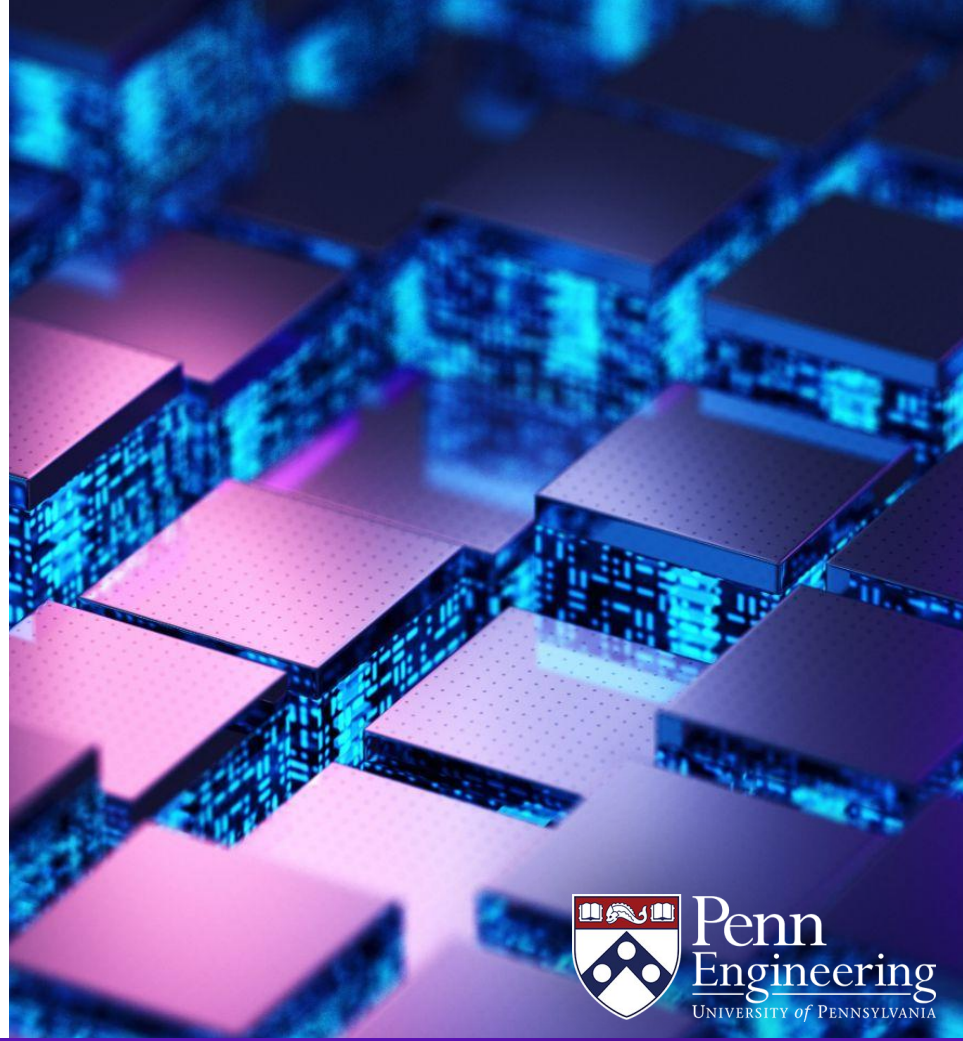
EAS 5830: BLOCKCHAINS

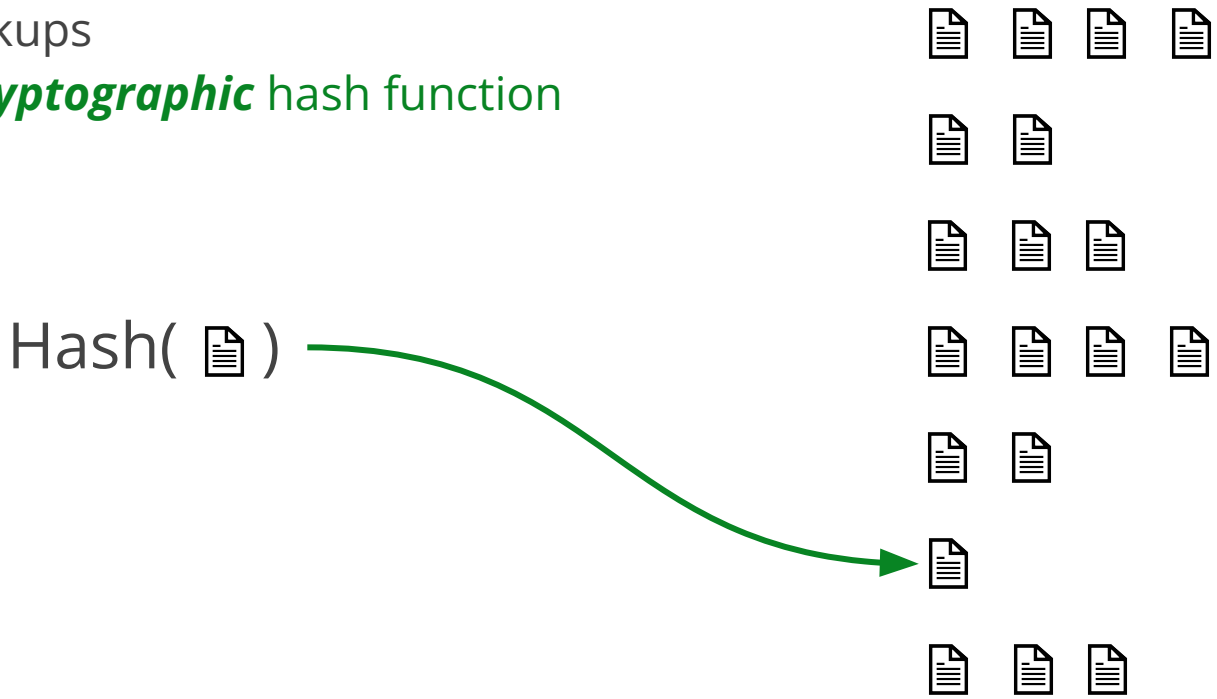
Cryptographic Hash Functions

Professor Brett Hemenway Falk



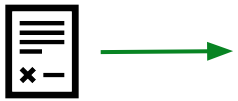
Hashing - Balls into Bins

- Store files in a location based on their hash
- Fast lookups
- Not a ***cryptographic*** hash function

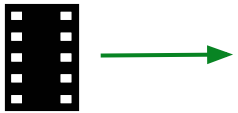


Cryptographic Hashing - a Digital Fingerprint

- Cryptographic hash functions should be collision resistant
- Can't find x, and y with $x \neq y$ and $h(x) = h(y)$
- Cryptographic hash functions take any bit string as input
- Fixed length output (e.g. 256 bits)
  - Note 256 bits = 64 hexadecimal digits (since 256/4 = 64)



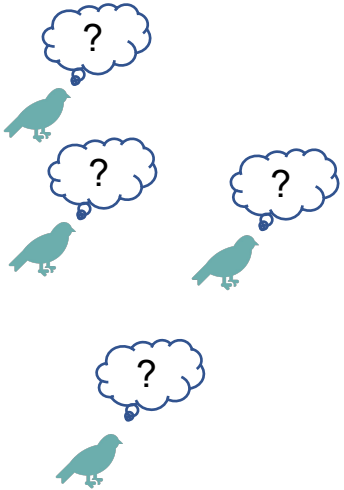
3c6b05d0fb0da56eb04549ce2bf746e50e4966136b8963dbfeb7aa23d017d25b

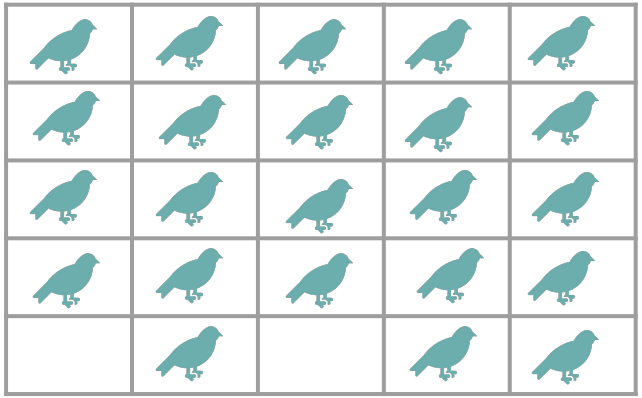


c4e5f54e93a124f45a7b9a08453fb2d4799f559f0876352df64fe4855b2e398b

c0862201e27e10f591cc0e3fb8d1dd4f6a4af2559d04a71fb0cb142d59b2f6b7

The Pigeonhole Principle





Once you have more pigeons than pigeonholes, at least one pigeonhole must have two pigeons

A Digital Fingerprint

- Each person has a fingerprint
- Fingerprint is much smaller than a person
- A fingerprint can uniquely identify the person
- Everyone must agree to use fingerprints
  - Ears could make better unique IDs than fingerprints

Hash Function Standards

- Everyone must agree on the same hash function

- NIST creates standards
  - SHA-0 (1993)
  - SHA-1 (1995)
  - SHA-2 (2001)
  - SHA-3 (2015)

Characteristics of Hash Functions

- Completely deterministic
- Variable length inputs (so we can hash any digital data)
- Fixed length outputs (so we can easily store and compare hashes)
- No collisions (fingerprints should be unique)

Characteristics of Hash Functions

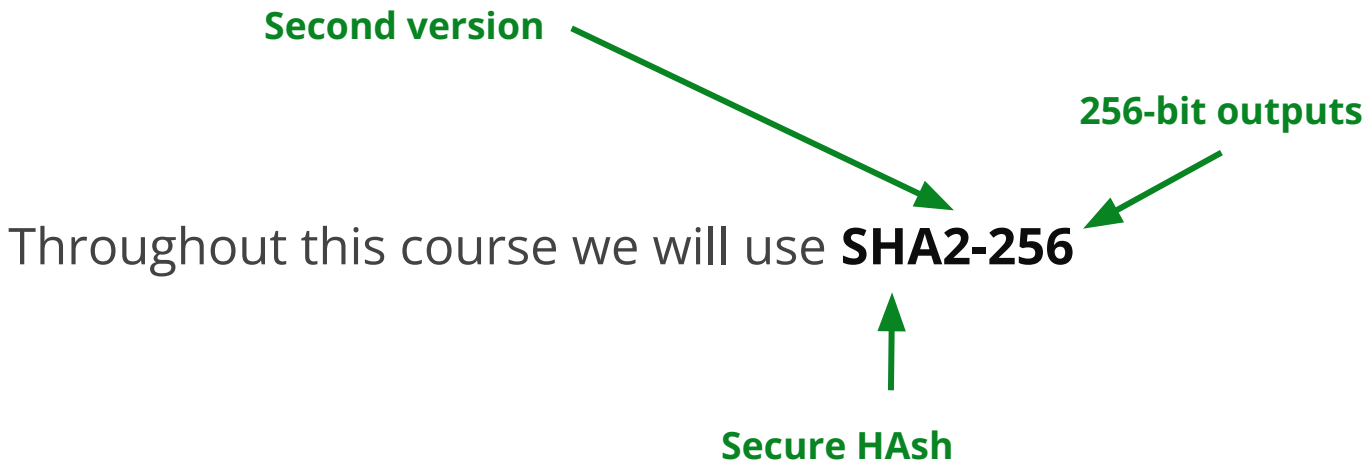
- Completely deterministic
- Variable length inputs (so we can hash any digital data)
- Fixed length outputs (so we can easily store and compare hashes)
- ~~No collisions (fingerprints should be unique)~~
- It should be intractable to find collisions
    - Hard to invert (i.e., hard to find pre-images)
- Small changes in input lead to large changes in output
- Outputs look “random”

Conflicting Properties of Hash Functions

- Hash algorithms are completely deterministic
  - Only useful if two people get the same output when they hash a file

- Outputs “look random”
  - Security requires that no one can find patterns or correlations in the hash function outputs
  - Formalized in the notion of the “Random Oracle Model”

SHA2-256


Second version
256-bit outputs
Throughout this course we will use SHA2-256
Secure HAsh

Examples of SHA2-256 Hashes

Input	Output
Bitcoin	b4056df6691f8dc72e56302ddad345d65fead3ead9299609a826e2344eb63aa4
bitcoin	6b88c087247aa2f07ee1c5956b8e1a9f4c7f892a70e324f1bb3d161e05ca107b
The full text of the Gettysburg Address	975fc511d28b82c65ce03d60e79d5271768687f4c736d9a2201e0a3738f20bdd

Applications of Hashing

- Compressing messages for signatures
- Detecting tampering
- Storing passwords
- Cryptographic Commitments
- Pseudorandom number generation
- Hash chains
- Proofs of work