DD2448 Foundations of Cryptography
Lecture 4

Douglas Wikström
KTH Royal Institute of Technology
dog@kth.se

March 24, 2020

Last Lecture: Linear Cryptanalysis Summary

1. Find linear approximation of S-Boxes.
2. Compute bias of each approximation.
3. Find linear trails.
4. Compute bias of linear trails.
5. Compute data and time complexity.
6. Estimate key bits from many plaintext-ciphertexts pairs.

Linear cryptanalysis is a **known plaintext attack**.

Ideal Block Cipher

Negligible Functions

Definition. A function $\epsilon(n)$ is negligible if for every constant $c > 0$, there exists a constant n_0, such that

$$\epsilon(n) < \frac{1}{n^c}$$

for all $n \geq n_0$.

Motivation. Events happening with negligible probability can not be exploited by polynomial time algorithms! (they "never" happen)

Negligible Functions

Definition. A function $\epsilon(n)$ is negligible if for every constant $c > 0$, there exists a constant n_0, such that

$$\epsilon(n) < \frac{1}{n^c}$$

for all $n \geq n_0$.

Motivation. Events happening with negligible probability can not be exploited by polynomial time algorithms! (they "never" happen)

Caveat! Theoretic notion. Interpret with care in practice.

Pseudo-Random Function

"Definition". A function is pseudo-random if no efficient adversary can distinguish between the function and a random function.

Pseudo-Random Function

"Definition". A function is pseudo-random if no efficient adversary can distinguish between the function and a random function.

Definition. A family of functions $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is pseudo-random if for all polynomial time oracle adversaries A

$$\left| \Pr_K \left[A^{F_K(\cdot)} = 1 \right] - \Pr_{R:\{0,1\}^n \to \{0,1\}^n} \left[A^{R(\cdot)} = 1 \right] \right|$$

is negligible.

Pseudo-Random Permutation

"Definition". A permutation and its inverse is pseudo-random if no efficient adversary can distinguish between the permutation and its inverse, and a random permutation and its inverse.

Pseudo-Random Permutation

"Definition". A permutation and its inverse is pseudo-random if no efficient adversary can distinguish between the permutation and its inverse, and a random permutation and its inverse.

Definition. A family of permutations $P : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ are pseudo-random if for all polynomial time oracle adversaries A

$$\left| \Pr_K \left[A^{P_K(\cdot), P_K^{-1}(\cdot)} = 1 \right] - \Pr_{\Pi \in \mathcal{S}_{2^n}} \left[A^{\Pi(\cdot), \Pi^{-1}(\cdot)} = 1 \right] \right|$$

is negligible, where $\mathcal{S}_{2^n}$ is the set of permutations of $\{0,1\}^n$.

Idealized Four-Round Feistel Network

Definition. Feistel round (H for "Horst Feistel").

$$H_{F_K}(L, R) = (R, L \oplus F(R, K))$$

Idealized Four-Round Feistel Network

Definition. Feistel round (H for "Horst Feistel").

$$H_{F_K}(L, R) = (R, L \oplus F(R, K))$$

Theorem. (Luby and Rackoff) If F is a pseudo-random family of functions, then

$$H_{F_{k_1}, F_{k_2}, F_{k_3}, F_{k_4}}(x) = H_{F_{k_4}}(H_{F_{k_3}}(H_{F_{k_2}}(H_{F_{k_1}}(x))))$$

(and its inverse) is a pseudo-random family of permutations.

Idealized Four-Round Feistel Network

Definition. Feistel round (H for "Horst Feistel").

$$H_{F_K}(L, R) = (R, L \oplus F(R, K))$$

Theorem. (Luby and Rackoff) If F is a pseudo-random family of functions, then

$$H_{F_{k_1}, F_{k_2}, F_{k_3}, F_{k_4}}(x) = H_{F_{k_4}}(H_{F_{k_3}}(H_{F_{k_2}}(H_{F_{k_1}}(x))))$$

(and its inverse) is a pseudo-random family of permutations.

Why do we need four rounds?

Perfect Secrecy

Perfect Secrecy (1/3)

When is a cipher perfectly secure?

Perfect Secrecy (1/3)

When is a cipher perfectly secure?

How should we formalize this?

Perfect Secrecy (2/3)

Definition. A cryptosystem has perfect secrecy if guessing the plaintext is as hard to do given the ciphertext as it is without it.

Perfect Secrecy (2/3)

Definition. A cryptosystem has perfect secrecy if guessing the plaintext is as hard to do given the ciphertext as it is without it.

Definition. A cryptosystem has perfect secrecy if

$$\Pr[M = m \,|\, C = c] = \Pr[M = m]$$

for every $m \in \mathcal{M}$ and $c \in \mathcal{C}$, where M and C are random variables taking values over $\mathcal{M}$ and $\mathcal{C}$.

Perfect Secrecy (3/3)

Game Based Definition. Exp_A^b, where A is a strategy:

1. $\mathsf{k} \leftarrow_R \mathcal{K}$
2. $(m_0, m_1) \leftarrow A$
3. $c = \mathsf{E}_{\mathsf{k}}(m_b)$
4. $d \leftarrow A(c)$, with $d \in \{0, 1\}$
5. Output d.

Definition. A cryptosystem has perfect secrecy if for every **computationally unbounded** strategy A,

$$\Pr\left[\mathrm{Exp}_A^0 = 1\right] = \Pr\left[\mathrm{Exp}_A^1 = 1\right] \quad .$$

One-Time Pad

One-Time Pad (OTP).

- **Key.** Random tuple $\mathsf{k} = (b_0, \ldots, b_{n-1}) \in \mathbb{Z}_2^n$.
- **Encrypt.** Plaintext $m = (m_0, \ldots, m_{n-1}) \in \mathbb{Z}_2^n$ gives ciphertext $c = (c_0, \ldots, c_{n-1})$, where $c_i = m_i \oplus b_i$.
- **Decrypt.** Ciphertext $c = (c_0, \ldots, c_{n-1}) \in \mathbb{Z}_2^n$ gives plaintext $m = (m_0, \ldots, m_{n-1})$, where $m_i = c_i \oplus b_i$.

Bayes' Theorem

Theorem. If A and B are events and $\Pr[B] > 0$, then

$$\Pr[A \mid B] = \frac{\Pr[A] \Pr[B \mid A]}{\Pr[B]}$$

Terminology:

$\Pr[A]$ – prior probability of A
$\Pr[B]$ – prior probability of B
$\Pr[A \mid B]$ – posterior probability of A given B
$\Pr[B \mid A]$ – posterior probability of B given A

One-Time Pad Has Perfect Secrecy

- **Probabilistic Argument.** Bayes implies that:

$$\begin{aligned}\Pr[M = m \,|\, C = c] &= \frac{\Pr[M = m]\Pr[C = c \,|\, M = m]}{\Pr[C = c]} \\ &= \Pr[M = m]\,\frac{2^{-n}}{2^{-n}} \\ &= \Pr[M = m] \quad .\end{aligned}$$

- **Simulation Argument.** The ciphertext is uniformly and independently distributed from the plaintext. We can **simulate** it on our own!

Bad News

Theorem. "For every cipher with perfect secrecy, the key requires at least as much space to represent as the plaintext."

Dangerous in practice to rely on no reuse of, e.g., file containing randomness!

Universal Hash Functions

Universal Hash Function

Definition. An ensemble $f = \{f_\alpha\}$ of hash functions $f_\alpha : X \to Y$ is (strongly) 2-universal if for every $x, x' \in X$ and $y, y' \in Y$ with $x \neq x'$ and a random α

$$\Pr[f_\alpha(x) = y \wedge f_\alpha(x') = y'] = \frac{1}{|Y|^2} \ .$$

Universal Hash Function

Definition. An ensemble $f = \{f_\alpha\}$ of hash functions $f_\alpha : X \to Y$ is (strongly) 2-universal if for every $x, x' \in X$ and $y, y' \in Y$ with $x \neq x'$ and a random α

$$\Pr[f_\alpha(x) = y \wedge f_\alpha(x') = y'] = \frac{1}{|Y|^2} \ .$$

I.e., for any fixed $x' \neq x$, the outputs $f_\alpha(x)$ and $f_\alpha(x')$ are uniformly and independently distributed when α is chosen randomly.

In particular x and x' are both mapped to the same value with probability $1/|Y|$.

Example

Example. The function $f : \mathbb{Z}_p \to \mathbb{Z}_p$ for prime p defined by

$$f(z) = az + b \bmod p$$

is strongly 2-universal.

Proof. Let $x, x', y, y' \in \mathbb{Z}_p$ with $x \neq x'$. Then

$$\begin{pmatrix} x & 1 \\ x' & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} y \\ y' \end{pmatrix}$$

has a unique solution. Random (a, b) satisfies this solution with probability $\frac{1}{p^2}$.

Universal Hash Function

Universal hash functions are **not** one-way or collision resistant!

Hash Functions

Hash Function

A hash function maps arbitrarily long bit strings into bit strings of fixed length.

The output of a hash function should be “unpredictable”.

Wish List

- Finding a pre-image of an output should be hard.
- Finding two inputs giving the same output should be hard.
- The output of the function should be “random”.

etc

Ensembles of Functions (1/3)

- Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial time computable function.

Ensembles of Functions (1/3)

- Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial time computable function.
- We can derive an ensemble $\{f_n\}_{n \in \mathbb{N}}$, with

$$f_n : \{0,1\}^n \to \{0,1\}^*$$

by setting $f_n(x) = f(x)$.

Ensembles of Functions (1/3)

- Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial time computable function.
- We can derive an ensemble $\{f_n\}_{n \in \mathbb{N}}$, with

$$f_n : \{0,1\}^n \to \{0,1\}^*$$

  by setting $f_n(x) = f(x)$.
- Note that we may recover f from the ensemble by $f(x) = f_{|x|}(x)$.

Ensembles of Functions (1/3)

- Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial time computable function.
- We can derive an ensemble $\{f_n\}_{n \in \mathbb{N}}$, with

$$f_n : \{0,1\}^n \to \{0,1\}^*$$

  by setting $f_n(x) = f(x)$.
- Note that we may recover f from the ensemble by $f(x) = f_{|x|}(x)$.
- When convenient we give definitions for a function, but it can be turned into a definition for an ensemble.

Ensembles of Functions (2/3)

- Consider $F = \{f_n\}_{n \in \mathbb{N}}$, where f_n is itself an ensemble $\{f_{n,\alpha_n}\}_{\alpha_n \in \{0,1\}^n}$, with

$$f_{n,\alpha_n} : \{0,1\}^{l(n)} \to \{0,1\}^{l'(n)}$$

for some length polynomials $l(n)$ and $l'(n)$.

Ensembles of Functions (2/3)

- Consider $F = \{f_n\}_{n \in \mathbb{N}}$, where f_n is itself an ensemble $\{f_{n,\alpha_n}\}_{\alpha_n \in \{0,1\}^n}$, with

$$f_{n,\alpha_n} : \{0,1\}^{l(n)} \to \{0,1\}^{l'(n)}$$

  for some length polynomials $l(n)$ and $l'(n)$.

- Here n is the security parameter and α_n is a "key" that is chosen randomly.

Ensembles of Functions (2/3)

- Consider $F = \{f_n\}_{n\in\mathbb{N}}$, where f_n is itself an ensemble $\{f_{n,\alpha_n}\}_{\alpha_n\in\{0,1\}^n}$, with

$$f_{n,\alpha_n} : \{0,1\}^{l(n)} \to \{0,1\}^{l'(n)}$$

  for some length polynomials $l(n)$ and $l'(n)$.

- Here n is the security parameter and α_n is a "key" that is chosen randomly.

- We may also view F as an ensemble $\{f_\alpha\}$, where $f_\alpha = \{f_{n,\alpha_n}\}_{n\in\mathbb{N}}$ and $\alpha = \{\alpha_n\}_{n\in\mathbb{N}}$.

Ensembles of Functions (3/3)

These conventions allow us to talk about what in everyday language is a "function" f in several convenient ways.

Now you can forget that and assume that everything works!

One-Wayness

Definition. A function $f : \{0,1\}^* \to \{0,1\}^*$ is said to be **one-way**[1] if for every polynomial time algorithm A and a random x

$$\Pr[A(f(x)) = x' \wedge f(x') = f(x)] < \epsilon(n)$$

for a negligible function ϵ.

Normally f is computable in polynomial time in its input size.

[1] "Enkelriktad" på svenska **inte** "enväg".