Cryptography

2 – Secret-key encryption: Block ciphers

G. Chênevert

September 23, 2019

ISEN
ALL IS DIGITAL!
LILLE
yncréa

Today

Malleability

Block ciphers

Construction

Modes of operation

Stream ciphers

Recall: A binary stream cipher is a symmetric cipher (E, D) with

$$E(k, x) = D(k, x) = x \oplus G(k)$$

where $G : \{0,1\}^m \to \{0,1\}^*$ is a CSPRNG.

Advantage: simplicity

Inconvenient: simplicity!

Example

Suppose Alice during a heated conversation sends to Bob:

$$m = \texttt{no, I'm not angry!}$$

encoded in ASCII then encrypted with Salsa20 using a 128-bit secret key.

Different attackers

Eve (a passive attacker): sees the ciphertext, learns nothing ✓

Oscar (an active attacker): is able to modify the ciphertext, may have a different goal !

Challenge

Find a way for Oscar (who doesn't know the secret key) to make it look like Eve is shouting by making sure Bob receives the message

$$m' = \texttt{NO, I'M NOT ANGRY!}$$

Hints

- the ASCII codes for upper- and lower-case letters differ by exactly 1 bit

  ex.: $\texttt{A} = 65 = 01\mathbf{0}00001$, $\texttt{a} = 97 = 01\mathbf{1}00001$

  NB: Space $= 32 = 00\mathbf{1}00000$ thus $\texttt{A} = \texttt{a} \oplus$ Space !

- **Malleability**:

$$E(k, m \oplus y) = E(k, m) \oplus y$$
$$D(k, c \oplus y) = D(k, c) \oplus y$$

  Can be seen as a weakness – or a feature!

Today

Malleability

Block ciphers

Construction

Modes of operation

New point of view

Consider (E, D) a symmetric cipher with $\mathcal{M} = \mathcal{C}$.

For given $k \in \mathcal{K}$,

$$E_k := E(k, \cdot\,) : \mathcal{M} \longrightarrow \mathcal{M}$$

admits $D_k := D(k, \cdot\,)$ as inverse

hence E_k is a **permutation** of $\mathcal{M}$ (bijection from $\mathcal{M}$ to $\mathcal{M}$)

Set of all permutations of a set X denoted $\mathfrak{S}_X$, size $|X|!$

Idea

E_k should be thought of as a *pseudo-random permutation* of $\mathcal{M}$.

In practice: undistinguishable from a random *function* $\mathcal{M} \to \mathcal{M}$.

Allows one to:

- reuse keys (with some care!)
- work with small messages (**blocks**)

Note: typically $|\mathcal{K}| \ll |\mathfrak{S}_{\mathcal{M}}| = |\mathcal{M}|! \approx |\mathcal{M}|^{|\mathcal{M}|}$

ex.: $|\mathcal{K}| = |\mathcal{M}| = 2^{128}$, $|\mathfrak{S}_{\mathcal{M}}| \approx |\mathcal{M}|^{|\mathcal{M}|} \approx 2^{43556142965880123323311949751266331066368}$ (!)

But I liked stream ciphers!

Block ciphers are typically slower, but

- can avoid linearity (malleability) problems;
- are much more versatile!

For example: you can get a *parallelizable* CSPRNG for free from a block cipher E

$$G(k) := E(k, 0) \parallel E(k, 1) \parallel E(k, 2) \parallel \cdots$$

Today

Malleability

Block ciphers

Construction

Modes of operation

Do these things actually exist?

Shannon's paradigm: *confusion* and *diffusion* (stream ciphers miss the diffusion part)

Essentially all modern examples use an iterative design where the plaintext is encrypted a certain number of times by a **round function** performing (a small amount of) confusion and diffusion

$$\begin{cases} x_0 = m, \\ x_{i+1} = R(k_i, x_i), \quad 0 \leq i < r \\ E(k, m) = x_r \end{cases}$$

preceded by a **key scheduling** process $k \mapsto (k_1, \cdots, k_r)$.

Famous examples

n-bit block, k-bit key, r rounds

- **Lucifer** (IBM, 1971) $n = k = 128$, $r = 16$
- **Data Encryption Standard** (NIST, 1977) $n = 64$, $k = 56$, $r = 16$

  Successful brute force attack in 1997!

  Still survives in the form of Triple DES for legacy hardware/software
- **Rijndael** (KU Leuven, 1998) *aka* **Advanced Encryption Standard** (NIST, 2000)

  $n = 128$, $k \in \{128, 192, 256\}$, $r \in \{10, 12, 14\}$.

But also: RC5/RC6, IDEA, Serpent, Blowfish/Twofish, ...

Design of DES

16-round **Feistel network** :

Write each $x_i = \ell_i \parallel r_i$

Round function:

$$\begin{cases} \ell_{i+1} = r_i \\ r_{i+1} = \ell_i \oplus F(k_i, r_i) \end{cases}$$

Easy to implement in hardware (and invert – exercise!)

Security proof

Theorem (Luby-Rackhoff, 1988)

Three rounds of a Feistel network with inner function F a CSPRNG using k as a seed is computationally undistinguishable from a random permutation.

In practice: increase the number of rounds to take into account the fact that F might not be a provably good CSPRNG.

Still: the original DES can now be broken by exhaustive key search in a couple of days (hours?) with COPACOBANA

Variants still in use

- **Triple DES**:

$$3E((k_1, k_2, k_3), m) := E(k_1, D(k_2, E(k_3, m)))$$

  Contains plain DES as the special case $k_1 = k_2 = k_3$

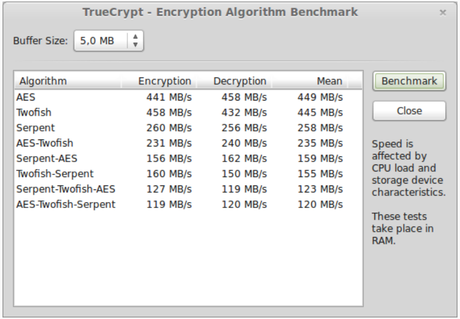
  (Double DES susceptible to a *meet-in-the-middle* attack)

- **DESX**:

$$EX((k_1, k_2, k_3), m) := k_1 \oplus E(k_2, m \oplus k_3)$$

Today

In practice: **use AES** or some other NIST finalist



Today

Malleability

Block ciphers

Construction

Modes of operation

Modes of operation

Now suppose the message to be encrypted is longer than a single block:

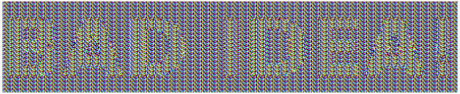
$$m = m_1 \parallel m_2 \parallel m_3 \parallel \; \cdots$$

How to use the block cipher ?

- **Electronic Code Book (ECB) mode**:

$$\begin{cases} c_i = E(k, m_i) \\ m_i = D(k, c_i) \end{cases}$$

ECB mode?



Problem: equal blocks yield equal ciphertexts

Should use (pseudo-)**probabilistic encryption**:

a given block shouldn't always have the same encryption

$\rightsquigarrow$ use of either *random value* or *nonce* (counter)

Let's look at two of the simplest ways to do it:

Cipher Block Chaining (CBC) mode

$$\begin{cases} c_0 = \text{random } \textbf{I}\text{nitial } \textbf{V}\text{alue} \\ c_i = E(k, m_i \oplus c_{i-1}) \end{cases}$$

- Encryption is sequential (but decryption can be parallelized)

$$m_i = D(k, c_i) \oplus c_{i-1}$$

- Message has to be padded to a multiple of the block length
- Crucial that random IV is non-predictable (chosen plaintext attack)

Randomized counter (CTR) mode

$$\begin{cases} c_0 = \text{random IV} \\ c_i = m_i \oplus E(k, c_0 + i) \end{cases}$$

- Block cipher is effectively turned into a stream cipher
- No padding problem
- Highly parallelizable
- Random IV prevents reuse of key stream

Other modes

Many other modes that achieve specific goals exist.

- feedback modes: CFB, OFB, . . .
- authenticated encryption: OCB, EAX, GCM, . . .
- device encryption: LRW, XEX, XTS, . . .