CS 120/CSCI E-177: Introduction to Cryptography

Salil Vadhan and Alon Rosen Nov. 7, 2006

Lecture Notes 13:

Private-Key Encryption in Practice

Recommended Reading.

- Katz–Lindell, Chapter 5.
- FIPS publication describing DES (link on webpage).
- FIPS publication describing AES.

1 Stream Ciphers vs. Block Ciphers

- Unlike what we've seen, private-key (aka symmetric) encryption schemes used in practice generally
  - are not be based on nice computational problems
  - are not proven secure via reductions
  - are designed for a particular input length (so can only be treated with concrete security)
  - but are extremely efficient
- Stream Ciphers
  - Essentially meant to be pseudorandom generators, used for stateful encryption.
  - Examples: linear feedback shift registers (not secure, but used as component in better stream ciphers), RC4, SEAL, ...
  - Extremely simple and fast
  - Practical issues: can generate pseudorandom bits offline and decrypt very quickly without buffering, but requires synchronization
- Block ciphers
  - For every key $k \in \{0,1\}^n$, $E_k : \{0,1\}^\ell \to \{0,1\}^\ell$ is a permutation, and both E_k and E_k^{-1} can be computed quickly given k. (n=*key length*, ℓ = *block length*)
  - Examples: DES, AES/Rijndael, IDEA, ...
  - Main tools for private-key encryption in practice.
  - Have both stateless modes and stateful/stream-like modes.
- How are they designed?
  - More of an art than science. Intuition/experience of designers, public critique important.

- "Diffusion": have each output bit affected by many input bits, each input bit influence many output bits — often achieved by repeating many "rounds" that involve swapping bits.
- "Confusion": avoid structured relationships (especially *linearity*) between input and output/key that are exploited in known attacks.
- Output should be "random-looking", have good statistical properties.
- Simplicity.
- Efficiency — extremely fast in hardware & software on wide variety of platforms.

2 The Data Encryption Standard (DES)

- Designed by IBM and the NSA, standardized in 1977.
- Most widespread block cipher — used by federal agencies, banks (ATM machines), SSL, ...
- Key length 56, block length 64.
- Computation of $\mathrm{DES}_k(m)$ is done by 16-round *Feistel network*:
  - $(\ell_0, r_0) \in \{0,1\}^{32} \times \{0,1\}^{32}$ is fixed permutation of bit positions of m.
  - $(\ell_i, r_i) = (r_{i-1}, \ell_{i-1} \oplus f_{k_{i-1}}(r_{i-1}))$. *Feistel transformation*
  - Subkey $k_{i-1} \in \{0,1\}^{48}$ consists of fixed permuted subset of bits of k.
  - Computation of round function $f_k(r)$:
    - $r \in \{0,1\}^{32}$ expanded to $E(r) \in \{0,1\}^{48}$ by repeating some bits and permuting bits.
    - $E(r) \oplus k$ broken into 6-bit blocks $B_1, \ldots, B_8$.
    - $C_j = S_j(B_j)$ for hardwired "S-box" $S_j : \{0,1\}^6 \to \{0,1\}^4$. (*Main source of DES's security.*)
    - $f_k(r)$ = fixed permutation of bits of $C_1 \cdots C_8$.
- Inversion: each Feistel transformation is a permutation, the inverse of the Feistel transformation is easy to compute given the subkey.
- Speed: $\approx$ 10 Mbits/sec in software, $>$ 1 Gbit/sec in hardware!

3 Modes of Operation

- Described for DES, but apply to any block cipher.
- Electronic Codebook Mode (ECB Mode): To encrypt message m, break m into blocks $m_1, m_2, \ldots$ of size 64, output $c_1, c_2, \ldots$, where $c_i = \mathrm{DES}_k(m_i)$.
- Cipher-Block Chaining Mode (CBC Mode): $c_0 \xleftarrow{R} \{0,1\}^{64}$, $c_i = \mathrm{DES}_k(c_{i-1} \oplus m_i)$.
- Counter Mode (CTR Mode): $c_i = \mathrm{DES}_k(\mathrm{IV} + i \bmod 2^{64}) \oplus m_i$.
- Output Feedback Mode (OFB Mode): $c_i = m_i \oplus z_i$, where $z_0 \xleftarrow{R} \{0,1\}^{64}$, $z_i = \mathrm{DES}_k(z_{i-1})$. (Stream cipher).

4 Cryptanalysis of DES

- Typically focuses on key recovery — given $(m_1, \mathrm{DES}_k(m_1)), \ldots, (m_q, \mathrm{DES}_k(m_q))$, find k. the pairs (m_i, c_i) are obtained by known plaintext attack, chosen plaintext attack or by listening on the communication channel. We have seen that the security against key recovery is not sufficient but necessary.
- Exhaustive search: $q = 2$ suffices, time $= \Theta(2^k)$. For all keys k, if $c_1 = \mathrm{DES}_k(m_1)$ and $c_2 = \mathrm{DES}_k(m_2)$, output k.
  - 1993: 56 hours using specialized \$250K key-search machine.
  - A possible solution against the fact that the keyspace is small is to use several DES keys. "Triple DES" $(\mathrm{DES}_{k_1}(\mathrm{DES}^{-1}_{k_2}(\mathrm{DES}_{k_3}(m))))$ seems better, but not as much as one might hope. (the reason for the inverse is for "backwards compatibility": when $k_2 = k_3$, we get single DES)
- Linear and Differential cryptanalysis: $q = 2^{43}$ for DES. The idea is to try to find relation between the input bits and the output bits, e.g. if the inputs differ by one bit, do we have correlation between the outputs?
  - Impractical against DES — too much memory required.
  - But devastating for 8-round DES and other block ciphers.

5 Advanced Encryption Standard (AES)

- Development initiated in 1997 by the National Institute of Standards and Technology (NIST). 15 submissions, several rounds of public analysis/discussion. *Rijndael* selected in October 2000, soon to be standardized.
- Block size = 128, variable key length = 128, 192, or 256, variable number of rounds = 10, 12, or 14, respectively.
- Computation of $\mathrm{AES}_k(m)$:
  - m written as 4×4 matrix of 8-bit values (viewed as elements of finite field $\mathbb{F}$ of size 2^8).
  - Each round consists of:
    - Applying a substitution (S-box) to each matrix entry. (Inversion and affine transformation in $\mathbb{F}$: $S(x) = ax^{-1} + b$ where a, b fixed) S and S^{-1} are easy to compute.
    - Left-shifting each row (0, 1, 2, 3 positions, respectively).
    - An operation on each column (polynomial arithmetic over $\mathbb{F}$: each column is viewed as a polynomial). This column substitution C and its inverse C^{-1} are easy to compute.
    - XORing entire matrix with the round key k_i.
  - Round key generation:
    - Actual k written as initial part of an array, each entry consisting of a *word* — four 8-bit values, again viewed as elements of $\mathbb{F}$.
    - Usually, next word is XOR of previous word and an earlier word.

    * Sometimes, next word formed by a left-shift and S-box applied to entries of previous word.

- Design criteria:
  - Protection against known attack methods (e.g. linear and differential cryptanalysis).
  - Efficiency (hardware and software) on wide range of platforms.
  - Simplicity.
- Only time will tell how good it really is!

6 Theory vs. Practice

- Why do people use block ciphers over "provable" constructions?
  - Efficiency — provable constructions much slower (modular arithmetic), require larger keys. For a key length k, the security of block ciphers seem to grow like $\Theta(2^k)$ and the security of provable number-theoretic constructions seems to grow like $2^{\Theta(k^{1/3})}$
  - History — block ciphers standardized before modern cryptography developed (in contrast to public-key crypto).

  How can we bridge the gap?
- Approach 1 (Most common): Model block ciphers as families of *pseudorandom permutations* (as in Bellare–Rogaway)
  + Can critique existing uses of block ciphers, e.g. some modes of operation secure (like CBC, CTR), some insecure (like ECB).
  + Can give some justification for Feistel network (converts PRFs to PRPs).
  − A very strong assumption, hard to evaluate!
- Approach 2 (Not so common): View block ciphers as one-way functions and apply provable constructions.
  + OWF assumption much weaker, easier to evaluate.
  − Resulting constructions unlikely to be as efficient.
  − Possibly not using full strength of block ciphers.
- Approach 3 (Occassionally used): Forget modelling and proofs, just take main ideas & understanding of goals from theory.