Fault based Collision Attacks on AES

Volker Krummel

Faculty of Computer Science, Electrical Engineering and Mathematics
University of Paderborn

joint work with

Johannes Blömer

Fault Diagnosis and Tolerance in Cryptography 2006

Overview

Outline

- Collision Attacks
- Security model/Scenarios
- Structure of AES
- Attacks

Collision Attacks

Collision (accidental)

- basic idea due to Dobbertin
- attacker detects (nearly) identical intermediate results during encryptions of different plaintexts
- use side-channel information to detect collisions
- Schramm et al. mounted collision attacks on DES and AES

Fault based Collisions

Collision (¬ accidental)

- combine concepts of collision and fault attacks
- induce faults to create collisions
- does not need faulty ciphertexts, only *collision information*
- breaks implementations protected by *MEM* (Memory Encryption Module)
- needs only a moderate number of faults

Security Model

- Extension $FAES_K$ of bijective function AES_K
    - $FAES_K(p, b)$
    - key K, plaintext p, Bit b
    - $FAES_K$ is not bijective $\Rightarrow$ collisions
- $\mathcal{A}$ can choose plaintexts
- $\mathcal{A}$ can induce faults into encryption process (bit flip)
- $\mathcal{A}$ gets "collision information"

Security Model

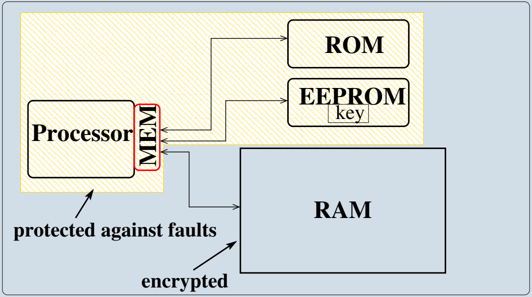
- Extension $FAES_K$ of bijective function AES_K
  - $FAES_K(p, b)$
  - key K, plaintext p, Bit b
  - $FAES_K$ is not bijective $\Rightarrow$ collisions
- $\mathcal{A}$ can choose plaintexts
- $\mathcal{A}$ can induce faults into encryption process (bit flip)
- $\mathcal{A}$ gets "collision information"

Collision Information

- collision information lets $\mathcal{A}$ detect collisions
- modeled as evaluation of an injective function f_K
  - depends on concrete implementation
  - inputs: p plaintext and bit position b
  - output: information about intermediate encryption state
- realizations:
  - may be a faulty ciphertext
  - CBC-MAC or hash value
  - side channel information, e.g. power trace

Scenarios

Schematic view



- kind of fault induction (flip, set or reset)
- precision of fault induction
- protection of smartcard
- collision information valid during the whole attack?

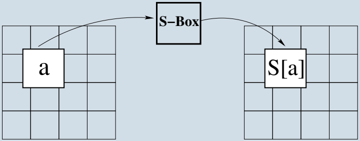
General Structure

- iterated block cipher with 10,12 or 14 rounds
- operates on 4×4 byte matrix (state)
- round function consists of the operations
- Notation: $p_i^{(r),(o)}$, ith byte of the state after operation o of round r

AES-Operations

1. SubBytes [B]
2. ShiftRows [R]
3. MixColumns [C]
4. AddRoundKey [K]

sketch



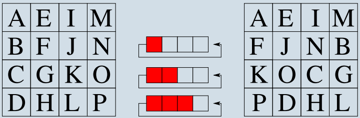
General Structure

- iterated block cipher with 10,12 or 14 rounds
- operates on 4×4 byte matrix (state)
- round function consists of the operations
- Notation: $p_i^{(r),(o)}$, ith byte of the state after operation o of round r

AES-Operations

1. SubBytes [B]
2. ShiftRows [R]
3. MixColumns [C]
4. AddRoundKey [K]

sketch



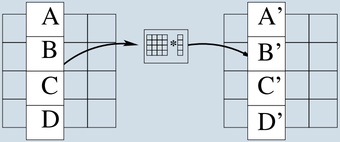
General Structure

- iterated block cipher with 10,12 or 14 rounds
- operates on 4×4 byte matrix (state)
- round function consists of the operations
- Notation: $p_i^{(r),(o)}$, ith byte of the state after operation o of round r

AES-Operations

1. SubBytes [B]
2. ShiftRows [R]
3. MixColumns [C]
4. AddRoundKey [K]

sketch



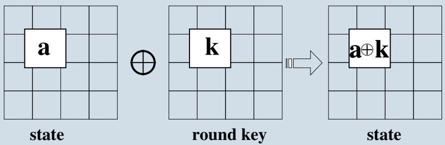
General Structure

- iterated block cipher with 10,12 or 14 rounds
- operates on 4×4 byte matrix (state)
- round function consists of the operations
- Notation: $p_i^{(r),(o)}$, ith byte of the state after operation o of round r

AES-Operations

1. SubBytes [B]
2. ShiftRows [R]
3. MixColumns [C]
4. AddRoundKey [K]

sketch



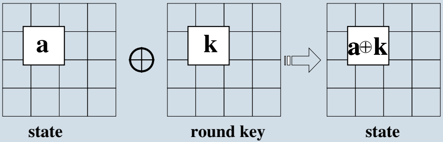
General Structure

- iterated block cipher with 10,12 or 14 rounds
- operates on 4×4 byte matrix (state)
- round function consists of the operations
- Notation: $p_i^{(r),(o)}$, ith byte of the state after operation o of round r

AES-Operations

1. SubBytes [B]
2. ShiftRows [R]
3. MixColumns [C]
4. AddRoundKey [K]

sketch



First Attack

Setting

- $\mathcal{A}$ can flip specific bit e of $p^{(1),(B)}$
- collision information remains valid
- smartcard is not protected

Precomputation

- $\mathcal{A}$ computes information about differences:
  - tables T_e, $0 \leq e \leq 7$ such that $T_e[y] := \{\{s,t\} \mid s+t=y, \mathbf{S}[s]+\mathbf{S}[t]=2^e\}$
  - 3 cases: $T_e[y]$ empty, $T_e[y]$ contains 2 elements or $T_e[y]$ contains 4 elements
- $\mathcal{A}$ collects collision information $f_K(p_0^{(1),(B)}, -)$ for all values of $p_0 \in \{0,\ldots,255\}$ and arbitrary but fixed $p_1,\ldots,p_{15}$

First Attack

Setting

- $\mathcal{A}$ can flip specific bit e of $p^{(1),(B)}$
- collision information remains valid
- smartcard is not protected

Precomputation

- $\mathcal{A}$ computes information about differences:
  - tables T_e, $0 \leq e \leq 7$ such that $T_e[y] := \{\{s,t\} \mid s+t=y, \mathbf{S}[s]+\mathbf{S}[t]=2^e\}$
  - 3 cases: $T_e[y]$ empty, $T_e[y]$ contains 2 elements or $T_e[y]$ contains 4 elements
- $\mathcal{A}$ collects collision information $f_K(p_0^{(1),(B)}, -)$ for all values of $p_0 \in \{0,\ldots,255\}$ and arbitrary but fixed $p_1,\ldots,p_{15}$

First Attack (2)

Attack

1. $\mathcal{A}$ chooses arbitrary value $q_0 \in \{0, \ldots, 255\}$
2. flip of bit e of $q_0^{(1),(B)}$ during the encryption of $(q_0, p_1, \ldots, p_{15})$
3. search p_0 such that $f_K(p_0^{(1),(B)}, -) = f_K(q_0^{(1),(B)}, e)$
4. $\mathcal{A}$ knows: $\{p_0 + k_0, q_0 + k_0\} \in T_e[p_0 + q_0]$
5. Hence, $k_0 \in \{p_0 + s \mid s \in T_e[p_0 + q_0]\}$

First Attack (3)

- $\mathcal{A}$ restricted k_0 to only 2 possible values
- repetition of the attack leads to a unique value for k_0
- expected number of induced faults:
  2 per key byte, 32 for the whole AES key

Second Attack

Setting

- $\mathcal{A}$ can flip specific bit e of $p^{(0),(K)}$
- collision information remains valid
- smartcard is protected by MEM (Memory Encryption Module)

Details

- MEM: $h : \{0,1\}^8 \to \{0,1\}^8$, $p_0 + k_0 \mapsto h(p_0 + k_0)$
- Fault: $h^{-1}(h(p_0 + k_0) + 2^e)$

$\Rightarrow$ impact on encryption unknown

Second Attack

Setting

- $\mathcal{A}$ can flip specific bit e of $p^{(0),(K)}$
- collision information remains valid
- smartcard is protected by MEM (Memory Encryption Module)

Details

- MEM: $h : \{0,1\}^8 \longrightarrow \{0,1\}^8$, $p_0 + k_0 \mapsto h(p_0 + k_0)$
- Fault: $h^{-1}(h(p_0 + k_0) + 2^e)$

$\Rightarrow$ impact on encryption unknown

Second Attack - 1. Part

Precomputation

- $\mathcal{A}$ collects collision information $f_K(h(p_0^{(0),(K)}), -)$ for all values of $p_0 \in \{0, \ldots, 255\}$ and arbitrary but fixed $p_1, \ldots, p_{15}$

First part

- $\mathcal{A}$ chooses an arbitrary $q_0 \in \{0, \ldots, 255\}$
- $\mathcal{A}$ encrypts $(q_0, p_1, \ldots, p_{15})$ flipping bit e of $h(q_0^{(0),(K)})$
- $\mathcal{A}$ searches p_0 s.t. $f_K(h(p_0^{(0),(K)}), -) = f_K(h(q_0^{(0),(K)}), e)$

$\Rightarrow$ $\mathcal{A}$ knows that $h(p_0 + k_0) + h(q_0 + k_0) = 2^e$

- repeating this $\mathcal{A}$ can compute function g_0 s.t. $g_0(x) = h(x + k_0) + c_0$
- but: c_0 unknown $\Rightarrow$ no information about k_0 :-(
- $\mathcal{A}$ computes $g_1 \ldots g_{15}$ as above s.t. $g_i(x) = h(x + k_i) + c_i$

Second Attack - 1. Part

Precomputation

- $\mathcal{A}$ collects collision information $f_K(h(p_0^{(0),(K)}), -)$ for all values of $p_0 \in \{0, \ldots, 255\}$ and arbitrary but fixed $p_1, \ldots, p_{15}$

First part

- $\mathcal{A}$ chooses an arbitrary $q_0 \in \{0, \ldots, 255\}$
- $\mathcal{A}$ encrypts $(q_0, p_1, \ldots, p_{15})$ flipping bit e of $h(q_0^{(0),(K)})$
- $\mathcal{A}$ searches p_0 s.t. $f_K(h(p_0^{(0),(K)}), -) = f_K(h(q_0^{(0),(K)}), e)$

$\Rightarrow$ $\mathcal{A}$ knows that $h(p_0 + k_0) + h(q_0 + k_0) = 2^e$

- repeating this $\mathcal{A}$ can compute function g_0 s.t. $g_0(x) = h(x + k_0) + c_0$
- but: c_0 unknown $\Rightarrow$ no information about k_0 :-(
- $\mathcal{A}$ computes $g_1 \ldots g_{15}$ as above s.t. $g_i(x) = h(x + k_i) + c_i$

Second Attack - 2. Part

Second part

- $\mathcal{A}$ guesses $\widehat{k}_0, \widehat{k}_i$
- $\mathcal{A}$ chooses $x \in \{0, \dots, 255\}$
- computes
$$\begin{array}{rl} & g_0(x + \widehat{k}_0) \\ + & g_i(x + \widehat{k}_i) \\ \hline & h(x + \widehat{k}_0 + k_0) + h(x + \widehat{k}_i + k_i) + c_0 + c_i \end{array}$$
- test hypothesis $\widehat{k}_0, \widehat{k}_i$ by checking if $g_0(x + \widehat{k}_0) + g_i(x + \widehat{k}_i)$ remains constant for several x

- expect that after inducing 285 faults only 256 candidates of the full AES key remains

Second Attack - 2. Part

Second part

- $\mathcal{A}$ guesses $\widehat{k}_0, \widehat{k}_i$
- $\mathcal{A}$ chooses $x \in \{0, \ldots, 255\}$
- computes
$$\begin{array}{rl} & g_0(x+\widehat{k}_0) \\ + & g_i(x+\widehat{k}_i) \\ \hline & h(x+\widehat{k}_0+k_0)+h(x+\widehat{k}_i+k_i)+c_0+c_i \end{array}$$
- test hypothesis $\widehat{k}_0, \widehat{k}_i$ by checking if $g_0(x+\widehat{k}_0)+g_i(x+\widehat{k}_i)$ remains constant for several x

- expect that after inducing 285 faults only 256 candidates of the full AES key remains

Conclusions

- combine concepts of collision and fault attacks
- induce faults to create collisions
- does not need faulty ciphertexts, only collision information
- breaks implementations protected by MEM
- needs only a moderate number of faults

Thank you for your attention!