CSE 484 / CSE M 584: Computer Security and Privacy

Cryptography:
Symmetric Encryption (continued),
Hash Functions, Message Authentication Codes

Spring 2015

Franziska (Franzi) Roesner
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials...
Reminders

• Homework #1 due today @5pm
• Checkpoint for lab #1 due Monday @5pm
  – Send key to Peter!!!11
Electronic Code Book (ECB) Mode

- Identical blocks of plaintext produce identical blocks of ciphertext
- No integrity checks: can mix and match blocks
Cipher Block Chaining (CBC) Mode: Encryption

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

Initialization vector (random)

Sent with ciphertext (preferably encrypted)
Counter Mode (CTR): Encryption

- Initial ctr (random) → ctr → ctr+1 → ctr+2 → ctr+3
- Key
- block cipher
- pt ⊕ Key ⊕ block cipher
- pt ⊕ Key ⊕ block cipher
- pt ⊕ Key ⊕ block cipher
- pt ⊕ Key ⊕ block cipher
- ciphertext

- Identical blocks of plaintext encrypted differently
- Still does not guarantee integrity; Fragile if ctr repeats
Counter Mode (CTR): Decryption

Initial ctr

<table>
<thead>
<tr>
<th>ctr</th>
<th>ctr+1</th>
<th>ctr+2</th>
<th>ctr+3</th>
</tr>
</thead>
<tbody>
<tr>
<td>block cipher</td>
<td>block cipher</td>
<td>block cipher</td>
<td>block cipher</td>
</tr>
<tr>
<td>ct + pt</td>
<td>ct + pt</td>
<td>ct + pt</td>
<td>ct + pt</td>
</tr>
<tr>
<td>Key</td>
<td>Key</td>
<td>Key</td>
<td>Key</td>
</tr>
</tbody>
</table>

4/17/15
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? 

• Fixed mapping from plaintexts to ciphertexts?  
  – What if attacker sees two identical ciphertexts and infers that the corresponding plaintexts are identical?
  – Implication: encryption must be randomized or stateful
How Can a Cipher Be Attacked?

• Attackers knows ciphertext and encryption algthm
  – 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
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
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

- Security against chosen-ciphertext attack (CCA)
  - Integrity protection – it is not possible to change the plaintext by modifying the ciphertext
Why Hide Everything?

• Leaking even a little bit of information about the plaintext can be disastrous

• Electronic voting
  – 2 candidates on the ballot (1 bit to encode the vote)
  – If ciphertext leaks the parity bit of the encrypted plaintext, eavesdropper learns the entire vote

• Also, want a strong definition, that implies other definitions (like not being able to obtain key)
Message Authentication Codes
Encryption schemes: A tool for protecting privacy.

So Far: Achieving 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.
  - NIST recommends a derivative called CMAC [FYI only]
Hash Functions
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 $y$. 

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 1/3 of this classroom 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 $\frac{365}{2} = 182.5$ 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 (informal)
  - Let $t$ be the number of values $x, x', x''$... we need to look at before finding the first pair $x, x'$ s.t. $h(x) = h(x')$
  - What is probability of collision for each pair $x, x'$? $1/2^n$
  - How many pairs would we need to look at before finding the first collision? $O(2^n)$
  - How many pairs $x, x'$ total? $\text{Choose}(t, 2) = \frac{t(t-1)}{2} \sim O(t^2)$
  - What is $t$, the number of values we need to look at? $2^{n/2}$
- 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**
  – Suppose $g$ is one-way
  – Define $h(x)$ as $g(x')$ 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. $\frac{1}{2}$