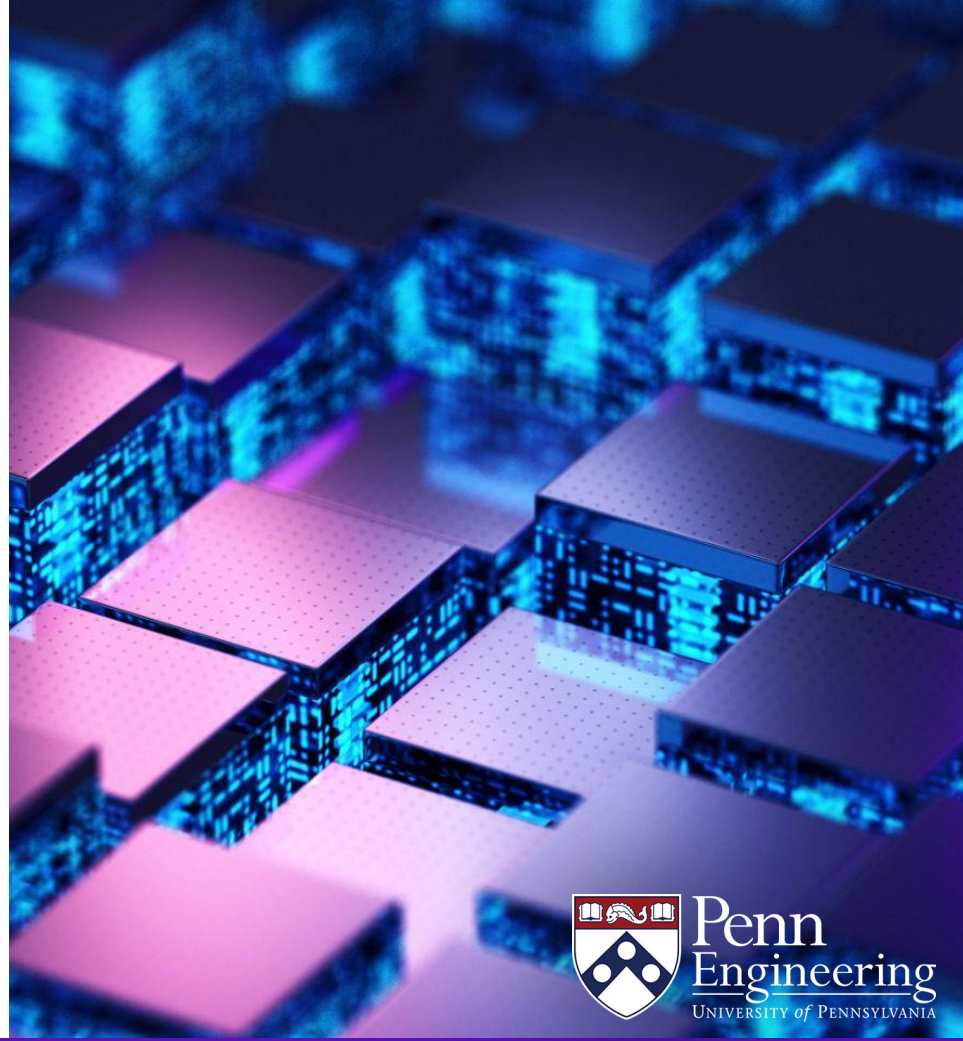
EAS 5830: BLOCKCHAINS

The SHA Family of Hash Functions

Professor Brett Hemenway Falk



The SHA family of hash functions

SHA0

- Designed by the NSA
- Published by NIST in 1993
- Output size 160 bits

SHA1

- Designed by the NSA
- Published by NIST in 1995
- Output size 160 bits

SHA2

- Designed by the NSA
- Published by NIST in 2001
- Output sizes 224, 256, 384, 512 bits

SHA3

- Designed by competition
- Standardized by NIST in 2015
- Output sizes 224, 256, 384, 512 bits

SHA2

- SHA2
- Developed by the NSA in 2001
- Standardized by NIST
- Takes arbitrary length inputs
- Outputs 256 bits (64 hexadecimal digits)

SHA3 Competition

- **October 2008** – Submissions due
- **December 2008** – First-round candidates announced
- **July 2009** – Second-round candidates announced
- **December 2010** – Final-round candidates announced
  - BLAKE
  - Grøstl
  - JH
  - Keccak
  - Skein
- **October 2012** – Keccak declared the winner
  - Ethereum (launched in 2013) implements Keccak
- **August 2015** - NIST makes final tweaks to Keccak to create SHA-3

SHA1 is 'broken'

- Published by NIST in 1995
- First collision found in February 2017
- SHA1 has 160-bit outputs
- Attack required "only" 2^{63} SHA1 computations
- 100,000 times faster than brute-force search
- Costs about $100K
- https://shattered.io/

When is a hash function broken?

- When you can find collisions:
  - $h(x) = h(y)$
- Note:
  - A random collision may not lead to a concrete attack
  - To be conservative, if any collisions are found, the hash function is considered ``broken”

What can you do if you break a hash function?

- Tamper with files
  - Insert viruses into widely distributed files
- Forge signatures
  - If signature is on a public-key this allows Man-in-the-Middle attacks
- Disrupt blockchains
  - Mine faster
  - Get elected in sortition systems (e.g. Algorand)
  - Insert blocks into the middle of a chain (double spend)
  - Forge signatures (spend from other people's accounts)
  - Break commitments (break contracts)

Further Reading

- The First 30 Years of Cryptographic Hash Functions and the NIST SHA-3 Competition
- SHA-3: Where We've Been, Where We're Going
- Lecture notes on Cryptography (Chapter 8)
- How cryptographers think about hashes (a theoretical perspective)
    - Collision-Free Hashing from Lattice Problems
    - A Design Principle for Hash Functions
    - Merkle-Damgård Revisited: How to Construct a Hash Function