Bitcoin:
Programming the Future of Money

Topics in Computer Science - ITCS 4010/5010, Spring 2025

Dr. Christian Kümmerle

Lecture 6

Hash Functions & SHA-256

UNIVERSITY OF NORTH CAROLINA
CHARLOTTE

Reading Quiz 2

READING QUIZ 2

List the arguments **why** according to Parker Lewis' article **"Bitcoin Obsoletes All Other Money"**, **economic systems tend to converge on a single form of money**.

General argument flow of article (simplified)

Is money a basic necessity? → Parker Lewis' analysis: Yes! → Do economic systems converge to a single form of money? → Yes! → Which is the money the world will converge to?

- Analyze monetary properties of different goods
- Pick the one with "best" properties

→ Bitcoin

Why? ←

- Improved pricing system if common money is used
- Adoption of other individuals is relevant → Now
  → In the future (store of value)
- Money with better monetary properties will influence individuals' adoption

READING QUIZ 2

What is the "path-dependent nature" of money? Explain this concept using information from the article of Vijay Bojapati.

- Usefulness of money depends on the “monetary premium” attached to a good, which partially depends on the **level of acceptance by other market participants.**
- Level of acceptance / demand of a good as money depends also **on past prices**, as individual **past price performance**/ stability as **indicator for future** usefulness (e.g., as store of value)

READING QUIZ 2

Is a monetary good more likely to be a **good store of value first**, and **then a medium of exchange**, or a **good medium of exchange first**, and then a **good store of value**?

First SoV, then MoE, as:

- Appropriateness of good as store of value depends on monetary properties, such as fungibility, verifiability, divisibility, scarcity, established history, durability, portability.
- After widespread adoption of a good as SoV, the purchasing power of it will become less volatile, and also not increase significantly overtime anymore.
- As a consequence, the opportunity cost of using this good as an exchange good in a trade will decrease, making it more attractive as medium of exchange

- Formula: Adoption of a money as MoE if

“Opportunity Cost of **not** using it as SoV" + "Transaction Cost" < "Cost of trade using alternative MoE”

GRESHAM'S LAW

Sir Thomas Gresham (1519-1579):

Principle: "**Bad money drives out good money**"

Interpretation:
A legally favored money (e.g., by legal tender laws/ fixed, overvalued exchange rate) will drive undervalued currency out of circulation (as medium of exchange).

Examples:
- Fixed gold silver exchange rate in 18th century Britain ("bimetallism")
- "Nakamoto-Gresham's Law"

Reference: Jörg Guido Hülsmann, "The Ethics of Money Production", Chapter 10 on "Legal Tender Laws", 2008.

Cryptographic Hash Functions

HASH FUNCTIONS

D: *domain*, i.e., set on which function is defined

R: *range*, i.e., set in which all outputs of function need to be included in

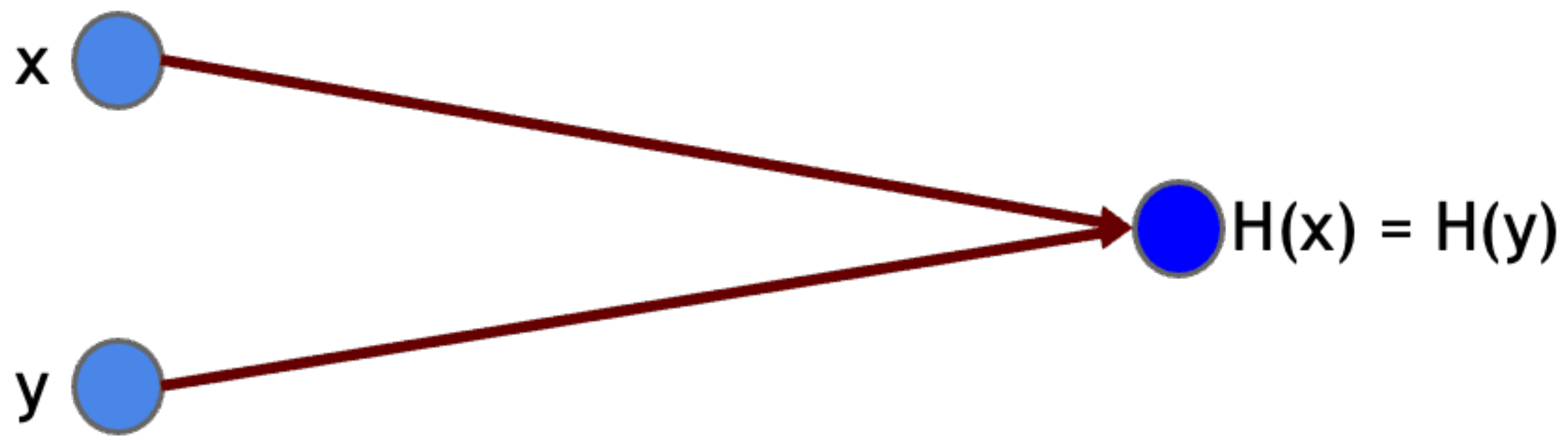
A hash function $H : D \rightarrow R$ is a function that satisfies

1. H takes strings $x \in D$ of **any length** as input
2. Outputs of H (i.e., length elements of R) are of **fixed size**
3. If $x \in \{0,1\}^n \in D$, then computing $H(x)$ has a time complexity of $O(n)$ (H is efficiently computable)

Example: 2. For us often, $R = \{0,1\}^{256}$ (256-bit strings)

CRPYTOGRAPHIC HASH PROPERTY 1: COLLUSION-FREE

A hash function $H : D \to R$ is called collision-free if it is "infeasible" to find two different inputs $x, y \in D, x \neq y$ with same output $H(x) = H(y)$.



Birthday Paradox

- What is the probability P that there **at least two people** in a room of n people were born on the same day (not considering the year)?
- $P = 1$ if $n > 365$
- $P > 0.5$ if $n > 23$ **if birthdays are uniformly distributed**
  (the threshold θ can be approximated as $\theta \sim \sqrt{365}$)

SHA256 hashes per second for different hardware:

- Standard current MacBook Pro:
  ≈ 2000 MH/s = $2 \cdot 10^9$ H/s
- One state-of-the-art Bitcoin mining Application-Specific Integrated Circuit (ASIC) (Bitmain Antminer S21 XP Immersion, Released in June 2024)
  $\approx$ 300 TH/s = $3 \cdot 10^{14}$ H/s
- Entire Bitcoin mining network
  $\approx$ 600 EH/s = $6 \cdot 10^{20}$ H/s

HOW TO BREAK COLLISION-FREENESS

How long does the entire Bitcoin network need to mine to find a SHA-256 collusion using the "birthday attack"?

Is there a faster way to find collisions?

- For some possible hash functions H: Yes!
- For others (such as SHA-256), we don't know of one.
- No hash function H has been mathematically proven to be "practically collision-free".

CRPYTOGRAPHIC HASH PROPERTY 2: HIDING

A hash function $H : D \to R$ is called hiding if it is "infeasible" to find x given $H(r \| x)$, where r is chosen from a probability distribution $\mathscr{P}$ with high min-entropy $\log \frac{1}{P_{\text{max}}}$, where $P_{\text{max}} = \max_i p_i$ and p_i is the probability of the i-th outcome of $\mathscr{P}$.

Example: Uniform distribution on $\{0,1\}^n$ has high min-entropy

CRPYTOGRAPHIC HASH PROPERTY 2: HIDING

Ideally, we would like to have something like this:

Given H(x), it is infeasible to find x.

Problem:

H("heads")

H("tails")

easy to find x!

Commitment Scheme

- *commit(msg, nonce)*, returns output *com*
  Takes message msg and random nonce as input and returns commitment *com*.

  Nonce: "Truly" random number that should only be used once.
- *verify(com, msg, nonce),* returns Boolean output. "Opens envelope"
  verify(com, msg, nonce) == True: If *com == commit(msg, nonce).*
  verify(com, msg, nonce) == False: Otherwise.

"Seals message" by computing and publishing *com*

"Open envelope" by publishing *msg and nonce,* as anyone can check validity using *verify()*

Desired properties:

- Hiding property & Binding Property

Desired properties:

- Hiding property: Given *com,* it is infeasible to find *msg*
- Binding property: It is infeasible to find pairs (*msg,nonce)* and *(msg',nonce') with msg* $\neq$ *msg'* and same commitment.

Q: How to implement a secure commitment scheme with hash functions?

APPLICATIONS OF HASH FUNCTIONS IN THE BITCOIN PROTOCOL

Cryptographic hash functions have multiple purposes for the protocol:

- **Consensus mechanism / bitcoin mining**:
  Central part of cryptographic puzzle to be solved (Double SHA-256)
  -> Ensures agreement on the state
  -> Inflation control
- **Creation of bitcoin addresses from public keys**
- **Checksums** for typed bitcoin addresses (prevention of typos / copy-paste errors)
- **Transaction identification**
- **Construction of transaction "blocks"** via Merkle trees
- **Data integrity** of chain of blocks

CRPYTOGRAPHIC HASH PROPERTY 3: PUZZLE-FRIENDLINESS

A hash function $H : D \rightarrow R$ is called puzzle-friendly if every n-bit $y \in R$, for any k is chosen from a probability distribution $\mathscr{P}$ with high min-entropy, one can only find x such that $H(k \,||\, x) = y$ in time complexity of $\Omega(2^n)$.

APPLICATION OF PUZZLE-FRIENDLINESS

Search Puzzle: Used to define the problem to be solved in bitcoin mining

Consists of

- a hash function $H : D \to R$
- a value, id (which we call the *puzzle-ID*), chosen from a high min-entropy distribution
- and a target set Y

A solution to this puzzle is a value x such that

$$H(id \,||\, x) \in Y.$$

Typically, choose Y as small subset of R.

BITCOIN MINING

- Use twice consecutive application of SHA-256 hash function as H
- Take as id the **block header** which is a function of block header information (contains all transaction information of transactions to be added and hash of previous block) and a random nonce ("coinbase nonce", related to the transaction that pays out block reward to miner)
- Solution x depends on **varying arbitrary value "header nonce"** that does not essential header information
- Y defined as subset of $\{0,1\}^{256}$ such that its big-endian representation is smaller than a certain difficulty value.

Different hash functions used in cryptography:

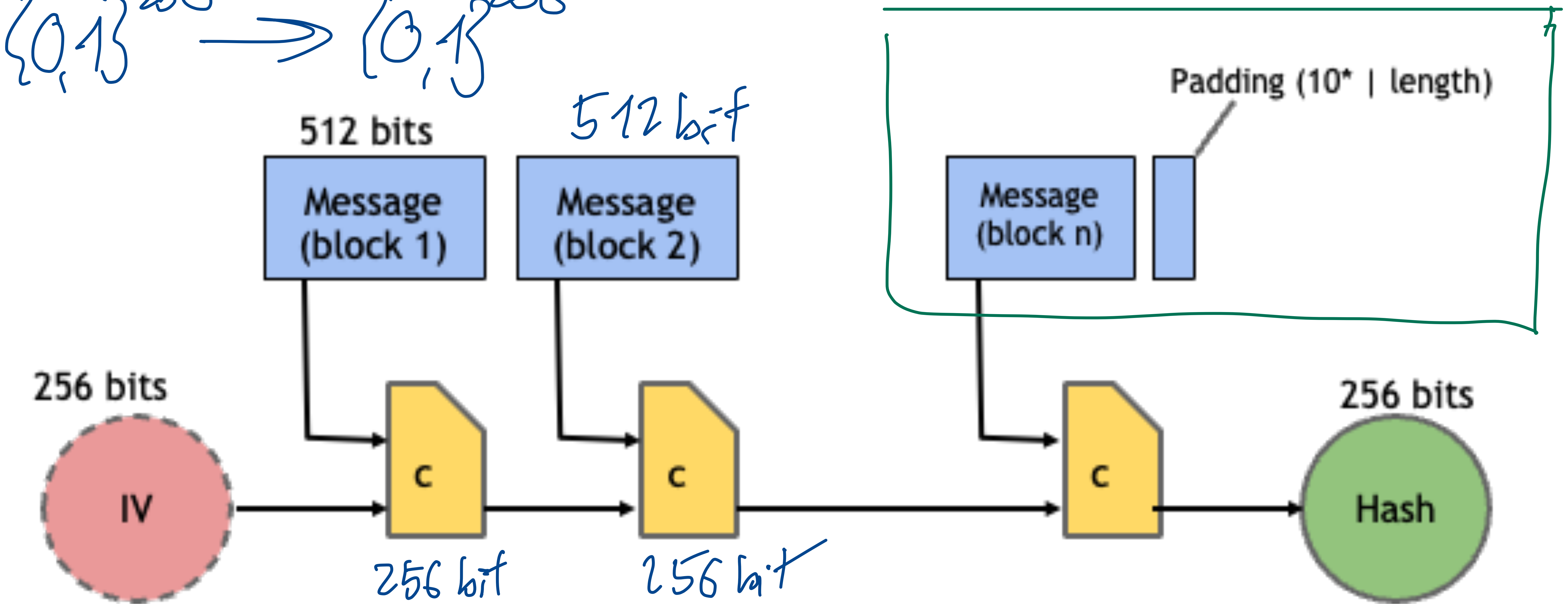
Name	Bits	Secure so far?	Used in Bitcoin?
SHA256	256	Yes	Yes
SHA512	512	Yes	Yes, in some wallets
RIPEMD160	160	Yes	Yes
SHA-1	160	No. A collision has been found.	No
MD5	128	No. Collisions can be trivially created. The algorithm is also vulnerable to pre-image attacks, but not trivially.	No

SHA-256

- Part of SHA-2 (Secure Hash Algorithm 2) hash functions designed by the US National Security Agency (NSA), published in 2001
- Has a range of 256 bits ($R = \{0,1\}^{256}$)
- Widely used in security protocols such as SSL/TLS for secure web browsing
- Mentioned in the Bitcoin white paper directly

MERKLE-DAMGÅRD CONSTRUCTION USED BY SHA-256

$c : \{0,1\}^{512} \times \{0,1\}^{256} \longrightarrow \{0,1\}^{256}$



IV: Initialization vector

c: A certain fixed-length domain collision-resistant function called **compression function** deterministic!

Compression function of SHA-256 is based on the [Davies-Meyer construction] applied to the SHACAL-2 block cipher.