Symmetric-Key Cryptography

CS 161: Computer Security
Prof. Raluca Ada Popa

Sept 13, 2016
Announcements

• Project due Sept 20
Special guests

- Alice
- Bob
- The attacker (Eve - “eavesdropper”, Malice)
- Sometimes Chris too
Cryptography

• Narrow definition: secure communication over insecure communication channels

• Broad definition: a way to provide formal guarantees in the presence of an attacker
Three main goals

• Confidentiality: preventing adversaries from reading our private data,
• Integrity: preventing attackers from altering some data,
• Authenticity: determining who created a given document
Modern Cryptography

• Symmetric-key cryptography
  – The same secret key is used by both endpoints of a communication

• Public-key (asymmetric-key) cryptography
  – Sender and receiver use different keys
Today: Symmetric-key Cryptography

Whiteboard & notes:
- Symmetric encryption definition
- Security definition
- One time pad (OTP)
- Block cipher
Advanced Encryption Standard (AES)

- Block cipher developed in 1998 by Joan Daemen and Vincent Rijmen
- Recommended by US National Institute for Standard and Technology (NIST)
- Block length $n = 128$, key length $k = 256$
AES ALGORITHM

- 14 cycles of repetition for 256-bit keys.
Algorithm Steps - Sub bytes

- each byte in the *state* matrix is replaced with a SubByte using an 8-bit substitution box
- $b_{ij} = S(a_{ij})$

<table>
<thead>
<tr>
<th>a_{0,0}</th>
<th>a_{0,1}</th>
<th>a_{0,2}</th>
<th>a_{0,3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_{1,0}</td>
<td>a_{1,1}</td>
<td>a_{1,2}</td>
<td>a_{1,3}</td>
</tr>
<tr>
<td>a_{2,0}</td>
<td>a_{2,1}</td>
<td>a_{2,2}</td>
<td>a_{2,3}</td>
</tr>
<tr>
<td>a_{3,0}</td>
<td>a_{3,1}</td>
<td>a_{3,2}</td>
<td>a_{3,3}</td>
</tr>
</tbody>
</table>

SubBytes

<table>
<thead>
<tr>
<th>b_{0,0}</th>
<th>b_{0,1}</th>
<th>b_{0,2}</th>
<th>b_{0,3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>b_{1,0}</td>
<td>b_{1,1}</td>
<td>b_{1,2}</td>
<td>b_{1,3}</td>
</tr>
<tr>
<td>b_{2,0}</td>
<td>b_{2,1}</td>
<td>b_{2,2}</td>
<td>b_{2,3}</td>
</tr>
<tr>
<td>b_{3,0}</td>
<td>b_{3,1}</td>
<td>b_{3,2}</td>
<td>b_{3,3}</td>
</tr>
</tbody>
</table>
### Shift Rows

- Cyclically shifts the bytes in each row by a certain offset
- The number of places each byte is shifted differs for each row

<table>
<thead>
<tr>
<th>No change</th>
<th>( a_{0,0} )</th>
<th>( a_{0,1} )</th>
<th>( a_{0,2} )</th>
<th>( a_{0,3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift 1</td>
<td>( a_{1,0} )</td>
<td>( a_{1,1} )</td>
<td>( a_{1,2} )</td>
<td>( a_{1,3} )</td>
</tr>
<tr>
<td>Shift 2</td>
<td>( a_{2,0} )</td>
<td>( a_{2,1} )</td>
<td>( a_{2,2} )</td>
<td>( a_{2,3} )</td>
</tr>
<tr>
<td>Shift 3</td>
<td>( a_{3,0} )</td>
<td>( a_{3,1} )</td>
<td>( a_{3,2} )</td>
<td>( a_{3,3} )</td>
</tr>
</tbody>
</table>

ShiftRows:
- \( a_{0,0} \)
- \( a_{0,1} \)
- \( a_{0,2} \)
- \( a_{0,3} \)
- \( a_{1,1} \)
- \( a_{1,2} \)
- \( a_{1,3} \)
- \( a_{1,0} \)
- \( a_{2,2} \)
- \( a_{2,3} \)
- \( a_{2,0} \)
- \( a_{2,1} \)
- \( a_{3,3} \)
- \( a_{3,0} \)
- \( a_{3,1} \)
- \( a_{3,2} \)
Uses

- Government Standard
  - AES is standardized as Federal Information Processing Standard 197 (FIPS 197) by NIST
  - To protect classified information
- Industry
  - SSL / TLS
  - SSH
  - WinZip
  - BitLocker
  - Mozilla Thunderbird
  - Skype

But used as part of symmetric-key encryption or other crypto tools
Symmetric-key encryption from block ciphers
Why block ciphers not enough for encryption by themselves?

- Can only encrypt messages of a certain size
- If message is encrypted twice, attacker knows it is the same message
Each block encrypted with a block cipher
Later (identical) message again encrypted
Symmetric key encryption scheme

• Can be reused (unlike OTP)
• Builds on block ciphers:
  – Can be used to encrypt long messages
  – Wants to hide that same block is encrypted twice
• Uses block ciphers in certain modes of operation
Electronic Code Book (ECB)

• Split message $M$ in blocks $P_1, P_2, \ldots$
• Each block is a value which is substituted, like a codebook
• Each block is encoded independently of the other blocks

$$C_i = EK(P_i)$$
Encryption

KeyGen = key gen of block cipher

Enc(K, P1|P2|P3) = (IV, C1, C2, C3)

Dec(K, (IV,C1,C2,C3)) = (P1, P2, P3)
Decryption

Electronic Codebook (ECB) mode decryption

What is the problem with ECB?
Does this achieve IND-KPA?

No, attacker can tell if $P_i = P_j$
Encrypted with ECB
Later (identical) message again encrypted with ECB
CBC: Encryption

IV may not repeat for messages with same $P_1$, choose it at random
CBC: Decryption

Cipher Block Chaining (CBC) mode decryption

Initialization Vector (IV) → Ciphertext → Block Cipher Decryption → Key → Plaintext

\[ P_1, C_1, P_2, C_2, P_3, C_3 \]
CBC

Popular, still widely used
Achieves IND-KPA, and more (IND-CPA)

Caveat: sequential encryption, hard to parallelize

CTR mode gaining popularity
Nonce is similar to IV for CBC, one should not use the same nonce for two messages; choose it at random.
CTR: Decryption

Dec(K, (nonce, C1, C2, C3)) = (P1, P2, P3)

Note, CTR decryption uses block cipher’s encryption, not decryption.
CBC vs CTR

**Security**: Both IND-KPA, and even IND-CPA
If you ever reuse the same nonce, CBC might leak some information about the initial plaintext blocks up to a first difference between two messages. CTR can leak information about various blocks in the message.

**Speed**: Both modes require the same amount of computation, but CTR is parallelizable
Summary

• Encryption protects confidentiality
• IND-KPA is a security game expressing message indistinguishability
• OTP is secure if used only once
• Block ciphers help build symmetric-key encryption schemes with reusable sizes and arbitrary message lengths by chaining them in cipher modes