TOPIC: 6.857

DATE: 3/2/15

FILE UNDER: Secret-sharing & block ciphers

PAGE: L08.1

Admin: MIT Bitcoin Expo March 7-8

Today:

- Shamir's secret sharing
- Block ciphers
  - DES
  - AES
  - Modes of operation (ECB, CTR, CBC, CFB)

Readings:

Ferguson et al.: Chapter 3, Chapter 21.9

Paar & Pelzl: Chapters 3, 4

Katz/Lindell: Chapters 6.2.3, 6.2.5, 13.3

Project idea:

Do a source-code review of an open-source implementation of a crypto library or crypto product.

Key management

Start with "secret sharing" (threshold cryptography).

- Assume Alice has a secret s. (e.g. a key)
- She wants to protect s as follows:

She has n friends $A_1, A_2, \ldots, A_n$

She picks a "threshold" t, $1 \le t \le n$.

She wants to give each friend A_i,

a "share" s_i of s, so that

- any t or more friends can reconstruct s
- any set of $< t$ friends can not.

ref: bitcoin "multisig"

Easy cases:

$t = 1$: $s_i = s$

$t = n$: $s_1, s_2, \ldots, s_{n-1}$ random

s_n chosen so that

$$s = s_1 \oplus s_2 \oplus \cdots \oplus s_n$$

What about $1 < t < n$?

Shamir's method ("How to Share a Secret", 1979)

Idea: 2 points determine a line
3 points determine a quadratic
...
t points determine a degree $(t-1)$ curve

Let $f(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1 x + a_0$

There are t coefficients. Let's work modulo prime p.

We can have t points: (x_i, y_i) for $1 \le i \le t$

They determine coefficients, and vice versa.

$$
\begin{tikzcd}
\{(x_i, y_i)\} \arrow[r, bend right, "\text{Polynomial Interpolation}"'] & (a_{t-1}, a_{t-2}, \ldots, a_1, a_0) \arrow[l, bend right, "\text{Polynomial Evaluation}"']
\end{tikzcd}
$$

Pt/value pairs — Coefficients

To share secret s (here $0 \le s < p$):

Let $y_0 = a_0 = s$

Pick $a_1, a_2, \ldots, a_{t-1}$ at random from $\mathbb{Z}_p$

Let share $s_i = (i, y_i)$ where $y_i = f(i)$, $1 \le i \le n$.

Evaluation is easy.

TOPIC:

FILE UNDER:

DATE:

Interpolation

Given (x_i, y_i) $1 \le i \le t$ (wlog)

Then $f(x) = \sum_{i=1}^{t} f_i(x) \cdot y_i$

where $f_i(x) = \begin{cases} 1 & \text{at } x = x_i \\ 0 & \text{for } x = x_j,\ j \ne i,\ 1 \le j \le t \end{cases}$

Furthermore:

$$f_i(x) = \frac{\prod_{j \ne i} (x - x_j)}{\prod_{j \ne i} (x_i - x_j)}$$

This is a polynomial of degree $t-1$. So f also has degree $t-1$.

Evaluating $f(0)$ to get s simplifies to

$$s = f(0) = \sum_{i=1}^{t} y_i \cdot \frac{\prod_{j \ne i} (-x_j)}{\prod_{j \ne i} (x_i - x_j)}.$$

Theorem: Secret sharing with Shamir's method is information-theoretically secure. Adversary with $< t$ shares has no information about s.

Pf: A degree $t-1$ curve can go through any point $(0, s)$ as well as any given d pts (x_i, y_i), if $d < t$. ∎

Refs: Reed-Solomon codes, erasure codes, error correction, information dispersal (Rabin).

TOPIC:

DATE:

FILE UNDER:

PAGE: L5.2

Block ciphers:

fixed-length P, C, K

DES: $|P| = |C| = 64$ bits $|K| = 56$ bits

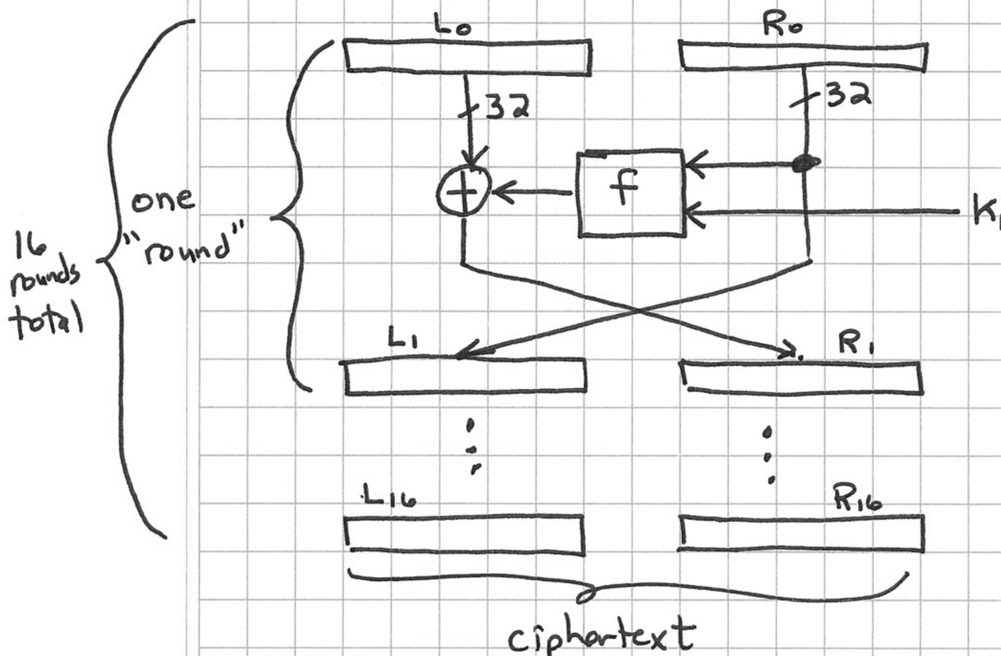
AES: $|P| = |C| = 128$ bits $|K| = 128, 192, 256$ bits

Use a "mode of operation" to handle variable-length input.

DES

"Data Encryption Standard"
Standardized in 1976. Now deprecated in favor of AES.

"Feistel structure":



plaintext 64 bits

"round key"

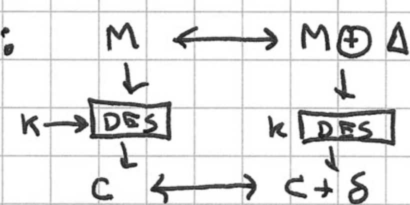
all 16 round keys derived from 64-bit encryption key (only 56 bits are really used) via "key schedule"

Note: Invertible for any f and any key schedule.

f uses 8 "S-boxes" mapping 6-bits ⇒ 4 bits nonlinearly.

Key is too short! (Breakable now quite easily by brute-force)

Subject to differential attacks:



2^{47} chosen pairs (Biham/Shamir)

Subject to linear attacks:

e.g. if $M_3 \oplus M_{15} \oplus C_2 \oplus K_{14} = 0$ (eqn on bits)
with prob $p = 1/2 + \varepsilon$

then need $1/\varepsilon^2$ samples to break (Matsui, 2^{43} PT/CT pairs)

TOPIC:

DATE:

FILE UNDER:

AES

"Advanced Encryption Standard" (U.S. govt)

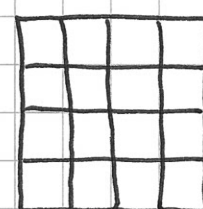
Replaces DES

AES "contest" 1997–1999:
15 algorithms submitted: RC6, Mars, Twofish, Rijndael, ...
winner = Rijndael (by Joan Daemen & Vincent Rijmen, (Belgians))

Specs: 128-bit plaintext/ciphertext blocks
128, 192, or 256-bit key
10, 12, or 14 rounds (dep. on key length)

Byte-oriented design (some math done in Galois field $GF(2^8)$)

View input as 4x4 byte array:



$4 \times 4 \times 8 = 128$

For version with 128-bit keys, 10 rounds:

- Derive 11 "round keys", each 128 bits (4x4x byte)
- In each round:
  1. XOR round key
  2. Substitute bytes (lookup table)
  3. Rotate rows (by different amts)
  4. Mix each column (by linear opn)

(last round has another round key xored in instead of mix-column)

- Output final state

See readings for details.
There are very fast implementations. Also Intel has put supporting hardware into its CPU's.

Security: Good; perhaps # rounds should be a bit larger...