xy$

- Deterministic: For a given input, output of the algorithm is fixed
  - Example: $f(x, y) = xy$
- Randomized: Algorithm that has access to uniform bits.
  - For a given input there may several possible outputs depending on the uniform bits
  - Example: $f_{rand}(x, y) = \begin{cases} xy \text{ if } r = 01 \\ 2xy \text{ if } r = 10 \end{cases}$

- Deterministic: For a given input, output of the algorithm is fixed
  - Example: $f(x, y) = xy$
- Randomized: Algorithm that has access to uniform bits.
  - For a given input there may several possible outputs depending on the uniform bits
  - Example: $f_{rand}(x, y) = \begin{cases} xy \text{ if } r = 01 \\ 2xy \text{ if } r = 10 \end{cases}$
  - Randomized algorithm is deterministic given the uniform bits.

Coin flipping over distance

Alice		**Bob**
Makes her call x		
$(dec, com) = Commit(x)$		
	$\xrightarrow{com}$	
		coin toss results in b
	$\xleftarrow{b}$	
	$\xrightarrow{x, dec}$	
		If $Decommit(com, dec) = 1$, Announce winner else STOP