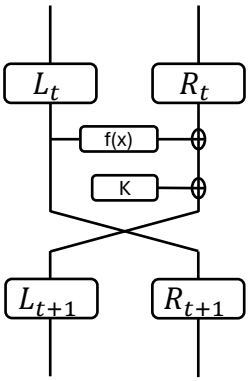
SFLL-HDで保護されたSIMECK32/64に対する差分故障解析

李　陽　研究室　Li Enze
電気通信大学　情報学専攻　セキュリティ情報学プログラム

背景 | Introduction

今は軽量暗号アルゴリズムが様々なデバイスで幅広く使われているが、差分故障解析（Differential Fault Analysis, DFA）のようなハードウェア層の脅威も存在している。

SIMECK暗号は、軽量ブロック暗号の一種であり、処理性能や実装面積の制約があるため、十分な安全性を確保することが難しい場合が存在している。



SIMECK は Feistel 構造を採用しており、ラウンド関数は以下のように定義される。

$$R_{k_i}(l_i, r_i) = (r_i \oplus f(l_i) \oplus k_i, l_i)$$

ここで、l_i とr_i はラウンド iにおける左右のワード k_iはラウンド鍵、fは非線形関数である。

関数fの定義は

f(x) = (x∧(x ≪5)) ⊕ (x≪1)

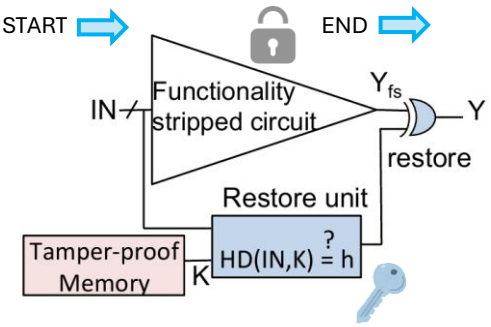
<<は左ローテーションを表す。

実験方法 | Methodology

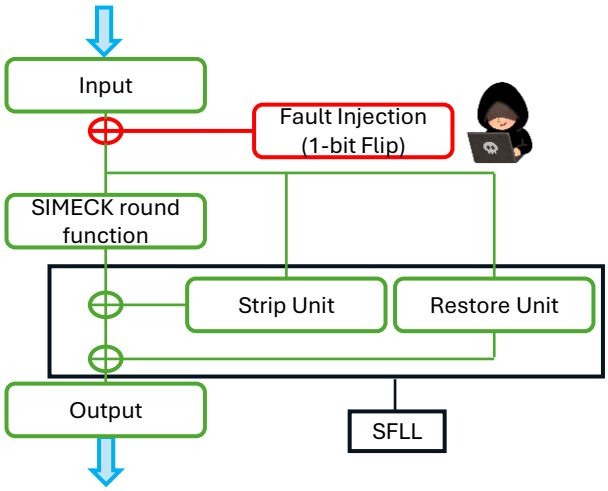
本実験は、SIMECK暗号にStripped-Functionality Logic Locking (SFLL)を適用し、誤りロック鍵が DFA に与える影響を評価する C 言語ベースの実験を行う。

実装暗号：32ラウンドSimeck32/64
ロック鍵の長さ：16

正しいロック鍵はランダムな16ビット値を生成し、暗号化処理時に適用する。



SFLLの構造



1ラウンドの流れ（故障ありの場合）

各ラウンドの左側16ビットの入力を用いて、StripとRestoreがそれぞれハミング距離を判定する。判定条件を満たすたびに、ラウンド出力のビット0を反転する。故障注入時には、対象ラウンドの入力の1ビットを反転してから、ラウンド関数の計算と両ユニットの判定を行う。

考査 | Discussion

この実験モデルでは、StripとRestoreが故障注入後の同じ入力を参照する。正しいロック鍵を用いると両者の判定が一致するため、出力と入力が全く同じである。

単一ビット故障を注入した例では、故障なしの場合と比べて、差分が現れるビットの位置と数がラウンドごとに変化した。SIMECK32/64自体でも故障差分は伝搬するため、今後はロックなしのモデルと同じ故障条件で比較し、SFLL-HDが差分伝搬とDFAのラウンド鍵復元率に与える影響を評価する。

