1-Stream Ciphers

message $\{0, 1\}^n$

key $\{0, 1\}^l$

PRG

output $\{0, 1\}^n$ Bitwise XOR

ciphertext $\{0, 1\}^n$

possible for $L \ll n$

1.1-Two Common Stream Ciphers

1) Block Cipher

Counter mode

Stream Cipher

2) PRG

RC4

ARC4

Stream Cipher

Ron’s Code 4
Key Schedule:

\[
i=0
\]

\[
j=0 \quad 0 \leq i \leq 255
\]

One Pass(i) {

\[
j= (j+S[i]) \mod 256 + k \lfloor i \mod L \rfloor \mod 256
\]

Swap \( S[i] \leftrightarrow S[j] \)  

key length

\[
i++
\]

}

(Sorted Array of Bytes)

(No duplicates)

(Shuffled Array)

internal state (key-dependent)

(Reshuffle)

(Pseudo-Random Selection)
Generator:

Input

\[ i \]
\[ j \]

\[ i = 0 \]
\[ j = 0 \]

While outputting {
\[ i = i + 1 \mod 256 \]
\[ j = (j + S[i]) \mod 256 \]
Swap \( S[i] \leftrightarrow S[j] \)
Output byte \( S[S[i] + S[j] \mod 256] \)
Throw out first 1024 bytes

\[ S = \]
\[ \begin{array}{c}
    \text{beginning} \\
    \downarrow \\
    \vdots \\
    \uparrow \\
    \text{swap} \\
    \downarrow \\
    \text{addition mod 256} \\
    \uparrow \\
    i \\
\end{array} \]

\[ \begin{array}{c}
    S[i] \\
    \downarrow \\
    j \\
\end{array} \]
Encrypt:

key → Key Schedule → S → Generation → output + ciphertext

Bitwise XOR

Decrypt:

key → Key Schedule → S → Generation → output + message

output → PRG(key)

deterministic (pseudo-random)

2-Block Ciphers

\[ C = \text{Enc}_k(m) \]

\[ m \in \{0,1\}^B \text{ block of a message, } B=128 \text{ bits} \]

\[ c \in \{0,1\}^L \]

\[ k \in \{0,1\}^l \text{ 128/256} \]

\[ m = \text{Dec}_k(c) \]

same key

Alice → Bob

Alice \{ k \ c=\text{Enc}_k(m) \} \rightarrow Bob \{ k \}

Alice knows \ m \ c=\text{Enc}_k(m) \}

Bob knows \ c \ m=\text{Dec}_k(c) \}
Stream Ciphers

- Encrypting bit-by-bit

Stream Cipher:

\[ m = 0 1 1 1 0 0 \quad \text{functional dependency} \]

\[ c = 1 0 1 1 0 1 \quad \text{value} \]

Block Cipher:

\[ m = \begin{array}{c}
0 1 1 1 0 0 \\
1 1 0 0
\end{array} \]

\[ c = \begin{array}{c}
1 0 1 1 0 1
\end{array} \]

Block Ciphers

- Fixed Input
- Compression
- Construction
- Arbitrarily
- Chaining
- lengthed input

Hash Function

Function

Markle-Damgard

Block Ciphers

- Sponge
- modes of operation

DES, AES

ECB, CBC, CM, GCM

AES-CBC-128

Block Ciphers

- confusing
- key length
Block Ciphers:

<table>
<thead>
<tr>
<th>Name</th>
<th>(Effective) key length</th>
<th>$10000 \rightarrow$ crack in 1 day</th>
<th>Insecure: key too short</th>
<th>Insecure: better attacks than brute-force of key</th>
<th>NIST $O(2^{112})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>40/56</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3DES</td>
<td>112/168</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES</td>
<td>128/192/256</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Block size: 128)

Aside:

Kirchhoff’s Principle:

[Paraphrase]: A crypto-system should be secure even if everything about the system except the key, is public knowledge.

[Short]: The enemy knows the system.

3-Modes of operation

* allows encryption of any number of blocks

Electric Codebook (ECB)

```
key
256

Encryption
128 bits
m
Encryption
128 bits
m

Block Cipher
128
128
```

NIST $O(2^{112})$
Encryption (Image)

Bitmap, each pixel is 128-bits

Blue = 0x....... (128-bits)

Red = 0x....... (128-bits)