Collision Attacks on AES-based MAC: Alpha-MAC

Alex Biryukov[1], **Andrey Bogdanov**[2], Dmitry Khovratovich[1] and Timo Kasper[2]

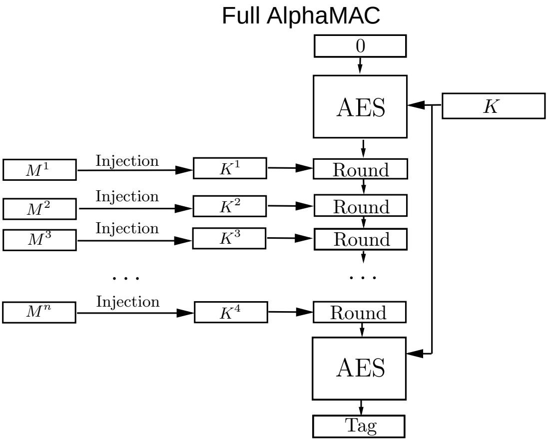
[1]University of Luxemburg, Luxemburg
`{alex.biryukov,dmitry.khovratovich}@uni.lu`

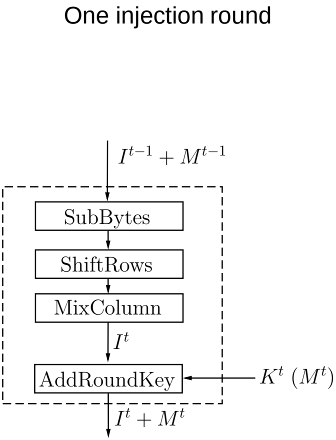
[2]Chair for Communication Security, Horst-Görtz Institute,
Ruhr-University Bochum, Germany
`{abogdanov,tkasper}@crypto.rub.de`

CHES'07, Vienna, Austria, 2007

Outline

AlphaMAC Specification and Notation





Attack Motivation

Practical Motivation

- $\approx$ 2.5 times faster than e.g. CBC-MAC ($4 + 20/t$ instead of 10 rounds)
- $\Rightarrow$ AlphaMAC can be efficiently applied in embedded systems
- $\Rightarrow$ Interesting target for side-channel analysis

Theoretical Motivation

- Improve traditional side-channel collision attacks
- Exploit the existence of collisions in AlphaMAC for selective forgery
- Show that the internal state has to be protected against SCA as well

General Assumptions and Attack Outline

Assumptions

- Keyed AES rounds are perfectly protected against side-channel attacks
- Unkeyed message injection AES rounds are not protected

Two Basic Attack Steps

- Obtain the 16-byte state I^1 by side-channel collision attacks
- Mount a selective forgery attack using collisions in AlphaMAC

Basic Collision Attack on AES: Outline

Attack Outline (Schramm et al)

- Generate random plaintexts of a special form
- Perform N measurements and detect simple collisions
- 16 simple collisions needed
  (construct 16 nonlinear equations)
- Solve the equations using pre-computed tables and test key candidates using a plaintext-ciphertext pair

Basic Attack on AES: Notation

$B = \textsc{MixColumn}(A),\ A = \textsc{ShiftRows}(\textsc{SubBytes}(P \oplus K))$

$$\begin{pmatrix} b_{0j} \\ b_{1j} \\ b_{2j} \\ b_{3j} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \times \begin{pmatrix} a_{0j} \\ a_{1j} \\ a_{2j} \\ a_{3j} \end{pmatrix}$$

b_{00}

If $P = (p_{ij})$ is plaintext and $K = (k_{ij})$ is subkey, then

$$\begin{aligned} b_{00} &= 02 \cdot a_{00} \oplus 03 \cdot a_{10} \oplus 01 \cdot a_{20} \oplus 01 \cdot a_{30} = \\ &= 02 \cdot S(p_{00} \oplus k_{00}) \oplus 03 \cdot S(p_{11} \oplus k_{11}) \\ &\quad \oplus 01 \cdot S(p_{22} \oplus k_{22}) \oplus 01 \cdot S(p_{33} \oplus k_{33}). \end{aligned}$$

Basic Attack on AES: Simple Collisions

$b_{00} = b'_{00}$

Second round:
$S(b_{00} \oplus k_{00}) = S(b'_{00} \oplus k_{00})$ detected $\Rightarrow$
$b_{00} \oplus k_{00} = b'_{00} \oplus k_{00} \Rightarrow$
$b_{00} = b'_{00}$

Collision equation

For two plaintexts P and P' with $p_{00} = p_{11} = p_{22} = p_{33} = \delta$ and $p'_{00} = p'_{11} = p'_{22} = p'_{33} = \epsilon$, $\delta \neq \epsilon$, one obtains the following, provided $b_{00} = b'_{00}$:

$$02 \cdot S(k_{00} \oplus \delta) \oplus 03 \cdot S(k_{11} \oplus \delta) \oplus 01 \cdot S(k_{22} \oplus \delta) \oplus 01 \cdot S(k_{33} \oplus \delta)$$
$$= 02 \cdot S(k_{00} \oplus \epsilon) \oplus 03 \cdot S(k_{11} \oplus \epsilon) \oplus 01 \cdot S(k_{22} \oplus \epsilon) \oplus 01 \cdot S(k_{33} \oplus \epsilon)$$

Basic Attack on AES: Attack Complexity

Collision probability

The probability that after N executions at least one collision $b_{00} = b'_{00}$ occurs in a single byte is:

$$p_N = 1 - \prod_{l=0}^{N-1}(1 - l/2^8)$$

Complexity

- The attacker needs at least 16 collisions, 4 for each column of B, so $p_N^{16} \geq 1/2$ and $N \approx 40$
- About 540 MByte pre-computed tables
- Chosen-plaintext possibility needed

Our Modifications to the Standard Collision Attacks

What We Do

- Several byte collisions:
  - Consider them as nonlinear equations over $GF(2^8)$
  - Solve these systems by brute-force
  - $\Rightarrow$ No precomputations and only negligible memory
- Look for collisions in 3 injection rounds:
  - Instead of working with only a single round
  - Possible due to the fact that no entropy is introduced in the injection rounds
  - A lower number of measurements needed

Collisions in the Second Round

2nd Injection Round

- After MixColumns & message addition in the 2nd injection round:

  $$02 \cdot S(i^1_{00} + m^1_{00}) + S(i^1_{22} + m^1_{22}) + m^2_{00} = 02 \cdot S(i^1_{00} + z^1_{00}) + S(i^1_{22} + z^1_{22}) + z^2_{00},$$

  M^1, Z^1, M^2, Z^2 = some message blocks
- After a further collision in another byte of the 0th column:
  - Two nonlinear equations over $GF(2^8)$ with variables $i^1_{00}, i^1_{22} \in GF(2^8)$.
  - Solve them by brute-force $\Rightarrow i^1_{00}$ and i^1_{22}
- Do the same for i^1_{02} and i^1_{20} $\Rightarrow i^1_{02}$ and i^1_{20}

Collisions in the Third Round

3rd Injection Round

- A collision detected in $i^3_{00} + k^3_{00}$, the following relation holds:

$$\begin{aligned}02 \cdot S(03 \cdot S(i^1_{11}) + S(i^1_{33}) + c_1 + m^2_{00})\\ +S(S(i^1_{13}) + 03 \cdot S(i^1_{31}) + c_2 + m^2_{22}) + m^3_{00} =\\ 02 \cdot S(03 \cdot S(i^1_{11}) + S(i^1_{33}) + c'_1 + z^2_{00})\\ +S(S(i^1_{13}) + 03 \cdot S(i^1_{31}) + c'_2 + z^2_{22}) + z^3_{00}\end{aligned}$$

  Z^2, M^2, Z^3, M^3 = some injected message blocks
  c_1, c_2, c'_1, c'_2 = some known constants
- 2 collisions in two bytes of the 0th column
  $\Rightarrow 03 \cdot S(i^1_{11}) + S(i^1_{33})$ and $S(i^1_{13}) + 03 \cdot S(i^1_{31})$
- 2 further collisions in the 2rd column
  $\Rightarrow 03 \cdot S(i^1_{13}) + S(i^1_{31})$ and $S(i^1_{11}) + 03 \cdot S(i^1_{33})$
- These 4 relations $\Rightarrow i^1_{11}$, i^1_{33}, i^1_{13} and i^1_{31}

Collisions in the Fourth Round

4th Injection Round

- By now 8 bytes of I^1 are known
- The collisions equations are more complex ...

$$\begin{gathered}02 \cdot S(03 \cdot S(f_2) + S(g_4) + c_{00} + m^3_{00}) + \\ 03 \cdot S(S(f_1) + 03 \cdot S(g_3) + c_{10}) + \\ S(S(g_2) + 03 \cdot S(f_4) + c_{20} + m^3_{22}) + \\ S(S(g_1) + 03 \cdot S(f_3) + c_{30}) + m^4_{00} \\ = \\ 02 \cdot S(03 \cdot S(f_2) + S(g_4) + c'_{00} + z^3_{00}) + \\ 03 \cdot S(S(f_1) + 03 \cdot S(g_3) + c'_{10}) + \\ S(S(g_2) + 03 \cdot S(f_4) + c'_{20} + z^3_{22}) + \\ S(S(g_1) + 03 \cdot S(f_3) + c'_{30}) + z^4_{00}\end{gathered}$$

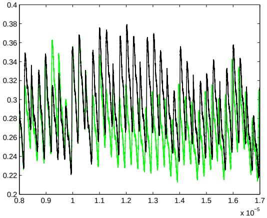
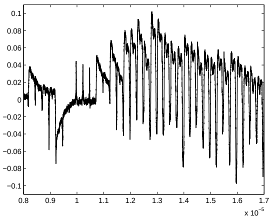
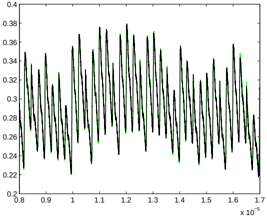
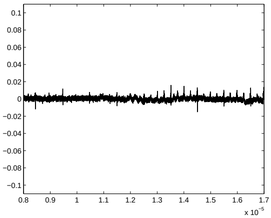
- ... nonetheless allowing to recover the remaining eight I^1 bytes!

Side-Channel Collision Attack Properties

Success Probability, Complexity, Assumptions

- Our side-channel attack works ...
  - ... with a probability of 0.56
  - ... for 29 known random messages (40 chosen plaintexts for AES)
  - ... if one-byte collisions are detectable

Measurements (S-box)

	Power curves		Differences
Unequal arguments		⇒	
Equal arguments		⇒	

Selective Forgery Attack

Facts

Lemma 1 [DR05, FSE'05]

Given I^1, the state value before iteration 1, the map

$$s : (M^1, M^2, M^3, M^4) \to I^5$$

from the sequence of 4 message blocks (M^1, M^2, M^3, M^4) to the state value before iteration 5 is a bijection.

Lemma 2

There exists an algorithm of complexity 2^{11} computing (M^1, M^2, M^3, M^4) from I^5 for a given initial internal state I^1 (inverting s is simple!)

Selective Forgery Attack

Outline

Attack Steps

- *Preliminaries:*
  - I^1 is known,
  - (M, σ) is a victim message-tag pair (4-byte M),
  - M' is a message to authenticate (4-byte M')
- *Step 1:* Compute the intermediate states I for M and I' for M'
- *Step 2:* Compute the 16-byte suffix $\delta = s^{-1}(I)$ for I'
- *Result:* $(M'||\delta, \sigma)$ is the forged message-tag pair

Conclusions

Conclusions

- New type of side-channel collision attacks (recovery of the AlphaMAC internal state):
  - 29 measurements needed only (instead of 40)
  - Known-message scenario (instead of selected plaintext)
- Internal hash of AlphaMAC is not collision resistant
  - New 4-to-1 collisions
  - Selective forgery attack

Outlook

- Apply the improved collision attacks (Andrey Bogdanov, SAC'07)