Generic Attacks against Beyond-Birthday-Bound MACs

Gaëtan Leurent[1], Mridul Nandi[2], Ferdinand Sibleyras[1]

[1] Inria équipe SECRET, Paris, France

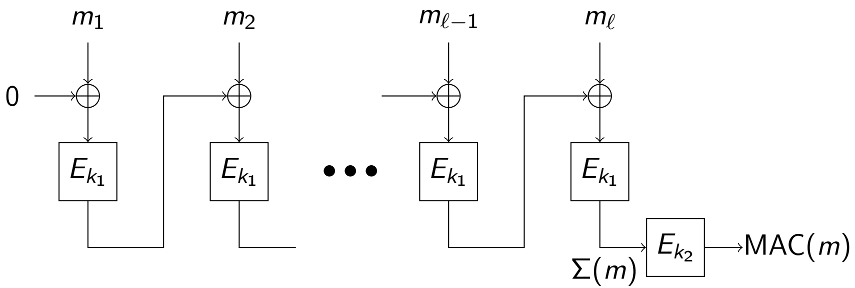
[2] Indian Statistical Institute, Kolkata, India

GT SECRET

Introduction

- **Symmetric cryptography:** Alice and Bob share the same key.
- **Active attacker:** Eve might intercept and manipulate Alice's messages...
- **Authentication:** Alice computes and appends a keyed MAC or tag T.

ECBC-MAC



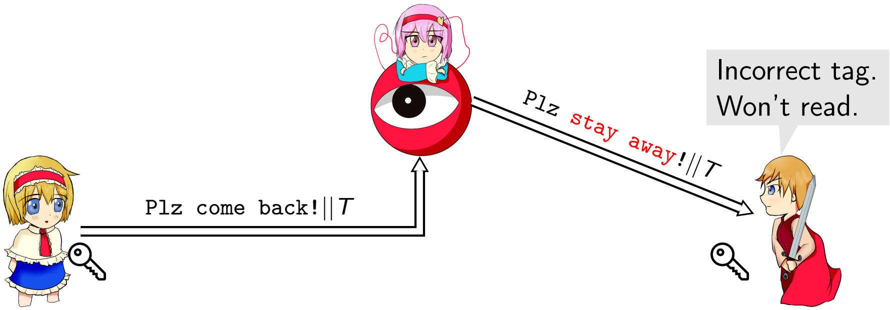
The plaintext m is padded and split into n-bit blocks.

$$\text{MAC}(m) = E_{k_2}(\Sigma(m))$$

Alice sends MAC(m) along with m to guarantee authenticity.

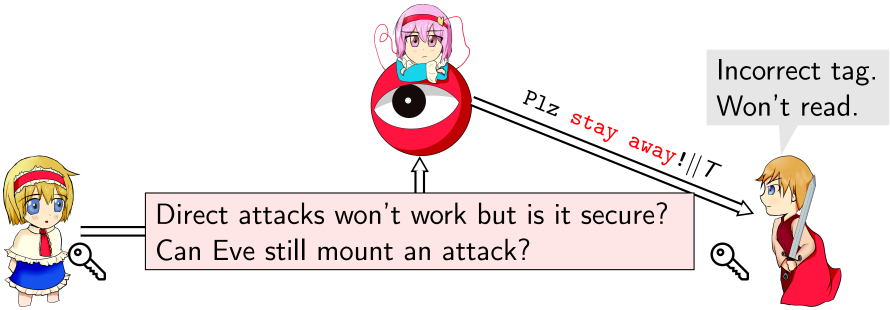
Introduction

- **Verifying:** Bob verifies the tag with the shared key and only reads the message if it is correct.
- **Forgery:** Eve cannot modify the message without forging a new and correct tag.



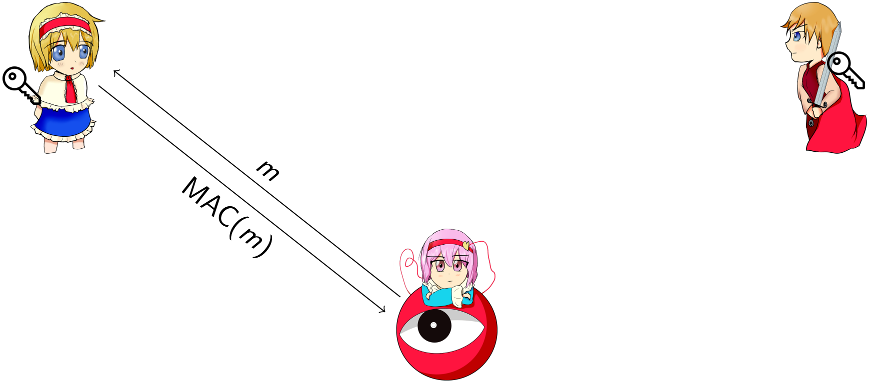
Introduction

- **Verifying:** Bob verifies the tag with the shared key and only reads the message if it is correct.
- **Forgery:** Eve cannot modify the message without forging a new and correct tag.

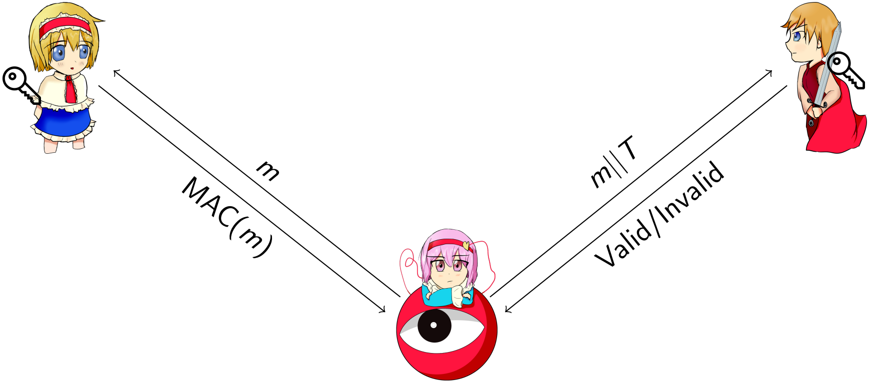


A security game

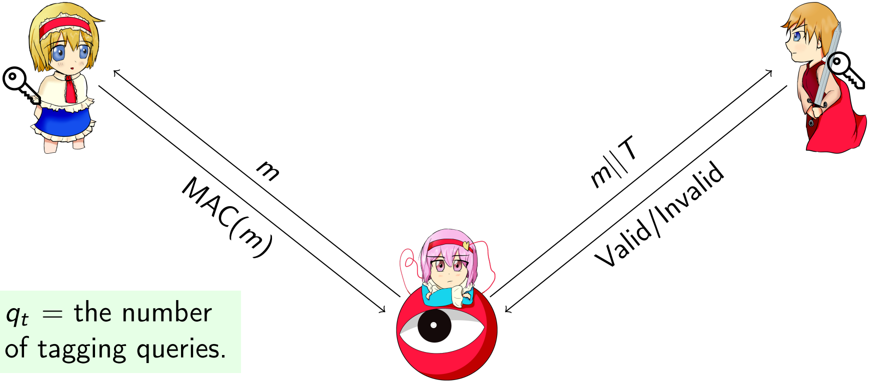
A security game



A security game

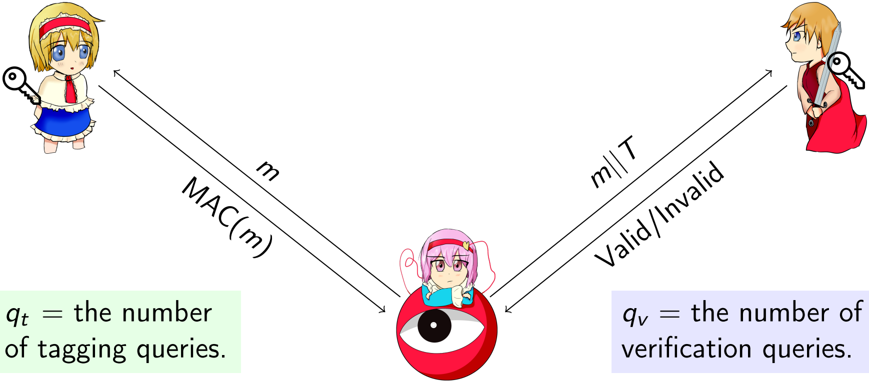


A security game



q_t = the number of tagging queries.

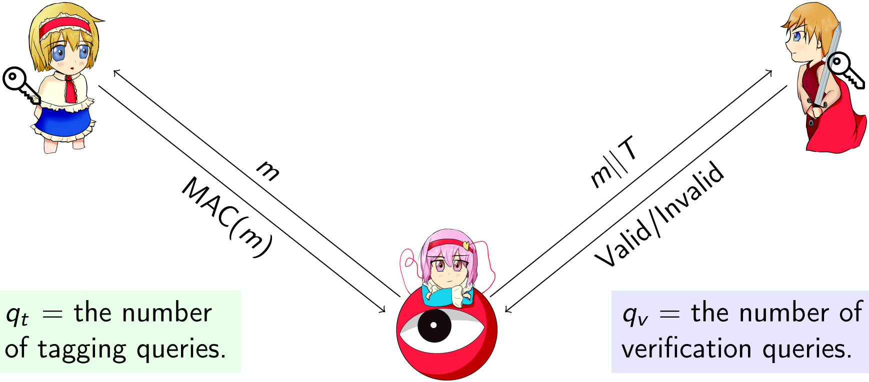
A security game



q_t = the number of tagging queries.

q_v = the number of verification queries.

A security game



q_t = the number of tagging queries.

q_v = the number of verification queries.

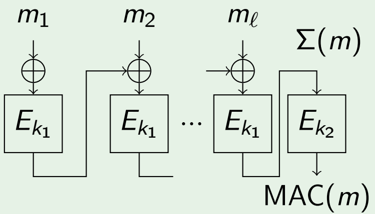
Can Eve forge a valid tag for a message that Alice never saw?

Case of ECBC

Properties of ECBC for all messages m, m', c:

$$
\begin{aligned}
\mathrm{MAC}(m) &= \mathrm{MAC}(m') \\
\implies \quad E_{k_2}(\Sigma(m)) &= E_{k_2}(\Sigma(m')) \\
\implies \quad \Sigma(m) &= \Sigma(m') \\
\implies \quad \Sigma(m||c) &= \Sigma(m'||c) \\
\implies \quad \mathrm{MAC}(m||c) &= \mathrm{MAC}(m'||c)
\end{aligned}
$$

ECBC mode

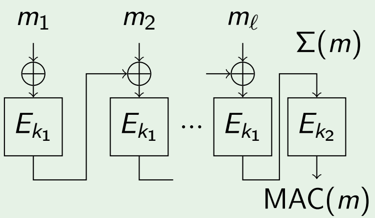


Case of ECBC

Properties of ECBC for all messages m, m', c:

$$
\begin{aligned}
& \mathrm{MAC}(m) = \mathrm{MAC}(m') \\
\Longrightarrow\quad & E_{k_2}(\Sigma(m)) = E_{k_2}(\Sigma(m')) \\
\Longrightarrow\quad & \Sigma(m) = \Sigma(m') \\
\Longrightarrow\quad & \Sigma(m||c) = \Sigma(m'||c) \\
\Longrightarrow\quad & \mathrm{MAC}(m||c) = \mathrm{MAC}(m'||c)
\end{aligned}
$$

ECBC mode

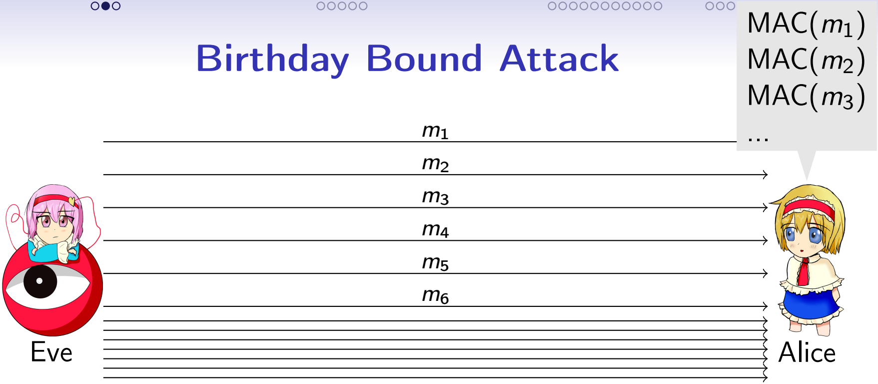


Simple collision approach

Look for a pair of messages X,Y that satisfies:

$$\Sigma(X) = \Sigma(Y) \iff \mathrm{MAC}(X) \oplus \mathrm{MAC}(Y) = 0$$

Birthday Bound Attack



Looking for collisions

Eve looks for $\mathrm{MAC}(m_i) = \mathrm{MAC}(m_j)$ for some $i \neq j$.
She has $\simeq q_t^2$ pairs for an n-bit relationship so chances grow as:

$$\mathbf{Adv}(\mathcal{A}) \simeq \frac{q_t^2}{2^n}$$

Forgery from collisions

Expansion property

$$\mathsf{MAC}(m) = \mathsf{MAC}(m') \implies \mathsf{MAC}(m\|c) = \mathsf{MAC}(m'\|c)\ \forall c$$

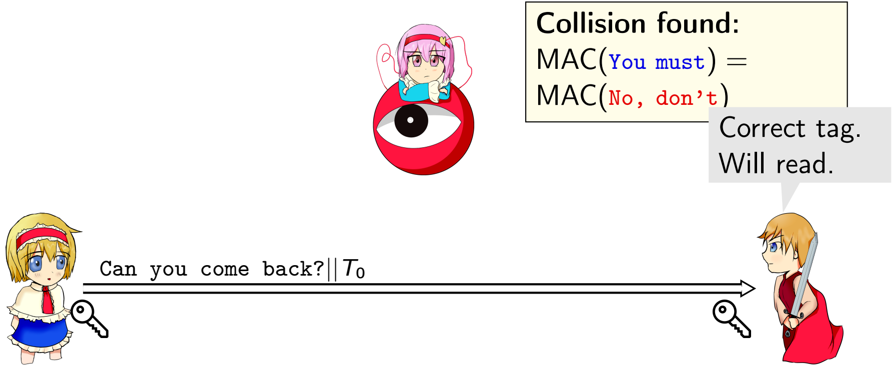
Collision found:
MAC(`You must`) =
MAC(`No, don't`)

`Can you come back?`$\|T_0$ ⟹

Forgery from collisions

Expansion property

$$\text{MAC}(m) = \text{MAC}(m') \implies \text{MAC}(m||c) = \text{MAC}(m'||c)\ \forall c$$



Forgery from collisions

Expansion property

$$\mathrm{MAC}(m) = \mathrm{MAC}(m') \implies \mathrm{MAC}(m||c) = \mathrm{MAC}(m'||c)\ \forall c$$

Tell Bob **he must** come back!

Collision found:
MAC(`You must`) =
MAC(`No, don't`)

Oh you are right!

Forgery from collisions

Expansion property

$\mathsf{MAC}(m) = \mathsf{MAC}(m') \implies \mathsf{MAC}(m||c) = \mathsf{MAC}(m'||c)\ \forall c$

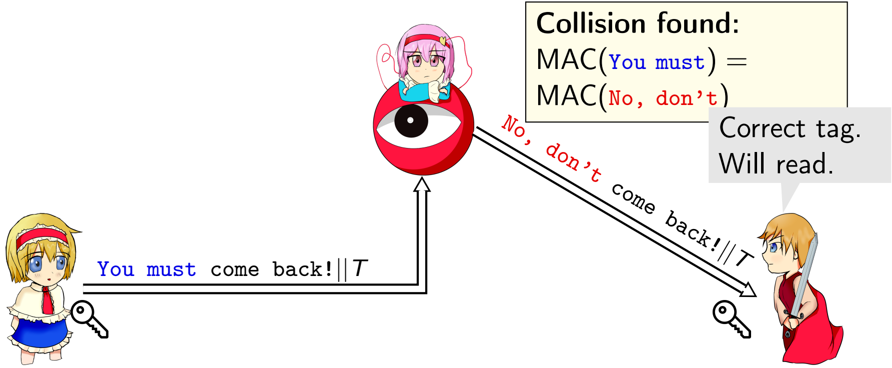
Collision found:
MAC(`You must`) =
MAC(`No, don't`)

`You must come back!`$||T$

Forgery from collisions

Expansion property

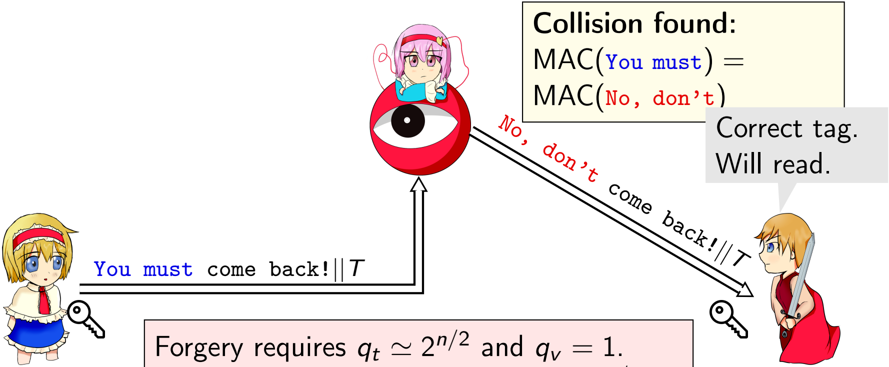
$$\mathrm{MAC}(m) = \mathrm{MAC}(m') \implies \mathrm{MAC}(m||c) = \mathrm{MAC}(m'||c)\ \forall c$$



Forgery from collisions

Expansion property

$$\mathrm{MAC}(m) = \mathrm{MAC}(m') \implies \mathrm{MAC}(m||c) = \mathrm{MAC}(m'||c)\ \forall c$$



Forgery requires $q_t \simeq 2^{n/2}$ and $q_v = 1$.
Not secure beyond birthday bound ($2^{n/2}$)

Going beyond

Problem

How to build a deterministic MAC scheme secure when $q_t > 2^{n/2}$?

Not so easy: This birthday bound attack is generic to all deterministic iterated MAC constructions with an n-bit internal state [Preneel, van Oorschot, CRYPTO'95].

Going beyond

Problem

How to build a deterministic MAC scheme secure when $q_t > 2^{n/2}$?

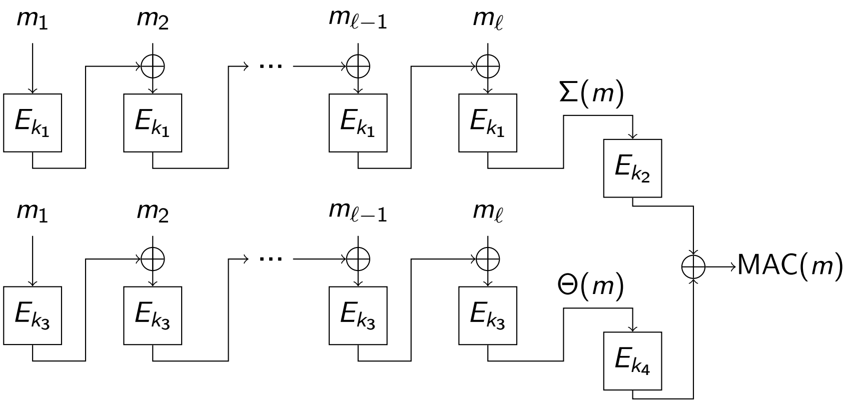
Not so easy: This birthday bound attack is generic to all deterministic iterated MAC constructions with an n-bit internal state [Preneel, van Oorschot, CRYPTO'95].

Idea: Double the size of the internal state to $2n$ bits.

Double-Block-Hash-Then-Sum Approach

XOR the two half-states at the end to recover an n-bit MAC.
Important research effort exploring this idea including:
SUM-ECBC, PMAC+, 3kf9, LightMAC+, GCM-SIV2, 1kPMAC+

Example: SUM-ECBC [Yasuda; CT-RSA'10]



$$\mathrm{MAC}(m) = E_{k_2}(\Sigma(m)) \oplus E_{k_4}(\Theta(m))$$

This paper

Problem

Many of those schemes are proven secure when $q_t < 2^{2n/3}$.
What happens when $q_t \geq 2^{2n/3}$?
Actual attacks or proof artefact?

This paper

Problem

Many of those schemes are proven secure when $q_t < 2^{2n/3}$.
What happens when $q_t \geq 2^{2n/3}$?
Actual attacks or proof artefact?

Results

A generic approach leading to an attack on all cited schemes using $q_v = 1$ and $q_t \simeq 2^{3n/4}$.

4-way collision for double-hash-then-sum schemes

Look for a quadruple of messages X, Y, Z, T that satisfies:

$$\mathcal{R}(X, Y, Z, T) := \begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(Y) = \Theta(Z) \\ \Sigma(Z) = \Sigma(T) \\ \Theta(T) = \Theta(X) \end{cases}$$

$$\mathcal{R}(X, Y, Z, T) \implies \mathrm{MAC}(X) \oplus \mathrm{MAC}(Y) \oplus \mathrm{MAC}(Z) \oplus \mathrm{MAC}(T) = 0$$

$$\mathrm{MAC}(X) = E(\Sigma(X)) \oplus E'(\Theta(X)) \qquad E'(\Theta(T)) \oplus E(\Sigma(T)) = \mathrm{MAC}(T)$$

$$\mathrm{MAC}(Y) = E(\Sigma(Y)) \oplus E'(\Theta(Y)) \qquad E'(\Theta(Z)) \oplus E(\Sigma(Z)) = \mathrm{MAC}(Z)$$

with $E(\Sigma(X)) = E(\Sigma(Y))$, $E'(\Theta(X)) = E'(\Theta(T))$, $E(\Sigma(T)) = E(\Sigma(Z))$, $E'(\Theta(Y)) = E'(\Theta(Z))$.

4-way collision for double-hash-then-sum schemes

With carefully crafted sets of messages for X, Y, Z, T:

$$\begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(Y) = \Theta(Z) \\ \Sigma(Z) = \Sigma(T) \end{cases} \implies \Theta(T) = \Theta(X).$$

Thus $\mathcal{R}(X, Y, Z, T) \iff \begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(Y) = \Theta(Z) \\ \Sigma(Z) = \Sigma(T) \end{cases}$ a 3n-bit condition.

4-way collision for double-hash-then-sum schemes

With carefully crafted sets of messages for X, Y, Z, T:

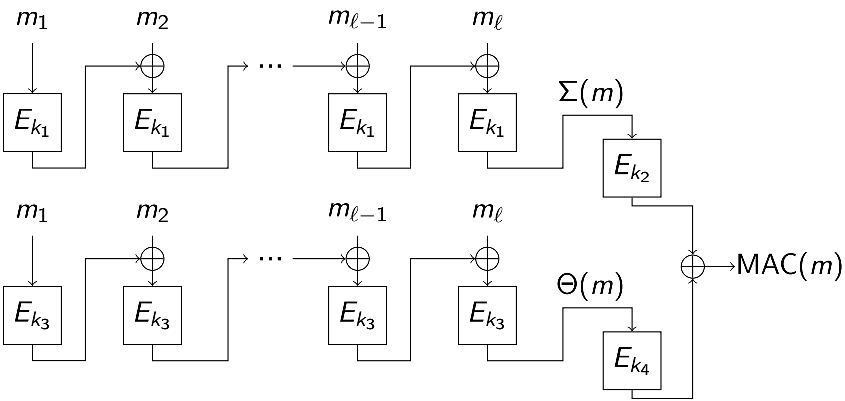
$$\begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(Y) = \Theta(Z) \\ \Sigma(Z) = \Sigma(T) \end{cases} \implies \Theta(T) = \Theta(X).$$

Thus $\mathcal{R}(X, Y, Z, T) \iff \begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(Y) = \Theta(Z) \\ \Sigma(Z) = \Sigma(T) \end{cases}$ a $3n$-bit condition.

Query complexity

There are $\simeq q_t^4$ quadruples for a $3n$-bit condition.
A good one with high probability after $q_t \simeq 2^{3n/4}$ queries.

Attack on SUM-ECBC



$$\mathrm{MAC}(m) = E_{k_2}(\Sigma(m)) \oplus E_{k_4}(\Theta(m))$$

Crafting the messages

$$X = 0||x; \qquad Y = 1||y; \qquad Z = 0||z; \qquad T = 1||t;$$

Crafting the messages

$$X = 0||x; \quad Y = 1||y; \quad Z = 0||z; \quad T = 1||t;$$

$$\mathcal{R} := \begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(Y) = \Theta(Z) \\ \Sigma(Z) = \Sigma(T) \\ \Theta(T) = \Theta(X) \end{cases} \iff \begin{cases} E_{k_1}(x \oplus E_{k_1}(0)) = E_{k_1}(y \oplus E_{k_1}(1)) \\ E_{k_3}(y \oplus E_{k_3}(1)) = E_{k_3}(z \oplus E_{k_3}(0)) \\ E_{k_1}(z \oplus E_{k_1}(0)) = E_{k_1}(t \oplus E_{k_1}(1)) \\ E_{k_3}(t \oplus E_{k_3}(1)) = E_{k_3}(x \oplus E_{k_3}(0)) \end{cases}$$

Crafting the messages

$$X = 0||x; \quad Y = 1||y; \quad Z = 0||z; \quad T = 1||t;$$

$$\mathcal{R} := \begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(Y) = \Theta(Z) \\ \Sigma(Z) = \Sigma(T) \\ \Theta(T) = \Theta(X) \end{cases} \iff \begin{cases} E_{k_1}(x \oplus E_{k_1}(0)) = E_{k_1}(y \oplus E_{k_1}(1)) \\ E_{k_3}(y \oplus E_{k_3}(1)) = E_{k_3}(z \oplus E_{k_3}(0)) \\ E_{k_1}(z \oplus E_{k_1}(0)) = E_{k_1}(t \oplus E_{k_1}(1)) \\ E_{k_3}(t \oplus E_{k_3}(1)) = E_{k_3}(x \oplus E_{k_3}(0)) \end{cases}$$

$$\iff \begin{cases} x \oplus E_{k_1}(0) = y \oplus E_{k_1}(1) \\ y \oplus E_{k_3}(1) = z \oplus E_{k_3}(0) \\ z \oplus E_{k_1}(0) = t \oplus E_{k_1}(1) \\ t \oplus E_{k_3}(1) = x \oplus E_{k_3}(0) \end{cases} \iff \begin{cases} x \oplus y \oplus z \oplus t = 0 \\ x \oplus y = E_{k_1}(0) \oplus E_{k_1}(1) \\ x \oplus t = E_{k_3}(0) \oplus E_{k_3}(1) \end{cases}$$

$\mathcal{R}(X, Y, Z, T)$ is indeed a $3n$-bit condition on the quadruple.

Filtering quadruples

$$\mathcal{R} \iff \begin{cases} x \oplus y \oplus z \oplus t = 0 \\ x \oplus y = E_{k_1}(0) \oplus E_{k_1}(1) \\ x \oplus t = E_{k_3}(0) \oplus E_{k_3}(1) \end{cases}$$

Observable Filters

The first equation of $\mathcal{R}$ in addition to the sum of MACs:

$$\begin{cases} x \oplus y \oplus z \oplus t = 0 \\ \mathrm{MAC}(0||x) \oplus \mathrm{MAC}(1||y) \oplus \mathrm{MAC}(0||z) \oplus \mathrm{MAC}(1||t) = 0 \end{cases}$$

Filtering quadruples

$$\mathcal{R} \iff \begin{cases} x \oplus y \oplus z \oplus t = 0 \\ x \oplus y = E_{k_1}(0) \oplus E_{k_1}(1) \\ x \oplus t = E_{k_3}(0) \oplus E_{k_3}(1) \end{cases}$$

Observable Filters

The first equation of $\mathcal{R}$ in addition to the sum of MACs:

$$\begin{cases} x \oplus y \oplus z \oplus t = 0 \\ \mathrm{MAC}(0||x) \oplus \mathrm{MAC}(1||y) \oplus \mathrm{MAC}(0||z) \oplus \mathrm{MAC}(1||t) = 0 \end{cases}$$

Not enough

It is a $2n$-bit filter for $q_t^4 \simeq 2^{3n}$ quadruples.
2^n quadruples to randomly pass the filter for only 1 respecting $\mathcal{R}$.

Amplifying the filter

$$\mathcal{R}\big((0||x),(1||y),(0||z),(1||t)\big) \iff \begin{cases} x \oplus y \oplus z \oplus t = 0 \\ x \oplus y = E_{k_1}(0) \oplus E_{k_1}(1) \\ x \oplus t = E_{k_3}(0) \oplus E_{k_3}(1) \end{cases}$$

$$\mathcal{R} \iff \begin{cases} (x \oplus 1) \oplus (y \oplus 1) \oplus (z \oplus 1) \oplus (t \oplus 1) = 0 \\ (x \oplus 1) \oplus (y \oplus 1) = E_{k_1}(0) \oplus E_{k_1}(1) \\ (x \oplus 1) \oplus (t \oplus 1) = E_{k_3}(0) \oplus E_{k_3}(1) \end{cases}$$

Amplifying the filter

$$\mathcal{R}\big((0||x),(1||y),(0||z),(1||t)\big) \iff \begin{cases} x \oplus y \oplus z \oplus t = 0 \\ x \oplus y = E_{k_1}(0) \oplus E_{k_1}(1) \\ x \oplus t = E_{k_3}(0) \oplus E_{k_3}(1) \end{cases}$$

$$\mathcal{R} \iff \begin{cases} (x \oplus 1) \oplus (y \oplus 1) \oplus (z \oplus 1) \oplus (t \oplus 1) = 0 \\ (x \oplus 1) \oplus (y \oplus 1) = E_{k_1}(0) \oplus E_{k_1}(1) \\ (x \oplus 1) \oplus (t \oplus 1) = E_{k_3}(0) \oplus E_{k_3}(1) \end{cases}$$

Related solutions

$\mathcal{R}\big((0||x),(1||y),(0||z),(1||t)\big) \iff$
$\mathcal{R}\big((0||x \oplus 1),(1||y \oplus 1),(0||z \oplus 1),(1||t \oplus 1)\big)$

In particular if we have a good solution x, y, z, t then it verifies:

$$\text{MAC}(0||x\oplus 1)\oplus\text{MAC}(1||y\oplus 1)\oplus\text{MAC}(0||z\oplus 1)\oplus\text{MAC}(1||t\oplus 1) = 0$$

Finding a good quadruple

Find a quadruple (x, y, z, t) such that:

$$
\begin{aligned}
&x && \oplus y && \oplus z && \oplus t && = 0 \\
&\mathrm{MAC}(0||x) && \oplus \mathrm{MAC}(1||y) && \oplus \mathrm{MAC}(0||z) && \oplus \mathrm{MAC}(1||t) && = 0 \\
&\mathrm{MAC}(0||x \oplus 1) && \oplus \mathrm{MAC}(1||y \oplus 1) && \oplus \mathrm{MAC}(0||z \oplus 1) && \oplus \mathrm{MAC}(1||t \oplus 1) && = 0
\end{aligned}
$$

Finding a good quadruple

Find a quadruple (x, y, z, t) such that:

$$\begin{array}{lllll} x & \oplus\, y & \oplus\, z & \oplus\, t & = 0 \\ \mathrm{MAC}(0||x) & \oplus \mathrm{MAC}(1||y) & \oplus \mathrm{MAC}(0||z) & \oplus \mathrm{MAC}(1||t) & = 0 \\ \mathrm{MAC}(0||x \oplus 1) & \oplus \mathrm{MAC}(1||y \oplus 1) & \oplus \mathrm{MAC}(0||z \oplus 1) & \oplus \mathrm{MAC}(1||t \oplus 1) & = 0 \end{array}$$

1. Query and build the following 4 lists of size $2^{3n/4}$:

$$\begin{aligned} L_1 &= \{x||\,\mathrm{MAC}(0||x)||\,\mathrm{MAC}(0||x \oplus 1)\} \\ L_2 &= \{y||\,\mathrm{MAC}(1||y)||\,\mathrm{MAC}(1||y \oplus 1)\} \\ L_3 &= \{z||\,\mathrm{MAC}(0||z)||\,\mathrm{MAC}(0||z \oplus 1)\} \\ L_4 &= \{t||\,\mathrm{MAC}(1||t)||\,\mathrm{MAC}(1||t \oplus 1)\} \end{aligned}$$

Finding a good quadruple

Find a quadruple (x, y, z, t) such that:

$$
\begin{array}{lllll}
x & \oplus\, y & \oplus\, z & \oplus\, t & = 0 \\
\mathrm{MAC}(0||x) & \oplus\, \mathrm{MAC}(1||y) & \oplus\, \mathrm{MAC}(0||z) & \oplus\, \mathrm{MAC}(1||t) & = 0 \\
\mathrm{MAC}(0||x \oplus 1) & \oplus\, \mathrm{MAC}(1||y \oplus 1) & \oplus\, \mathrm{MAC}(0||z \oplus 1) & \oplus\, \mathrm{MAC}(1||t \oplus 1) & = 0
\end{array}
$$

1. Query and build the following 4 lists of size $2^{3n/4}$:

$$
\begin{aligned}
L_1 &= \{x||\,\mathrm{MAC}(0||x)||\,\mathrm{MAC}(0||x \oplus 1)\} \\
L_2 &= \{y||\,\mathrm{MAC}(1||y)||\,\mathrm{MAC}(1||y \oplus 1)\} \\
L_3 &= \{z||\,\mathrm{MAC}(0||z)||\,\mathrm{MAC}(0||z \oplus 1)\} \\
L_4 &= \{t||\,\mathrm{MAC}(1||t)||\,\mathrm{MAC}(1||t \oplus 1)\}
\end{aligned}
$$

2. Find $\ell_1, \ell_2, \ell_3, \ell_4$ in L_1, L_2, L_3, L_4 respectively such that $\ell_1 \oplus \ell_2 \oplus \ell_3 \oplus \ell_4 = 0$.

Finding a good quadruple

1. Query and build L_1, L_2, L_3, L_4 of size $2^{3n/4}$.
2. Find $\ell_1, \ell_2, \ell_3, \ell_4$ in L_1, L_2, L_3, L_4 respectively such that $\ell_1 \oplus \ell_2 \oplus \ell_3 \oplus \ell_4 = 0$.

Algorithm cost

Step 1 costs $q_t = \mathcal{O}(2^{3n/4})$ queries and as much memory.

Step 2 is about solving an instance of the 4-XOR problem.
Solve it in $\mathcal{O}(2^{3n/4})$ memory and $\mathcal{O}(2^{3n/2})$ time.

Optimizing time complexity

SUM-ECBC and GCM-SIV2: optimize the time complexity at the cost of queries.

Optimizing time complexity

SUM-ECBC and GCM-SIV2: optimize the time complexity at the cost of queries.

Related solutions

$$\mathcal{R}\big((0||x),(1||y),(0||z),(1||t)\big) \iff$$
$$\mathcal{R}\big((0||x\oplus c),(1||y\oplus c),(0||z\oplus c),(1||t\oplus c)\big)\ \forall c$$

So $\mathcal{R} \implies \forall c :$
$$\mathrm{MAC}(0||x\oplus c)\oplus\mathrm{MAC}(1||y\oplus c)\oplus\mathrm{MAC}(0||z\oplus c)\oplus\mathrm{MAC}(1||t\oplus c) = 0$$

Optimizing time complexity

Let $\mathcal{C} = \{c : c < 2^{3n/7}\}$ we sum the relations:

$$\bigoplus \begin{cases} \mathrm{MAC}(0||x \oplus 0) \oplus \mathrm{MAC}(1||y \oplus 0) \oplus \mathrm{MAC}(0||z \oplus 0) \oplus \mathrm{MAC}(1||t \oplus 0) = 0 \\ \mathrm{MAC}(0||x \oplus 1) \oplus \mathrm{MAC}(1||y \oplus 1) \oplus \mathrm{MAC}(0||z \oplus 1) \oplus \mathrm{MAC}(1||t \oplus 1) = 0 \\ \mathrm{MAC}(0||x \oplus 2) \oplus \mathrm{MAC}(1||y \oplus 2) \oplus \mathrm{MAC}(0||z \oplus 2) \oplus \mathrm{MAC}(1||t \oplus 2) = 0 \\ \mathrm{MAC}(0||x \oplus 3) \oplus \mathrm{MAC}(1||y \oplus 3) \oplus \mathrm{MAC}(0||z \oplus 3) \oplus \mathrm{MAC}(1||t \oplus 3) = 0 \\ \mathrm{MAC}(0||x \oplus 4) \oplus \mathrm{MAC}(1||y \oplus 4) \oplus \mathrm{MAC}(0||z \oplus 4) \oplus \mathrm{MAC}(1||t \oplus 4) = 0 \\ \dots \end{cases}$$

Optimizing time complexity

Let $\mathcal{C} = \{c : c < 2^{3n/7}\}$ we sum the relations:

$$\bigoplus_{c \in \mathcal{C}} \mathsf{MAC}(0||x \oplus c) \oplus \mathsf{MAC}(1||y \oplus c) \oplus \mathsf{MAC}(0||z \oplus c) \oplus \mathsf{MAC}(1||t \oplus c) = 0$$

Optimizing time complexity

Let $\mathcal{C} = \{c : c < 2^{3n/7}\}$ we sum the relations:

$$\bigoplus_{c\in\mathcal{C}} \mathsf{MAC}(0||x\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathsf{MAC}(1||y\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathsf{MAC}(0||z\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathsf{MAC}(1||t\oplus c) = 0$$

Optimizing time complexity

Let $\mathcal{C} = \{c : c < 2^{3n/7}\}$ we sum the relations:

$$\bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(0||x\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(1||y\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(0||z\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(1||t\oplus c) = 0$$

Only the most significant $4n/7$ bits of x, y, z, t are meaningful and must respect a $3 \cdot 4n/7 = 12n/7$-bit relationship.

Optimizing time complexity

Let $\mathcal{C} = \{c : c < 2^{3n/7}\}$ we sum the relations:

$$\bigoplus_{c \in \mathcal{C}} \mathrm{MAC}(0||x \oplus c) \oplus \bigoplus_{c \in \mathcal{C}} \mathrm{MAC}(1||y \oplus c) \oplus \bigoplus_{c \in \mathcal{C}} \mathrm{MAC}(0||z \oplus c) \oplus \bigoplus_{c \in \mathcal{C}} \mathrm{MAC}(1||t \oplus c) = 0$$

Only the most significant $4n/7$ bits of x, y, z, t are meaningful and must respect a $3 \cdot 4n/7 = 12n/7$-bit relationship.

$$L_1 = \left\{ x_{[3n/7:n]} || \bigoplus_{c \in \mathcal{C}} \mathrm{MAC}(0||x \oplus c) || \bigoplus_{c \in \mathcal{C}} \mathrm{MAC}(0||(x \oplus \delta) \oplus c) \right\}$$

For $|L| = 2^{3n/7}$ the 4-XOR problem takes $\mathcal{O}(2^{6n/7})$ time.
One element requires $2^{3n/7}$ queries, a total of $\mathcal{O}(2^{6n/7})$ queries.

Optimizing time complexity

Let $\mathcal{C} = \{c : c < 2^{3n/7}\}$ we sum the relations:

$$\bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(0||x\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(1||y\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(0||z\oplus c) \oplus \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(1||t\oplus c) = 0$$

Only the most significant $4n/7$ bits of x, y, z, t are meaningful and must respect a $3 \cdot 4n/7 = 12n/7$-bit relationship.

$$L_1 = \left\{ x_{[3n/7:n]} || \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(0||x \oplus c) || \bigoplus_{c\in\mathcal{C}} \mathrm{MAC}(0||(x \oplus \delta) \oplus c) \right\}$$

For $|L| = 2^{3n/7}$ the 4-XOR problem takes $\mathcal{O}(2^{6n/7})$ time.
One element requires $2^{3n/7}$ queries, a total of $\mathcal{O}(2^{6n/7})$ queries.
Previously we used $\mathcal{O}(2^{3n/2})$ time and $\mathcal{O}(2^{3n/4})$ queries.
Thus this optimization uses less time but more queries.

Forgery from quadruples

$\Sigma(m)$ and $\Theta(m)$ are built the same way as simple ECBC's $\Sigma(m)$.
In particular for all suffixes c:

$$\Sigma(m) = \Sigma(m') \implies \Sigma(m||c) = \Sigma(m'||c)$$

The same holds for Θ.

Forgery from quadruples

$\Sigma(m)$ and $\Theta(m)$ are built the same way as simple ECBC's $\Sigma(m)$.
In particular for all suffixes c:

$$\Sigma(m) = \Sigma(m') \implies \Sigma(m||c) = \Sigma(m'||c)$$

The same holds for Θ.

Expansion property SUM-ECBC

$\mathcal{R}(X, Y, Z, T) \implies \mathcal{R}(X||c, Y||c, Z||c, T||c)\ \forall c$

Therefore Eve can forge in a very similar manner.

Forgery from quadruples

Expansion property SUM-ECBC (reminder)

$$\mathcal{R}(X, Y, Z, T) \implies \mathcal{R}(X||c, Y||c, Z||c, T||c)\ \forall c$$

Quadruple found:
MAC(`You should`)
MAC(`Plz help`)
MAC(`You must`)
MAC(`Plz never`)

Tell Bob he **should** come back!

T_1

`You should come back!`$||T_1$

Forgery from quadruples

Expansion property SUM-ECBC (reminder)

$$\mathcal{R}(X, Y, Z, T) \implies \mathcal{R}(X||c, Y||c, Z||c, T||c)\ \forall c$$

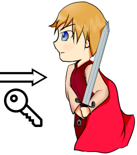
Quadruple found:

MAC(`You should`)
MAC(`Plz help`)
MAC(`You must`)
MAC(`Plz never`)

T_1

Correct tag.
Will read.

`You should come back!`$||T_1$



Forgery from quadruples

Expansion property SUM-ECBC (reminder)

$$\mathcal{R}(X, Y, Z, T) \implies \mathcal{R}(X||c, Y||c, Z||c, T||c)\ \forall c$$

Quadruple found:

MAC(`You should`)

MAC(`Plz help`)

MAC(`You must`)

MAC(`Plz never`)

Plz help tell Bob to come back!

T_1, T_2

`Plz help come back!`$||T_2$

Forgery from quadruples

Expansion property SUM-ECBC (reminder)

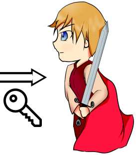
$\mathcal{R}(X, Y, Z, T) \implies \mathcal{R}(X||c, Y||c, Z||c, T||c)\ \forall c$

Quadruple found:
MAC(`You should`)
MAC(`Plz help`)
MAC(`You must`)
MAC(`Plz never`)

T_1, T_2

Correct tag.
Will read.

`Plz help come back!`$||T_2$



Forgery from quadruples

Expansion property SUM-ECBC (reminder)

$$\mathcal{R}(X, Y, Z, T) \implies \mathcal{R}(X||c, Y||c, Z||c, T||c)\ \forall c$$

Quadruple found:

MAC(`You should`)

MAC(`Plz help`)

MAC(`You must`)

MAC(`Plz never`)

Tell Bob he **must** come back!

T_1, T_2, T_3

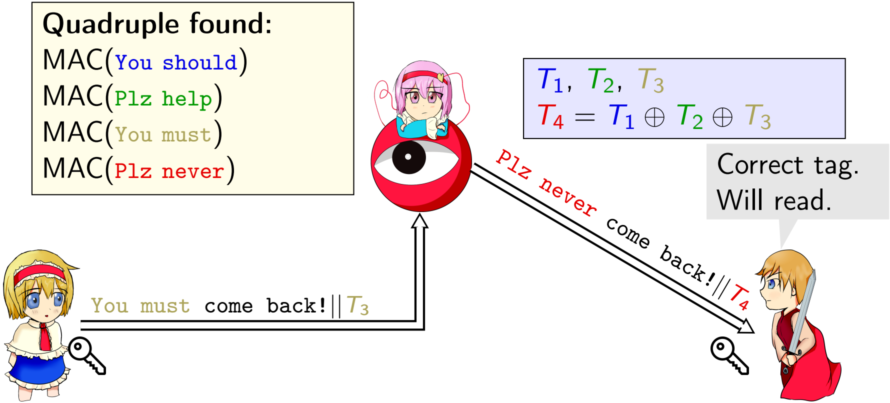
$T_4 = T_1 \oplus T_2 \oplus T_3$

`You must come back!`$||T_3$ →

Forgery from quadruples

Expansion property SUM-ECBC (reminder)

$$\mathcal{R}(X, Y, Z, T) \implies \mathcal{R}(X||c, Y||c, Z||c, T||c)\ \forall c$$



Conclusion

Main results:

- Most of our attacks use $2^{3n/4}$ queries and $2^{3n/2}$ time.
- Variant for SUM-ECBC & GCM-SIV2: $2^{6n/7}$ queries and time.

3kf9 [Zhang, Wu, Sui, Wang; AC'12]

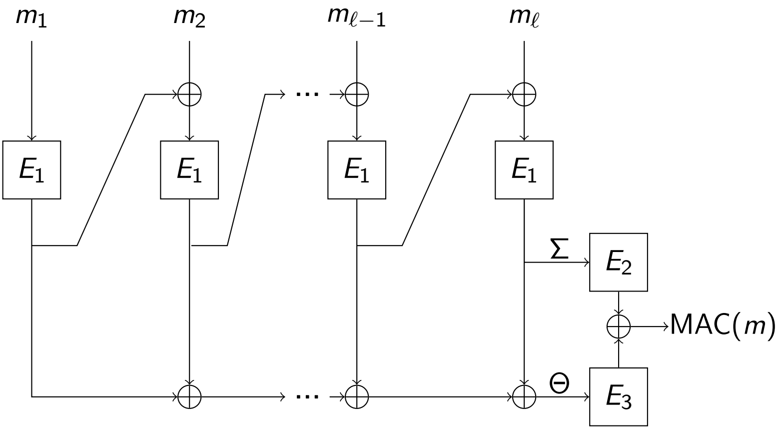
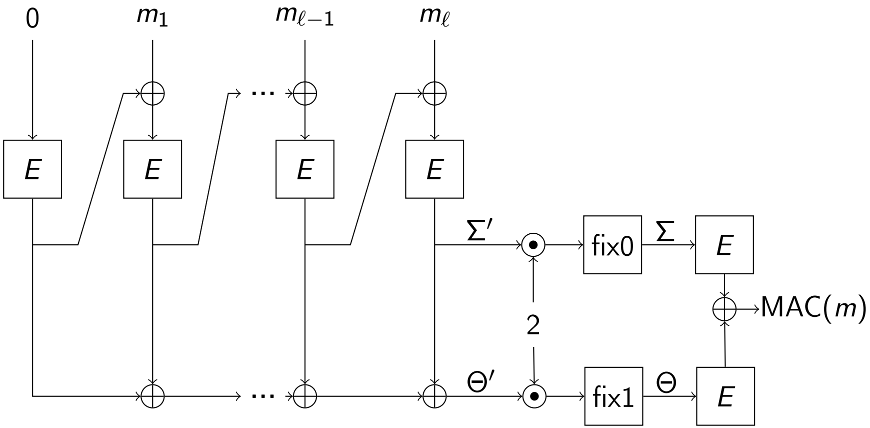
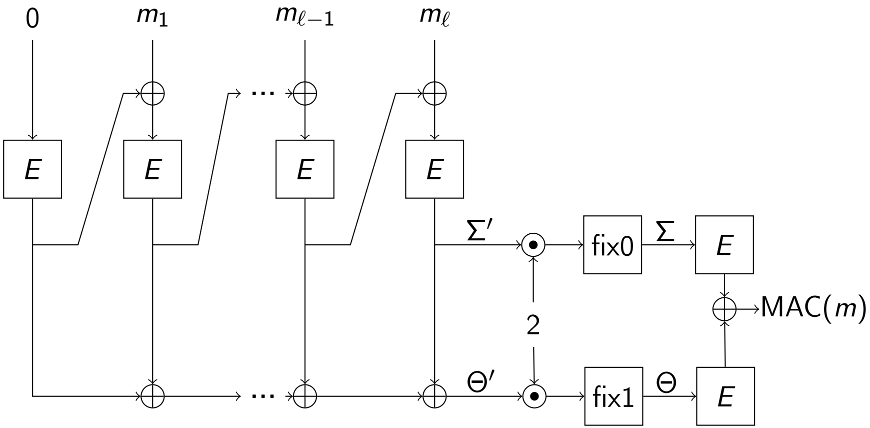


Figure: Diagram for 3kf9 with an $\ell-$block message.

1kf9[Datta, Dutta, Nandi, Paul, Zhang; 2015, withdrawn'17]



1kf9[Datta, Dutta, Nandi, Paul, Zhang; 2015, withdrawn'17]



$$\mathcal{R}(X,Y) := \begin{cases} \Sigma'(X) = \Sigma'(Y) \\ 2\Theta'(X) = 2\Theta'(Y) \oplus 1 \end{cases} \implies \begin{cases} \Sigma(X) = \Sigma(Y) \\ \Theta(X) = \Theta(Y) \end{cases}$$

Crafting the messages

$$X = x||0; \qquad Y = y||d; \quad \text{where } d = 2^{-1}$$

Crafting the messages

$$X = x||0; \qquad Y = y||d; \quad \text{where } d = 2^{-1}$$

$$\mathcal{R}(X, Y) \iff \begin{cases} E\big(0 \oplus E(x \oplus E(0))\big) = E\big(E(y \oplus E(0)) \oplus d\big) \\ \Theta'(X) = \Theta'(Y) \oplus d \end{cases}$$

Crafting the messages

$$X = x||0; \qquad Y = y||d; \quad \text{where } d = 2^{-1}$$

$$\mathcal{R}(X,Y) \iff \begin{cases} E\big(0 \oplus E(x \oplus E(0))\big) = E\big(E(y \oplus E(0)) \oplus d\big) \\ \Theta'(X) = \Theta'(Y) \oplus d \end{cases}$$

$$\Leftrightarrow \begin{cases} E\big(x \oplus E(0)\big) = E\big(y \oplus E(0)\big) \oplus d \\ E(0) \oplus E\big(x \oplus E(0)\big) \oplus \Sigma'(X) = E(0) \oplus E\big(y \oplus E(0)\big) \oplus \Sigma'(Y) \oplus d \end{cases}$$

Crafting the messages

$$X = x||0; \qquad Y = y||d; \quad \text{where } d = 2^{-1}$$

$$\mathcal{R}(X,Y) \iff \begin{cases} E\big(0 \oplus E(x \oplus E(0))\big) = E\big(E(y \oplus E(0)) \oplus d\big) \\ \Theta'(X) = \Theta'(Y) \oplus d \end{cases}$$

$$\Leftrightarrow \begin{cases} E\big(x \oplus E(0)\big) = E\big(y \oplus E(0)\big) \oplus d \\ E(0) \oplus E\big(x \oplus E(0)\big) \oplus \Sigma'(X) = E(0) \oplus E\big(y \oplus E(0)\big) \oplus \Sigma'(Y) \oplus d \end{cases}$$

$$\iff \begin{cases} E\big(x \oplus E(0)\big) = E\big(y \oplus E(0)\big) \oplus d \\ E\big(x \oplus E(0)\big) = E\big(y \oplus E(0)\big) \oplus d \end{cases} \iff \Sigma'(X) = \Sigma'(Y)$$

$\mathcal{R}(X,Y)$ is an n-bit relation on a couple: Birthday Bound!
Look for collision $\text{MAC}(X) = \text{MAC}(Y)$.

Discussion

Easy forge after found collision:

Expansion property 1kf9

$\mathcal{R}(X, Y) \implies \mathcal{R}(X||c, Y||c)\ \forall c$

Different multiplications can't help:
Full collision on Σ'
Set $d :=$ inverse of the Θ' multiplication.

Conclusion

Main results:

- Most of our attacks use $2^{3n/4}$ queries and $2^{3n/2}$ time.
- Variant for SUM-ECBC & GCM-SIV2: $2^{6n/7}$ queries and time.

Additionally:

- Withdrawn 1kf9 shown to allow Birthday Bound Attacks and therefore is not a BBB scheme.
- Recent results on security of LightMAC+ [Naito, CT-RSA'18] proved wrong by our attack.

Conclusion

Mode	Attacks (this work) Queries	Time	Type
SUM-ECBC	$\mathcal{O}(2^{3n/4})$	$\tilde{\mathcal{O}}(2^{3n/2})$	Universal
	$\mathcal{O}(2^{6n/7})$	$\tilde{\mathcal{O}}(2^{6n/7})$	Universal
GCM-SIV2	$\mathcal{O}(2^{3n/4})$	$\tilde{\mathcal{O}}(2^{3n/2})$	Universal
	$\mathcal{O}(2^{6n/7})$	$\tilde{\mathcal{O}}(2^{6n/7})$	Universal
PMAC+	$\mathcal{O}(2^{3n/4})$	$\tilde{\mathcal{O}}(2^{3n/2})$	Existential
LightMAC+	$\mathcal{O}(2^{3n/4})$	$\tilde{\mathcal{O}}(2^{3n/2})$	Existential
1kPMAC+	$\mathcal{O}(2^{3n/4})$	$\tilde{\mathcal{O}}(2^{3n/2})$	Existential
3kf9	$\mathcal{O}(\sqrt[4]{n} \cdot 2^{3n/4})$	$\tilde{\mathcal{O}}(2^{5n/4})$	Universal
1kf9	$\mathcal{O}(2^{n/2})$	$\tilde{\mathcal{O}}(2^{n/2})$	Universal

Except 1kf9, all above schemes have a proof that they are secure while $q_t < 2^{2n/3}$. We showed they are not secure when $q_t \geq 2^{3n/4}$.
Open question: What happens when $2^{2n/3} \leq q_t < 2^{3n/4}$?