

# Hash Functions, Bit Commitment, and Zero-Knowledge Proofs

Ryder LiuLin

December 7, 2022

# Overview



1. Cryptography Introduction
2. 1-way Functions
3. Bit Commitment
4. Zero-Knowledge Proofs



# Cryptography Introduction

# Cryptography Introduction

## Short Introduction to Cryptography



Study of secrecy and communication across insecure channels to potentially dishonest users.

Basic assumptions:

- ▶ All users have at least equal computational power to us (usually assumed to be equal)
- ▶ All methods and processes are well-known to all users

# Cryptography Introduction

## Background: P vs. NP



**Class P:** problems which can be solved with a polynomial-time algorithm (**quickly solvable**)

**Class NP:** problems whose solutions can be verified in polynomial time (**quickly checkable**)

- ▶ Clearly,  $P \subseteq NP$
- ▶ Converse is unknown (widely believed to be false)

# Cryptography Introduction

## NP-hard and NP-complete

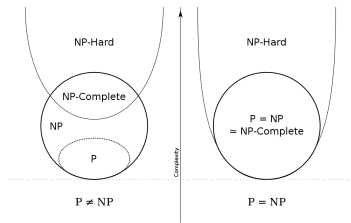


Any NP problem can be easily reduced to an NP-hard problem, and all NP-hard problems in NP are called NP-complete.

Note: If there exists an NP-complete problem in P, all NP problems must be in P ( $P = NP$ ).

Examples of NP-complete problems:

- ▶ Boolean satisfiability problem (**SAT**)
- ▶ Graph coloring problem
- ▶ Subgraph isomorphism problem
- ▶ Traveling salesman problem



Complexity Venn diagram



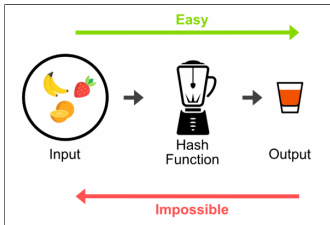
# 1-way Functions



# 1-way Functions

## Definition and Motivation

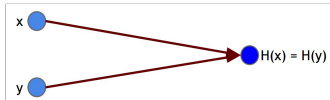
A 1-way function (or cryptographic hash function) is a deterministic function that is easy to compute and hard to invert. The most useful ones have an possibly infinite input length and fixed output length.



1-way-ness

Input		Digest
Fox	cryptographic hash function	DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog	cryptographic hash function	0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABE
The red fox jumps over the blue dog	cryptographic hash function	8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEF6 4819
The red fox jumps over the blue dog	cryptographic hash function	FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
The red fox jumps over the blue dog	cryptographic hash function	BACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

Hides information about input



Collision resistance





# 1-way Functions

## Defining Hardness-to-Invert

1-way functions' strength can be defined by strong and weak:

- ▶ **Strong**: any inversion algorithm succeeds with negligible probability
- ▶ **Weak**: any inversion algorithm fails with noticeable probability

**Theorem**: Weak 1-way functions exist if and only if strong 1-way functions exist.[1]

# 1-way Functions

## Hardcore Predicates and Functions



Hardcore predicate of a 1-way function: **secure bit** based on a 1-way function's output.

Hardcore function is a generalization: secure sequence of bits.

Examples of hardcore predicates:

- ▶ The least significant bit of an RSA output
- ▶ XOR of a random subset of bits of a 1-way function



No noticeable advantage



# Bit Commitment

# Bit Commitment



Commitment scheme: protocol for a sender to commit to a bit **unambiguously** and **secretly**.

2 phases:

- ▶ Commit phase: sender commits to value publicly
- ▶ Reveal phase: sender reveals committed value and commitment process

Such a commitment scheme can be designed based on hardcore predicates or pseudorandom number generators.



# Zero-Knowledge Proofs

# Zero-Knowledge Proofs

## Introduction



Zero-knowledge proof (ZK-pf): protocol for a sender to **unambiguously** demonstrate knowledge to a verifier while keeping the knowledge **secret**.

**Simulation paradigm: zero-knowledge** means that anything the verifier was able to compute after the ZK-pf, it could compute beforehand.

# Zero-Knowledge Proofs

## Creating ZK-Pfs



A ZK-pf for a solution to an NP-complete problem is a ZK-pf for a solution to any NP problem, since NP-complete problems can be reduced to any other NP problem.

**Theorem:** Polynomially many repetitions of a ZK-pf remains zero-knowledge.[1]

So we can repeat a proof that provides non-zero evidence over and over to create strong ZK-pfs.

# Zero-Knowledge Proofs

## Using SAT (E3SAT)



Prover must first:

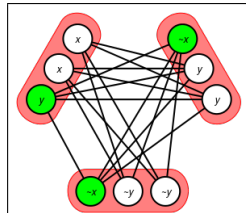
- ▶ Permute variable order and labels, clause order
- ▶ Invert each variable/assignment with probability 0.5
- ▶ Commit to new formula and assignment

Verifier can ask for either:

- ▶ New formula and the variable permutations
- ▶ One clause of the new formula and those assignments

$(x \text{ OR } y \text{ OR } z) \text{ AND } (x \text{ OR } \bar{y} \text{ OR } z) \text{ AND}$   
 $(x \text{ OR } y \text{ OR } \bar{z}) \text{ AND } (x \text{ OR } \bar{y} \text{ OR } \bar{z}) \text{ AND}$   
 $(\bar{x} \text{ OR } y \text{ OR } z) \text{ AND } (\bar{x} \text{ OR } \bar{y} \text{ OR } \bar{z})$

Formulaic Form




Tripartite Graph Form



# References



-  O. Goldreich.  
*Foundations of Cryptography.*  
Cambridge University Press, Cambridge, United Kingdom, 2004.