Lecture Notes: Linear Cryptanalysis (LC)

Course: INFO-F-537 Cryptanalysis (Topic 3)
Document compiled by Luca Volonterio

Based on: *INFOF537-Notes-LC* & *SecretKeyPrimitive*

January 12, 2026

Abstract

This document covers **Topic 3** for the oral exam. Linear Cryptanalysis (LC) is a statistical **Known-Plaintext Attack (KPA)** that approximates the non-linear behavior of a cipher (S-boxes) using linear equations. This guide explains the concept of correlations, linear approximations, the Piling-Up Lemma, how to detect and exploit linear biases (Matsui's algorithms), how to build a PRP distinguisher, and the defense mechanisms used in modern ciphers like AES via the Wide Trail Strategy.

Contents

1 Introduction: Definitions and Scope

1.1 What is Linear Cryptanalysis?

Linear Cryptanalysis (LC) is a **shortcut attack** (white-box setting) introduced by Mitsuru Matsui. It targets block ciphers by exploiting statistical biases in linear approximations of the cipher's operations.

- **Core Idea:** Instead of looking for differences (as in DC), we look for **linear relationships** (modulo 2) between bits of the plaintext P, ciphertext C, and key K that hold with a probability $p \neq 1/2$.
- **Linear Approximation:** We try to find an equation of the form

$$\bigoplus_{i \in I} P_i \oplus \bigoplus_{j \in J} C_j = \bigoplus_{k \in K} K_k \tag{1}$$

  where I, J, K are sets of bit positions (linear masks on input, output, and key).
- **Bias:** If the equation holds with probability p, the *bias* is $\epsilon = p - 1/2$ and the correlation is $c = 2\epsilon = 2p - 1$.
- **Why this works:** If $|\epsilon| > 0$, the cipher can be distinguished from a random permutation (where $\epsilon \approx 0$) and the key bits appearing in the linear relation can be recovered with enough known plaintext/ciphertext pairs.

Exam Tip: Crucial distinction: Differential Cryptanalysis (DC) is a *Chosen-Plaintext Attack* (we inject chosen differences). LC is a *Known-Plaintext Attack* (we passively observe pairs). This makes LC especially relevant in scenarios where active injection is impossible, but large traffic can be monitored.

2 Correlations: Definition and Estimation

2.1 Theoretical Correlation

Let X, Y be binary random variables in $\{0, 1\}$. The (Walsh) correlation of X and Y is defined as

$$C(X, Y) = 2 \cdot \Pr[X = Y] - 1. \tag{2}$$

Equivalently, using expectations of parity:

$$C(X, Y) = \mathbb{E}\big[(-1)^{X \oplus Y}\big]. \tag{3}$$

Properties:

- $C(X, Y) \in [-1, +1]$.
- If $X = Y$ (always), then $\Pr[X = Y] = 1 \Rightarrow C = 1$.
- If $X = \bar{Y}$ (always opposite), then $\Pr[X = Y] = 0 \Rightarrow C = -1$.
- If X and Y are independent and balanced, $\Pr[X = Y] = 1/2 \Rightarrow C = 0$.
- For any binary Z, $C(X, Y) = C(X \oplus Y, 0)$, so it is enough to study correlations with the constant zero.

2.2 Empirical Estimation and Variance

In a real attack, the correlation is unknown and must be estimated from samples. Assume that we observe N i.i.d. pairs (X_t, Y_t), $t = 1, \ldots, N$. Define

$$Z_t = (-1)^{X_t \oplus Y_t} \in \{-1, +1\}. \tag{4}$$

The *empirical correlation estimator* is

$$\hat{C} = \frac{1}{N} \sum_{t=1}^{N} Z_t. \tag{5}$$

If the true correlation is $C = \mathbb{E}[Z_t]$, then:

- $\mathbb{E}[\hat{C}] = C$ (the estimator is unbiased).
- $\mathrm{Var}(Z_t) = 1 - C^2$.
- $\mathrm{Var}(\hat{C}) = \dfrac{1 - C^2}{N} \approx \dfrac{1}{N}$ when $|C|$ is small.

Therefore, the standard deviation of $\hat{C}$ is approximately $1/\sqrt{N}$ for small correlations. To *reliably detect* a correlation of magnitude $|C| = |2\epsilon| = 2|\epsilon|$, we need

$$|C| \gg \frac{1}{\sqrt{N}} \quad \Rightarrow \quad N \gg \frac{1}{C^2} = \frac{1}{4\epsilon^2}. \tag{6}$$

Up to constant factors, this gives the standard LC rule of thumb:

$$N \approx \frac{1}{\epsilon^2} \tag{7}$$

known-plaintext pairs to exploit a bias ϵ.

Exam Tip: For exam purposes, quoting the complexity as $N \propto 1/\epsilon^2$ and explaining the link with the variance of the empirical correlation is usually enough.

3 Linear Masks and Trails

3.1 Linear Masks for a Function

Let $f : \{0,1\}^n \to \{0,1\}^m$ and let $u \in \{0,1\}^n$, $v \in \{0,1\}^m$ be *linear masks*. The inner product is defined as

$$u \cdot x = \bigoplus_{i=1}^{n} u_i x_i \tag{8}$$

and similarly for $v \cdot f(x)$. The correlation of the linear approximation

$$u \cdot x \approx v \cdot f(x) \tag{9}$$

is denoted by $C_f(u, v)$ and defined as

$$C_f(u, v) = 2 \cdot \Pr_x [u \cdot x = v \cdot f(x)] - 1 = \mathbb{E}_x[(-1)^{u \cdot x \oplus v \cdot f(x)}]. \tag{10}$$

For consistency with the course notes, the *Linear Probability (LP)* associated to this approximation is taken as the squared correlation

$$LP_f(u, v) = C_f(u, v)^2, \tag{11}$$

and the corresponding *weight* is

$$w_f(u, v) = -\log_2 LP_f(u, v) = -\log_2 \left(C_f(u, v)^2\right). \tag{12}$$

Note that the actual success probability of the approximation is

$$\Pr_x [u \cdot x = v \cdot f(x)] = \frac{1 + C_f(u, v)}{2}, \tag{13}$$

but in the lecture notes the symbol LP is reserved for C^2 when computing trail weights.

3.2 Linear Trails and Round Decomposition

In an iterated block cipher, a linear approximation over several rounds can often be decomposed into a sequence of masks $(u^{(1)}, v^{(1)}), \ldots, (u^{(r)}, v^{(r)})$ for each round function. Assuming that the round keys are independent and uniformly random, these per-round correlations behave (approximately) like independent random variables in the piling-up lemma sense.

4 The Piling-Up Lemma

4.1 Statement

Let $X_1, \ldots, X_n$ be independent binary random variables and define

$$Z = X_1 \oplus X_2 \oplus \cdots \oplus X_n. \tag{14}$$

If each X_i has correlation

$$c_i = C(X_i, 0) = 2 \cdot \Pr[X_i = 0] - 1, \tag{15}$$

then the correlation of their XOR is

$$C(Z, 0) = \prod_{i=1}^{n} c_i. \tag{16}$$

4.2 Application to Linear Cryptanalysis

In LC, if a linear trail is composed of per-round approximations with correlations $c_1, \ldots, c_r$, and if the round keys make these approximations independent, the overall trail correlation is

$$C_{\text{trail}} = \prod_{i=1}^{r} c_i. \tag{17}$$

Since $|c_i| < 1$ for non-trivial S-boxes, the absolute correlation decreases exponentially with the number of active S-boxes (and rounds).

Exam Tip: Link this explicitly to design: the goal is to force any non-trivial linear trail to involve many S-boxes so that the product of their correlations is extremely small.

5 Linear Approximation Table (LAT)

5.1 Definition

For a given S-box $S : \{0,1\}^n \to \{0,1\}^n$, the *Linear Approximation Table (LAT)* is a table whose entry (α, β) stores

$$C_S(\alpha, \beta) = C\big(\alpha \cdot x, \beta \cdot S(x)\big) = 2 \cdot \Pr_x\left[\alpha \cdot x = \beta \cdot S(x)\right] - 1. \tag{18}$$

Values of $C_S(\alpha, \beta)$ close to zero mean good resistance; high absolute values indicate linear weaknesses.

5.2 Example and Weights

For small S-boxes or toy functions (like the χ-like functions in the notes), one can explicitly compute all entries in the LAT. For such functions it often happens that

$$C_S(\alpha, \beta) \in \{-1/2, 0, +1/2\}. \tag{19}$$

In that case the Linear Probability in the sense of the notes is

$$LP_S(\alpha, \beta) = C_S(\alpha, \beta)^2 \in \left\{0, \frac{1}{4}\right\}, \tag{20}$$

and the corresponding weights

$$w_S(\alpha, \beta) = -\log_2 LP_S(\alpha, \beta) \tag{21}$$

take only a few discrete values (e.g. 2 when $|C_S| = 1/2$).

6 Distinguishers and Matsui's Algorithms

6.1 PRP Distinguisher from a Linear Approximation

Assume we know a linear approximation of the whole cipher

$$P \cdot \alpha \oplus C \cdot \beta = K \cdot \gamma \tag{22}$$

with bias ϵ and correlation $c = 2\epsilon$. For a *fixed but unknown* key K, the right-hand side $K \cdot \gamma$ is a constant bit. Define

$$X = P \cdot \alpha \oplus C \cdot \beta. \tag{23}$$

Under the real cipher, X has bias ϵ, while under a random permutation it is (essentially) unbiased.

A PRP distinguisher works as follows:

1. Collect N plaintext/ciphertext pairs (P_t, C_t).
2. For each pair, compute
$$X_t = P_t \cdot \alpha \oplus C_t \cdot \beta.$$
3. Estimate the empirical correlation $\hat{C}$ of the sequence X_t with the constant zero (or with a guessed constant).
4. If $|\hat{C}|$ is significantly larger than what is expected from random noise ($\approx 1/\sqrt{N}$), output "cipher"; otherwise output "random permutation".

The data complexity is driven by the same $N \approx 1/\epsilon^2$ rule as before.

6.2 Matsui's Algorithm 1: Recovering a Key Bit

Matsui's Algorithm 1 recovers one bit (or one linear combination) of the key.

1. Find a linear approximation of the *full* cipher
$$P \cdot \alpha \oplus C \cdot \beta = K \cdot \gamma$$
with bias ϵ.
2. Collect N known plaintext/ciphertext pairs (P_t, C_t).
3. For each pair, compute
$$b_t = P_t \cdot \alpha \oplus C_t \cdot \beta.$$
4. Let T_0 be the number of indices t such that $b_t = 0$.
5. If $T_0 > N/2$, guess $K \cdot \gamma = 0$. If $T_0 < N/2$, guess $K \cdot \gamma = 1$.
6. With $N \approx 1/\epsilon^2$, the success probability for this bit is close to 1.

This is essentially a *sign test* on the estimated bias.

6.3 Matsui's Algorithm 2: Last-Round Key Recovery

Algorithm 2 uses a linear approximation that covers all but the last round, and then guesses the last-round key bits.

1. Choose a linear trail for the first $R - 1$ rounds with good correlation.
2. For each hypothesis k' on (part of) the last-round subkey:
   (a) Partially decrypt (or encrypt) the last round under k' to obtain an intermediate value $V_t(k')$ for each pair (P_t, C_t).

(b) Evaluate the linear approximation on P_t and $V_t(k')$, and compute the empirical correlation $\hat{C}(k')$.

3. The correct key hypothesis k^* is expected to yield the highest absolute correlation $|\hat{C}(k^*)|$; wrong keys behave like noise with $|\hat{C}(k')| \approx 0$.

Exam Tip: Conceptually separate: (1) the PRP distinguisher based on correlation, and (2) the key-recovery phase that searches over subkeys and uses the distinguisher as a scoring function.

7 Wide Trail Strategy and Defense

7.1 Minimizing Correlation of Linear Trails

The Wide Trail Strategy is a design methodology (used e.g. in Rijndael/AES) to resist both DC and LC. Its core principles are:

- Minimize the maximum correlation (or maximum linear probability) of any individual S-box.
- Maximize the minimum number of *active S-boxes* in any differential or linear trail across several rounds.

If the maximum S-box correlation is $|c_{\text{sbox}}|$, and any 4-round trail must involve at least N_{act} active S-boxes, then the correlation of any 4-round trail is bounded by

$$|C_{\text{trail}}| \leq |c_{\text{sbox}}|^{N_{\text{act}}}. \tag{24}$$

7.2 AES Example and Security Strength

In AES, the diffusion layer (ShiftRows + MixColumns) is designed with a high *branch number* (MDS property) so that any non-zero input mask to MixColumns spreads to multiple output bytes. For 4 rounds, any linear trail must involve at least 25 active S-boxes.

The AES S-box has maximum correlation bounded by about 2^{-3}, so for a 4-round trail:

$$|C_{\text{trail}}| \leq (2^{-3})^{25} = 2^{-75}. \tag{25}$$

For the full 10-round AES-128, any non-trivial linear trail has even smaller correlation, so the data complexity $N \approx 1/\epsilon^2$ becomes far beyond 2^{128}. In the language of primitive security, this implies that practical LC attacks cannot beat the brute-force security level of AES as a PRP/PRF.

Exam Tip: When discussing security strength, explicitly compare the required number of chosen/known texts to the block size and to the best generic attack (brute force). If LC needs more data than the birthday bound or than 2^k work, it is not a practical threat.

8 Self-Assessment Questions

Level 1: Concepts

Q: What is the fundamental difference between Differential and Linear Cryptanalysis?
Answer: Differential Cryptanalysis looks for high-probability propagation of *differences* (ΔX) and is typically a Chosen-Plaintext Attack. Linear Cryptanalysis looks for high *correlations* (linear biases) between plaintext, ciphertext, and key bits, and is a Known-Plaintext Attack.

Q: What is the Piling-Up Lemma used for?

Answer: It is used to compute the total correlation of a linear trail as the product of the correlations of independent components. For LC, this means multiplying the per-round or per-S-box correlations along the trail: $C_{\text{total}} = \prod_i c_i$.

Level 2: Design & Strategy

Q: How does the Wide Trail Strategy defend against Linear Cryptanalysis?
Answer: It forces any valid linear trail to activate many S-boxes, each having a small maximum correlation. The product of many small correlations is extremely close to zero, making the bias undetectable with any feasible number of known plaintexts.

Q: What is the data complexity of a linear attack with bias ϵ?
Answer: To detect or exploit a bias of magnitude $|\epsilon|$, the attacker needs on the order of $N \approx 1/\epsilon^2$ known plaintext/ciphertext pairs. If $|\epsilon|$ is very small (e.g. $|\epsilon| = 2^{-64}$), the required data ($\approx 2^{128}$ pairs) is completely impractical.

Level 3: Analysis

Q: Why does the key K disappear in Differential Cryptanalysis but appear in Linear Cryptanalysis?
Answer: In DC, we XOR two executions: $(x \oplus k) \oplus (x' \oplus k) = x \oplus x'$, so the key cancels out in the difference domain. In LC, we approximate operations with linear equations, but the XOR with the key $x \oplus k$ is itself linear, so it survives as an additive constant term $K \cdot \gamma$, which is precisely the unknown the attack tries to recover.

Q: How is a PRP distinguisher used inside Matsui's Algorithm 2?
Answer: For each last-round subkey guess, the attacker partially decrypts and evaluates the linear approximation, estimating the empirical correlation. The distinguisher then treats the key guess that yields the largest absolute correlation as the most likely, since wrong keys should behave like a random permutation with correlation near zero.