EECS 3481 APPLIED CRYPTOGRAPHY

CRYPTOGRAPHIC HASH FUNCTIONS

PROF H ROUMANI
Dept. of Electrical Engineering and Computer Science, York University

1

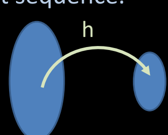
WHAT

Some figures from Cryptography and Network Security, W. Stallings, Prentice-Hall

2

WHAT IS A HASH FUNCTION?

Function h: x → y=h(x) maps a message of an *arbitrary* size to a *fixed*-size (n) bit sequence.



- Manifestly Many-to-One
  |domain| = ∞, |range| = 2^n
- Fast to compute (by S/W and H/W)
  Often used on large inputs
- Digests the message
  A single bit flip will likely not lead to a collision

Explore the md5sum and sh1sum Linux commands.

3

3

Hash via JCE

```
MessageDigest md = MessageDigest.getInstance("<algo>");
byte[] hash = md.digest( message expressed as byte[] );
```

ALGORITHM	GROUP	DIGEST	BLOCK
MD5	MD	128	512
SHA-1	SHA-1	160	512
SHA-224	SHA-2	224	512
SHA-256	SHA-2	256	512
SHA-512	SHA-2	512	1024
SHA3-256	SHA-3	256	1088
SHA3-512	SHA-3	512	576

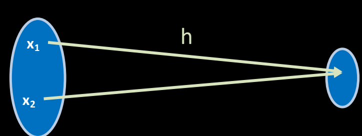
4

4

What is a Cryptographic Hash Function?

It is a hash function with one or more of these properties:

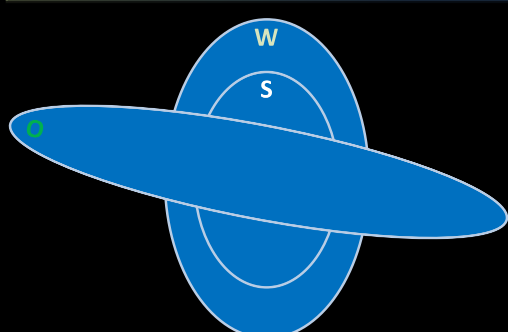
1. Pre-image Resistance (one-way-ness)
   Given y, infeasible to find *any* x: h(x) = y
2. 2nd Pre-image [Weak Collision] Resistance
   Given any x1, infeasible to find x2≠x1: h(x1) = h(x2)
3. Strong Collision Resistance
   Infeasible to find any x1, x2 pair (x2≠x1): h(x1) = h(x2)



5

5

One-Way / Weak / Strong Venn Diagram



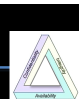
6

6

WHY

7

The I in CIA



Confidentiality ✅

*How about the **integrity** of*

- Content?
- Sender?
  (aka AUTH)
- Sending?
  (aka Non-Repudiation)
- Time of Sending?
  (aka Freshness)

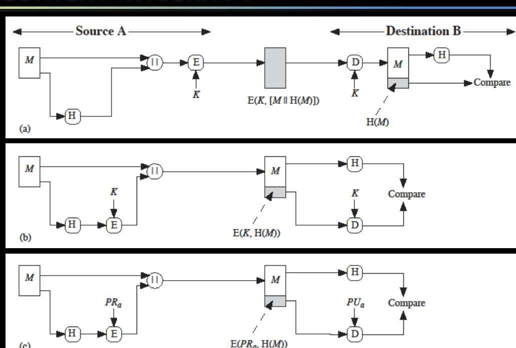
Can symmetric or asymmetric Crypto (as-is / augmented) come to the rescue?

No, encryption does not automatically provide integrity.
The *third building block is needed.*

8

8

Content Integrity I



9

9

Content Integrity II

- Digest = the Hash
  AKA *tag*, *checksum*, or *PRF* (Pseudo-Random Function)
- MAC (Message Authenticated Code)
  Encrypt digest with a secret key: MAC = E[K, h(m)]
- HMAC
  Combine digest with a secret key
  h[K1 || h(K2 || m] where K1 and K2 are derived from K
- Signature
  Encrypt digest with a private key.
  For RSA = $[\ h(m)\ ]^d$ mod n

10

10

Sender Integrity (AKA Authentication)

- **Symmetric Crypto**
  Ensures auth but can repudiate and no freshness
- **Challenge-Response**
  Alice sends nonce n to Bob; he returns *E(k, n) or HMAC(k, n).*
  For mutual auth, she returns *E(k, f(n)) or HMAC(k, f(n)). Can still repudiate but ensures freshness.*
- **Asymmetric Crypto**
  Alice signs a nonce (encrypts its hash with her private key and sends it. This yields auth. + freshness + non-repudiation.

11

11

Time Integrity [Done at the message or protocol level]

- **Sequence Numbers**
  Adds session overhead: a counter per party.
- **Timestamps**
  Requires frequent clock synchronization and tolerance to network delays (by providing time windows).
- **Request-Response Nonce**
  Ensures "freshness" with an unpredictable, random nonce.
  See Challenge-Response in previous slide.

12

12

Hashing Applications (beyond messaging)

- Software Download
  Provide a link to S/W and post its hash on a read-only site.
- Password Storage Best-Practice
  Rather than storing the password, store only its hash.
- Blob Indexing, Fingerprinting, and Caching
  Use the blob's hash as a key.
- Online Bidding (Zero-Knowledge)
  Blind/Salt your bid then hash it.
- Blockchain Immutability and Mining
  Each block has the hash of its predecessor. Proof of work thru hash constraints, e.g. < 2^{254}.

13

13

Exercises

For each of the applications shown, what is the key property of the hashing function (Oneway/Weak/Strong)?

- *Software downloads*
- *Password storage*
- *Bidding*
- *Blob indexing*

For each of the three use cases in Slide #9, critique the security of the case in terms of:

- *Confidentiality*
- *Content Integrity*
- *Sender Integrity*
- *Source Repudiation*

14

14

ATTACKS

15

Birthday Attack

- x people in a room. What is the probability W of at least one sharing your birthday?
- x people in a room. What is the probability S of at least two sharing a birthday?
- $S \approx 1 - \exp(-x^2/2N)$ where N=365
- To achieve a probability of more than 50-50, we need $x >\approx 1.177 * \mathrm{sqrt}(N)$

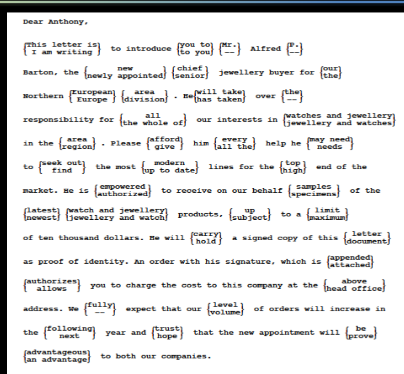
>> Only $2^{n/2}$ evals to find collisions in an n-bit hash ! <<

Example: to fabricate a message, make $2^{n/2}$ variations in the real message and $2^{n/2}$ of the fraudulent. Prob(match) > 50%

16

16

Message Fabrication Example



Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --} Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the} Northern {European / Europe} {area / division}. He {will take / has taken} over {the / --} responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches} in the {area / region}. Please {afford / give} him {every / all the} help he {may need / needs} to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the {latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum} of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document} as proof of identity. An order with his signature, which is {appended / attached} {authorizes / allows} you to charge the cost to this company at the {above / head office} address. We {fully / --} expect that our {level / volume} of orders will increase in the {following / next} year and {trust / hope} that the new appointment will {be / prove} {advantageous / an advantage} to both our companies.

Figure from Cryptography and Network Security, W. Stallings, Prentice-Hall

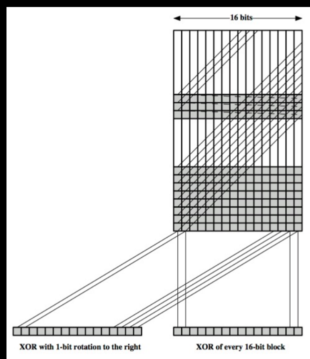
17

17

HOW

18

Toy Example



19

19

Approaches

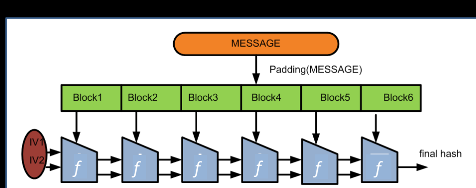
- **Formula-Based**
  Examples: $y = \Sigma x \bmod n$ or $\Sigma x^2 \bmod n$
- **Iterative Block Compression**
  Examples: Merkle–Damgard (SHA1/2)
- **Sponge**
  Examples: Keccak (SHA3)

20

20

Merkle-Damgard Compression

Used in MD5, SHA1, and SHA2

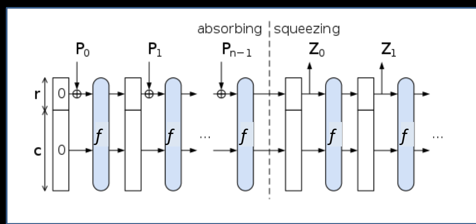


For example, you can use any symmetric cipher in the CBC mode. IV1 seeds the CBC while IV2 seeds the keys. The ciphertext of each block is the key for the next.

21

21

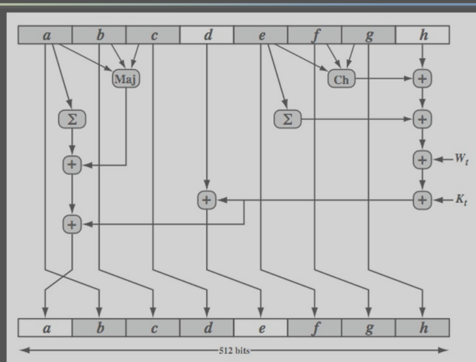
KECCAK SPONGE



Used in SHA3 (224-512)

22

TYPICAL KECCAK F FUNCTION



23

HOW: TYPICAL *f* OPERATIONS

Ch(e,f,g) = (e AND f) XOR (NOT e AND g)

Maj(a,b,c) = (a AND b) XOR (a AND c) XOR (b AND c)

∑(a) = ROTR(a,28) XOR ROTR(a,34) XOR ROTR(a,39)

∑(e) = ROTR(e,14) XOR ROTR(e,18) XOR ROTR(e,41)

+ = addition modulo 2^64

K_t = a 64-bit additive constant

W_t = a 64-bit word from the current 512-bit input block

$W_t = \sigma_0(W_{t-2}) + W_{t-7} + \sigma_1(W_{t-15}) + W_t-16$ (t = 16…79)

σ_0 = ROTR(1) xor ROTR(8) xor SHR(7), σ_1 = ROTR(19) xor ROTR(60) xor SHR(6)

24