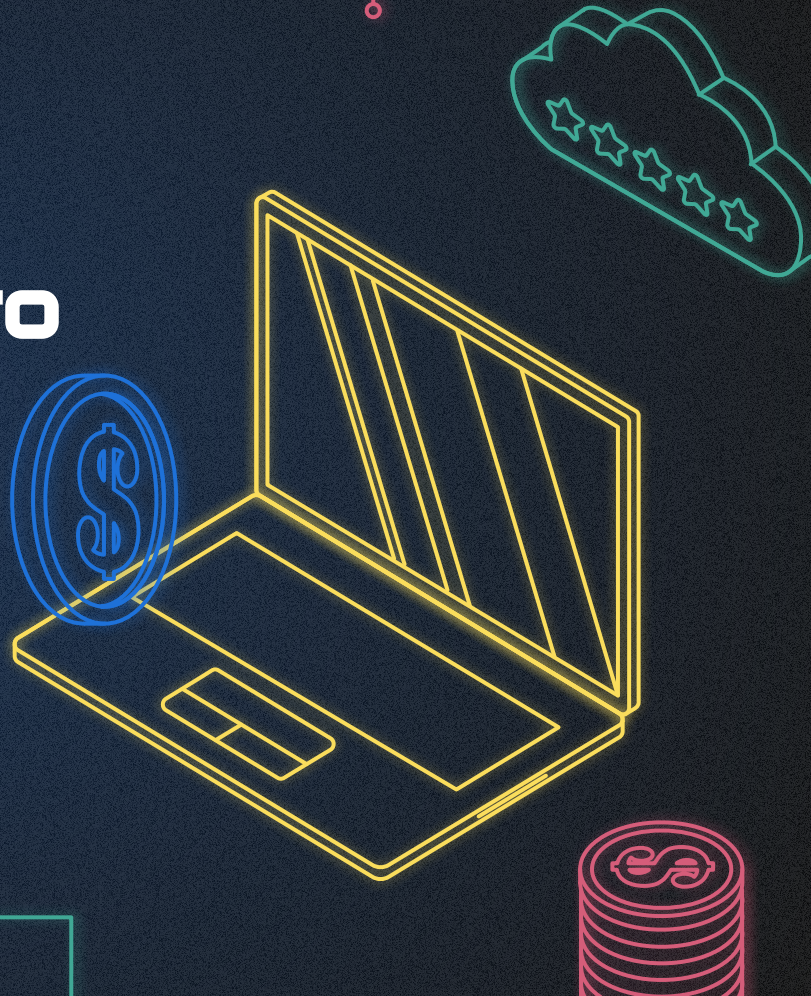
Week #1

Syllabus and Intro to Cryptography

Welcome to CMSC398F!



Syllabus

- Syllabus is available on the ELMS page
- Lecture slides will be available on the github after every lecture: https://ter.ps/cmsc398F
- TA office hours for every week will be announced on piazza
- No textbooks from the syllabus are required
- Feedback form released every two weeks
- Piazza coming soon

Syllabus

Quick Rundown of topics:

- Introduction to Cryptocurrency
- Hash functions and attacks
- History of Bitcoin
- Blockchain Structure
- Proof-of-Work
- Mining, Faucets
- Wallets & Anonymity
- Crypto Markets, Market Caps, Investors
- Bitcoin as a Platform
- Introduction to Smart Contracts
- DAOs and ICOs
- Smart contract development
- Miscellaneous

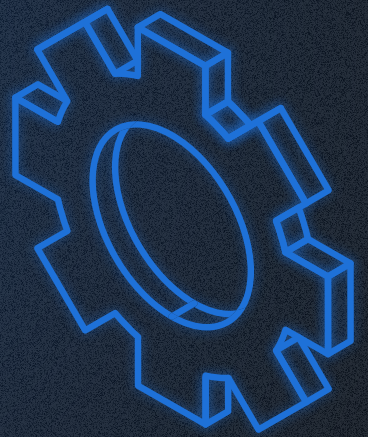
Syllabus (Class Structure)

- Class Participation (10%)
  Attendance is mandatory. Let us know if you cannot make it.
- Quizzes (20%)
  10 multiple choice questions based on the lecture
- Projects (30%)
  Simplified implementations of blockchain technology
- Midterm (20%)
  Concept-based exam. Consists of multiple choice and short answer questions
- Final (20%)
  Implementation of a smart contract
  More details later in the semester

Contact Us

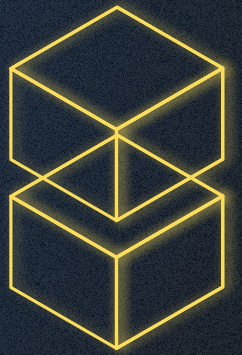
Om: ompathak@umd.edu
Nikhil: nghate@umd.edu
Soham: sdigamba@umd.edu

LET'S BEGIN!

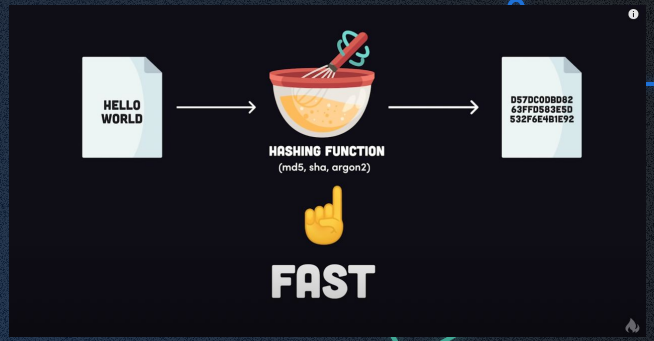


Cryptography and hash functions

- Cryptography is the study and practice of sending secure, encrypted messages or data between two or more parties.
- Cryptography allows transactions to be “trustless” – and makes secure transactions between strangers possible without a “trusted intermediary” like a bank or Venmo in the middle.
- Hash functions are mathematical functions that transform or "map" a given set of data into a bit string of fixed size, also known as the "hash value."
- Hash Function

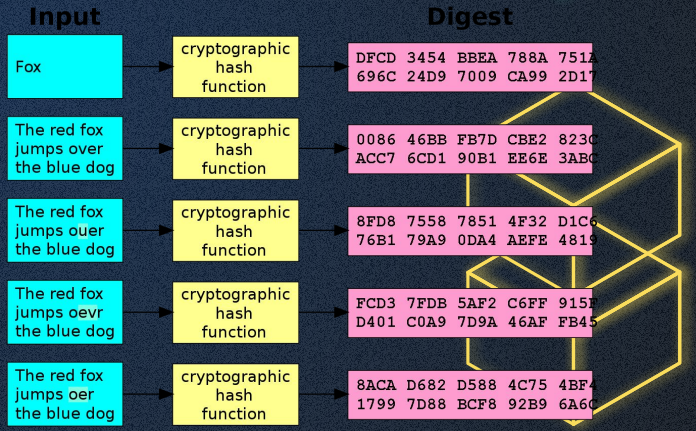


Hash Functions



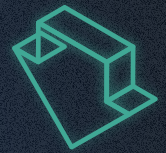
3 Properties of Hash Functions:

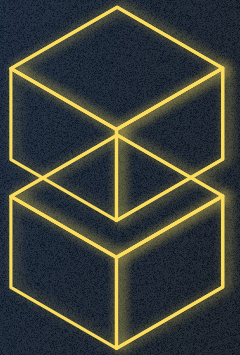
- **Collision-free**: no two inputs should map to the same output hash.
- **Irreversible:** You should not be able to guess the input value based on the output value.
- **Puzzle-friendly**: Should be difficult to select an input that provides a predefined output. Thus, the input should be selected from a distribution that's as wide as possible.



Encryption vs. Hashing

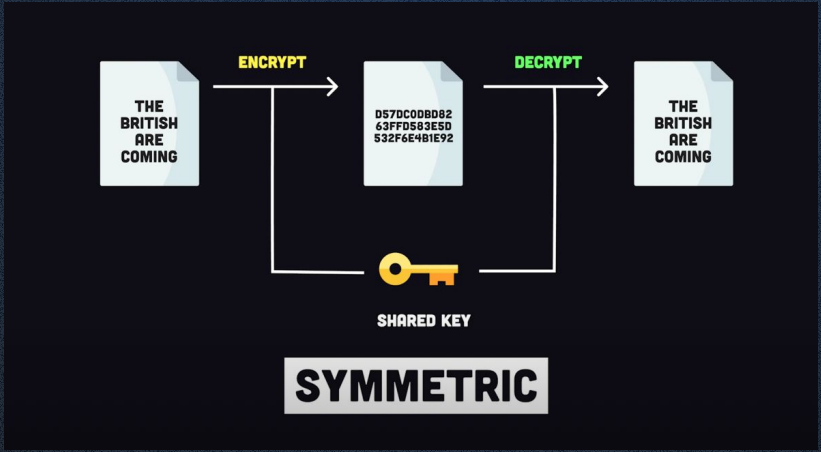
- **Encryption** is a two way function that includes encryption and decryption, and is used to send data securely.
  - It is the process of converting plaintext to cypher-text, which can be converted back to plain text.
- Hashing and encryption are **NOT** the same
- Hashing is a one way function that turns plain text into a unique digest that is irreversible. It is used to validate the integrity of messages, compare large data, and in hash tables





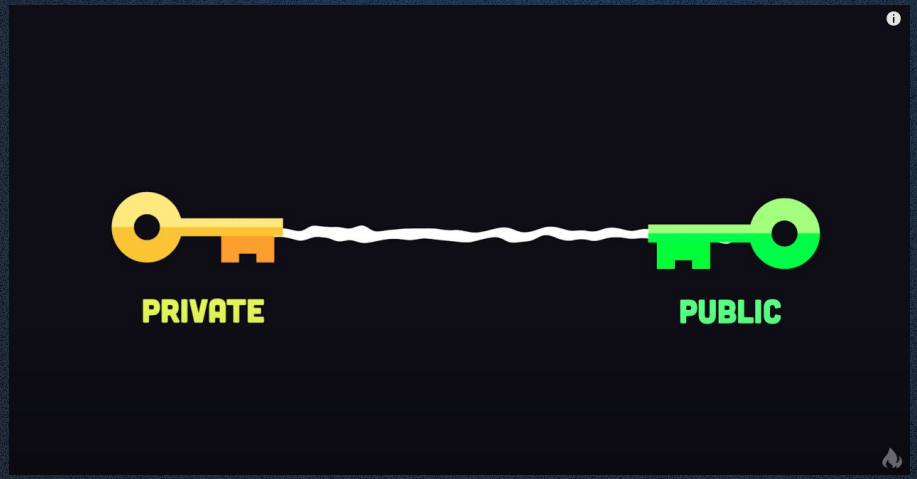
Symmetric Encryption

- What if we want to send a message to someone securely, and we want them to be able to read the message?
- In symmetric encryption, we use a hash function to scramble up a message (into cypher-text), but also provide a 'secret key' to decrypt the message
- The secret key must be shared between the sender and receiver to decrypt the message
- Key size is small, which means less storage space and faster transmission, so it is good for bulk encryption.



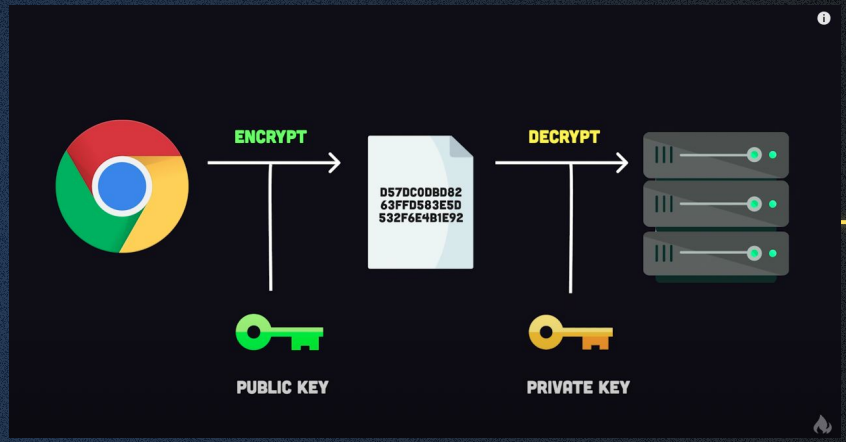
Key-pairs

- There is a limitation when it comes to symmetric encryption: both parties must share a private key.
- Instead, we can use two keys: a public key and private key, which are mathematically linked.



Asymmetric Encryption

- Asymmetric encryption uses public and private keys to share information.
- Sender and receiver both have their own pair of keys. First, the sender obtains the receivers public key.
- The public key is used by the sender to encrypt a message, and the corresponding private key held by the receiver is used to decrypt the message.
- This way, the sender and receiver do not need to share the same key.
- We use this all the time when we visit websites with https.
- Asymmetric and symmetric encryption are often used together to leverage the strength of both algorithms
- Public-private key encryption consists of mathematically complex hash functions
  - SHA-256
  - SCRYPT
  - RSA



Application of Encryption

- Elliptic Curve Cryptography (ECC) is a type of asymmetric cryptography that is used in Bitcoin's Blockchain. It is used to ensure that funds can only be spent by their rightful owners using digital signatures (more on this later).
- The Bitcoin network issues all users a private key (essentially a really strong password) from which it cryptographically generates a linked public key.
- Your public key is generated from your private key via hashing. It's virtually impossible to reverse this process, so nobody can guess your private key from your public key.
- As the public and private keys are linked, the network knows that your bitcoin belong to you – and will remain yours as long as you have your private key.

Summary

- Hash functions and introduction to encryption functions
- Before next lecture, read a summary of the white paper by Satoshi Nakamoto.
- First Quiz posted!