Admin

• Additional office hours scheduled
  – 12:30-1:30pm on Fridays
  – A single Zoom room for the whole 12:30-2:30pm timeslot
When is an Encryption Scheme “Secure”?

• Hard to recover the key?
  – What if attacker can learn plaintext without learning the key?

• Hard to recover plaintext from ciphertext?
  – What if attacker learns some bits or some function of bits?
How Can a Cipher Be Attacked?

- Attackers knows ciphertext and encryption algorithm
  - What else does the attacker know? Depends on the application in which the cipher is used!
Chosen Plaintext Attack

PIN is encrypted and transmitted to bank

cipher(key,PIN)

Crook #1 changes his PIN to a number of his choice

Crook #2 eavesdrops on the wire and learns ciphertext corresponding to chosen plaintext PIN

... repeat for any PIN value
How Can a Cipher Be Attacked?

- Attackers knows ciphertext and encryption algorithm
  - What else does the attacker know? Depends on the application in which the cipher is used!

- Ciphertext-only attack
- KPA: Known-plaintext attack (stronger)
  - Knows some plaintext-ciphertext pairs
- CPA: Chosen-plaintext attack (even stronger)
  - Can obtain ciphertext for any plaintext of his choice
- CCA: Chosen-ciphertext attack (very strong)
  - Can decrypt any ciphertext except the target
Very Informal Intuition

• Security against chosen-plaintext attack (CPA)
  – Ciphertext leaks no information about the plaintext
  – Even if the attacker correctly guesses the plaintext, he cannot verify his guess
  – Every ciphertext is unique, encrypting same message twice produces completely different ciphertexts
    • Implication: encryption must be randomized or stateful

• Security against chosen-ciphertext attack (CCA)
  – Integrity protection – it is not possible to change the plaintext by modifying the ciphertext
So Far: Achieving Privacy

Encryption schemes: A tool for protecting privacy.

Message = M
Ciphertext = C
Now: Achieving Integrity

Message authentication schemes: A tool for protecting integrity.

Integrity and authentication: only someone who knows KEY can compute correct MAC for a given message.
Reminder: CBC Mode Encryption

- Identical blocks of plaintext encrypted differently
- Last cipherblock depends on entire plaintext
  - Still does not guarantee integrity
CBC-MAC

- Not secure when system may MAC messages of different lengths (*more in section!*).
- NIST recommends a derivative called CMAC [FYI only]
Another Tool: Hash Functions
You Just Did This

```bash
franzi@codered:~/sploits$ md5sum sploit0.c
3a2e6ce795bce4d06df1ff6835d25cea  sploit0.c
franzi@codered:~/sploits$
```
Hash Functions: Main Idea

- Hash function $H$ is a lossy compression function
  - Collision: $h(x) = h(x')$ for distinct inputs $x$, $x'$
- $H(x)$ should look “random”
  - Every bit (almost) equally likely to be 0 or 1
- Cryptographic hash function needs a few properties...
Property 1: One-Way

• Intuition: hash should be hard to invert
  – “Preimage resistance”
  – Let $h(x') = y \in \{0,1\}^n$ for a random $x'$
  – Given $y$, it should be hard to find any $x$ such that $h(x)=y$

• How hard?
  – Brute-force: try every possible $x$, see if $h(x)=y$
  – SHA-1 (common hash function) has 160-bit output
    • Expect to try $2^{159}$ inputs before finding one that hashes to $y$. 

5/5/20  CSE 484 / CSE M 584 - Spring 2020
Property 2: Collision Resistance

• Should be hard to find $x \neq x'$ such that $h(x) = h(x')$
Birthday Paradox

• Are there two people in the first 1/8 of this class that have the same birthday?
  – 365 days in a year (366 some years)
    • Pick one person. To find another person with same birthday would take on the order of 365/2 = 182.5 people
    • Expect birthday “collision” with a room of only 23 people.
    • For simplicity, approximate when we expect a collision as $\sqrt{365}$.

• Why is this important for cryptography?
  – $2^{128}$ different 128-bit values
    • Pick one value at random. To exhaustively search for this value requires trying on average $2^{127}$ values.
    • Expect “collision” after selecting approximately $2^{64}$ random values.
    • 64 bits of security against collision attacks, not 128 bits.
Property 2: Collision Resistance

• Should be hard to find $x \neq x'$ such that $h(x) = h(x')$
• Birthday paradox means that brute-force collision search is only $O(2^{n/2})$, not $O(2^n)$
  – For SHA-1, this means $O(2^{80})$ vs. $O(2^{160})$
One-Way vs. Collision Resistance

One-wayness does not imply collision resistance.

Collision resistance does not imply one-wayness.

You can prove this by constructing a function that has one property but not the other. (Details on next slide, FYI only.)
One-Way vs. Collision Resistance
(Details here mainly FYI)

• One-wayness does not imply collision resistance
  – Suppose \( g \) is one-way
  – Define \( h(x) = g(x') \) where \( x' \) is \( x \) except the last bit
    • \( h \) is one-way (to invert \( h \), must invert \( g \))
    • Collisions for \( h \) are easy to find: for any \( x \), \( h(x_0) = h(x_1) \)

• Collision resistance does not imply one-wayness
  – Suppose \( g \) is collision-resistant
  – Define \( y = h(x) \) to be 0x if \( x \) is \( n \)-bit long, 1g(x) otherwise
    • Collisions for \( h \) are hard to find: if \( y \) starts with 0, then there are no collisions, if \( y \) starts with 1, then must find collisions in \( g \)
    • \( h \) is not one way: half of all \( y \)'s (those whose first bit is 0) are easy to invert (how?); random \( y \) is invertible with probab. \( \frac{1}{2} \)
Property 3: Weak Collision Resistance

• Given randomly chosen $x$, hard to find $x'$ such that $h(x)=h(x')$
  – Attacker must find collision for a specific $x$. By contrast, to break collision resistance it is enough to find any collision.
  – Brute-force attack requires $O(2^n)$ time

• Weak collision resistance does not imply collision resistance.
Hashing vs. Encryption

• Hashing is one-way. There is no “un-hashing”
  – A ciphertext can be decrypted with a decryption key...
    hashes have no equivalent of “decryption”
• Hash(x) looks “random” but can be compared for equality with Hash(x’)
  – Hash the same input twice → same hash value
  – Encrypt the same input twice → different ciphertexts
• Cryptographic hashes are also known as “cryptographic checksums” or “message digests”
Application: Password Hashing

• Instead of user password, store $\text{hash(password)}$
• When user enters a password, compute its hash and compare with the entry in the password file
• Why is hashing better than encryption here?

• System does not store actual passwords!
• Don’t need to worry about where to store the key!
• Cannot go from hash to password!
**Application: Password Hashing**

• Which property do we need?
  – One-wayness?
  – (At least weak) Collision resistance?
  – Both?
Application: Software Integrity

Goal: Software manufacturer wants to ensure file is received by users without modification.

Idea: given goodFile and hash(goodFile), very hard to find badFile such that hash(goodFile)=hash(badFile)
Application: Software Integrity

• Which property do we need?
  – One-wayness?
  – (At least weak) Collision resistance?
  – Both?
Which Property Do We Need?
One-wayness, Collision Resistance, Weak CR?

- UNIX passwords stored as hash(password)
  - **One-wayness**: hard to recover the/a valid password
- Integrity of software distribution
  - **Weak collision resistance**
  - But software images are not really random... may need **full collision resistance** if considering malicious developers
Which Property Do We Need?

• UNIX passwords stored as hash(password)
  – **One-wayness**: hard to recover the/a valid password

• Integrity of software distribution
  – **Weak collision resistance**
  – But software images are not really random... may need **full collision resistance** if considering malicious developers

• Private auction bidding
  – Alice wants to bid B, sends H(B), later reveals B
  – **One-wayness**: rival bidders should not recover B (this may mean that she needs to hash some randomness with B too)
  – **Collision resistance**: Alice should not be able to change her mind to bid B’ such that H(B)=H(B’)