RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

MATTERS NOW

SESSION ID: CRYP-R04

ZERO-SUM PARTITIONS OF PHOTON PERMUTATIONS

Lorenzo Grassi

Ph.D. Student
IAIK, Graz

Zero-Sum Partitions of PHOTON Permutations

Qingju Wang, Lorenzo Grassi, Christian Rechberger

April, 2018

Introduction (1/2)

Hash functions are one of the most important primitives in symmetric-key cryptography.

Sponge functions are a way of building hash functions from a fixed permutation.

Modern cryptanalytic approaches target both hash function primitives and underlying ciphers or permutations.

Introduction (2/2)

PHOTON [GPP11] is a (lightweight) family of sponge-like hash proposed at CRYPTO 2011 and recently standardized by ISO.

W.r.t. the security claims made by the designers, *we show - for the first time - zero-sum partitions for (almost) all of those full 12-round (inner) permutation variants that use a 4-bit S-Box.*

Our results are theoretical in nature:

there is currently no reason to believe that the security of PHOTON as a hash function is endangered.

Introduction (2/2)

PHOTON [GPP11] is a (lightweight) family of sponge-like hash proposed at CRYPTO 2011 and recently standardized by ISO.

W.r.t. the security claims made by the designers, *we show - for the first time - zero-sum partitions for (almost) all of those full 12-round (inner) permutation variants that use a 4-bit S-Box.*

Our results are theoretical in nature:

there is currently no reason to believe that the security of PHOTON as a hash function is endangered.

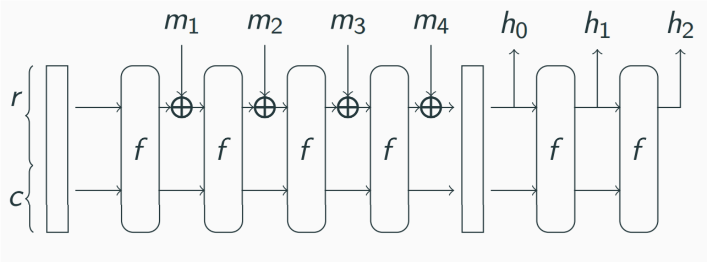
Table of Contents

Part I

PHOTON

PHOTON [GPP11]

PHOTON is a (lightweight) family of sponge-like hash function



PHOTON Family

5 Variants of PHOTON denoted by PHOTON-$n/r/r'$:

- n is the bit-size of the hash output
- r and r' are the input and the output bit rate respectively
- c is the bit-size of the capacity part of the internal state
- $t = c + r$ is the internal state size

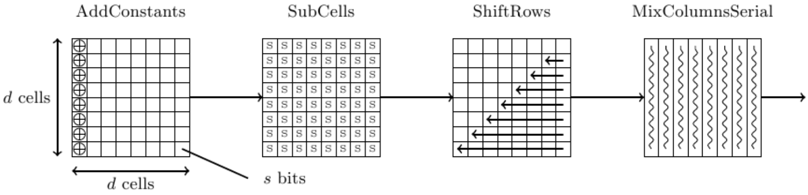
Table: Parameters of PHOTON-$n/r/r'$ with **4-bit S-Box**

Versions	t	n	c	r	r'	d
PHOTON-80/20/16	100	80	80	20	16	5
PHOTON-128/16/16	144	128	128	16	16	6
PHOTON-160/36/36	196	160	160	36	36	7
PHOTON-224/32/32	256	224	224	32	32	8

PHOTON Permutation

The internal state is viewed as a $d \times d$ matrix of 4-bit cells.

The internal Permutation of PHOTON iterates 12 times a round composed of 4 operations:



Part II

Zero-Sum

Zero-Sum

Let F be a function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_{2^m}$.

A zero-sum for F of size K is a subset $\{x_1, \dots, x_K\} \subset \mathbb{F}_{2^n}$ of elements which sum to zero and for which the corresponding images by F also sum to zero, i.e.

$$\bigoplus_{i=1}^{K} x_i = \bigoplus_{i=1}^{K} F(x_i) = 0.$$

Given a function F and an affine subspace $\mathcal{A} \subset \mathbb{F}_{2^n}$ with dimension $(deg(F) + 1)$, then

$$\bigoplus_{x \in \mathcal{A}} F(x) = 0.$$

Zero-Sum

Let F be a function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_{2^m}$.

A zero-sum for F of size K is a subset $\{x_1, \ldots, x_K\} \subset \mathbb{F}_{2^n}$ of elements which sum to zero and for which the corresponding images by F also sum to zero, i.e.

$$\bigoplus_{i=1}^{K} x_i = \bigoplus_{i=1}^{K} F(x_i) = 0.$$

Given a function F and an affine subspace $\mathcal{A} \subset \mathbb{F}_{2^n}$ with dimension $(deg(F) + 1)$, then

$$\bigoplus_{x \in \mathcal{A}} F(x) = 0.$$

Zero-Sum Partition

Let F be a function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_{2^m}$.

A zero-sum partition for F of size $K = 2^k$ is a collection of 2^{n-k} disjoint sets $\{X_1, \ldots, X_{2^{n-k}}\}$ with the following properties

- $X_i = \{x_1, \ldots, x_{2^k}\} \subset \mathbb{F}_{2^n}$ for each i such that

$$\bigcup_i X_i = \mathbb{F}_{2^n};$$

- for each $X_i = \{x_1, \ldots, x_{2^k}\}$

$$\bigoplus_{i=1}^{2^k} x_i = \bigoplus_{i=1}^{2^k} F(x_i) = 0.$$

Zero-Sum: Inside-Out Approach (1/2)

Given a permutation P

$$P(\cdot) = R^{r+s}(\cdot) \equiv \underbrace{R \circ \cdots \circ R}_{r+s \text{ times}}(\cdot)$$

how to construct a zero-sum?

(1st) Consider affine subspaces $X = \{x^i\}_i$ and $Y = \{y^i\}_i$ such that

$$\bigoplus_i R^{-s}(y^i) = 0 \qquad \bigoplus_i R^{r}(x^i) = 0.$$

The previous equalities are satisfied if

$$dim(X) \geq deg(R^r) + 1 \quad \text{and} \quad dim(Y) \geq deg(R^{-s}) + 1$$

Zero-Sum: Inside-Out Approach (1/2)

Given a permutation P

$$P(\cdot) = R^{r+s}(\cdot) \equiv \underbrace{R \circ \cdots \circ R}_{r+s \text{ times}}(\cdot)$$

how to construct a zero-sum?

(1st) Consider affine subspaces $X = \{x^i\}_i$ and $Y = \{y^i\}_i$ such that

$$\bigoplus_i R^{-s}(y^i) = 0 \qquad \bigoplus_i R^r(x^i) = 0.$$

The previous equalities are satisfied if

$$dim(X) \geq deg(R^r) + 1 \quad \text{and} \quad dim(Y) \geq deg(R^{-s}) + 1$$

Zero-Sum: Inside-Out Approach (2/2)

(2nd) Since one can work with the intermediate states, chooses texts in $X \oplus Y$:

- define the *plaintexts* p_i *as the* s*-round decryption of* $X \oplus Y$;
- define the corresponding *ciphertexts* c_i *as the* r*-round encryption of* $X \oplus Y$.

Note that:

$$X \oplus Y = \bigcup_{y \in Y} X \oplus y = \bigcup_{x \in X} Y \oplus x.$$

Result: *A zero-sum* $\{p_i\}_{i=1,\dots,K}$ *with the properties* $\bigoplus_{i=1}^{K} p_i = \bigoplus_{i=1}^{K} c_i = 0$ *is created for permutation* P.

Zero-Sum: Inside-Out Approach (2/2)

(2nd) Since one can work with the intermediate states, chooses texts in $X \oplus Y$:

- define the *plaintexts p_i as the s-round decryption of $X \oplus Y$*;
- define the corresponding *ciphertexts c_i as the r-round encryption of $X \oplus Y$*.

Note that:

$$X \oplus Y = \bigcup_{y \in Y} X \oplus y = \bigcup_{x \in X} Y \oplus x.$$

Result: *A zero-sum $\{p_i\}_{i=1,\dots,K}$ with the properties $\bigoplus_{i=1}^{K} p_i = \bigoplus_{i=1}^{K} c_i = 0$ is created for permutation P.*

Part III

MILP Automatic Tool to search Zero-Sum based on Division Property

Division Property

Division Property: "generalization" of Integral Property

Definition [Tod15] *Let* $\mathbb{X} \subset (\mathbb{F}_{2^n})^m$, *and* $k^i \in \{0, 1, 2, ..., n\}$ *for* $i = 0, ..., m-1$. $\mathbb{X}$ *has the division property* $\mathcal{D}_{\mathbf{k}}^{n,m}$ *- where* $\mathbf{k} = (k^0, k^1, ..., k^{m-1})$ *- if*

$$\bigoplus_{x \in \mathbb{X}} x^{\mathbf{u}} = 0$$

for all $\mathbf{u}$ *such that*

$$\{\mathbf{u} = (u_0, u_1, \ldots, u_{m-1}) \in (\mathbb{F}_{2^n})^m \mid (wt(u_0), ..., wt(u_{m-1})) \not\succeq \mathbf{k}\}$$

(where $wt(\cdot)$ is the Hamming weight - $a \succeq b$ means that $a^i \geq b^i$ for all i)

- Construct input multiset with division property $\mathcal{D}_{\mathbf{k}_0}^{n,m}$
- Propagate the initial division property r rounds to get the output divsion property $\mathcal{D}_{\mathbf{k}_r}^{n,m}$
- Extract useful integral from $\mathcal{D}_{\mathbf{k}_r}^{n,m}$

Division Property

Division Property: "generalization" of Integral Property

Definition [Tod15] *Let* $\mathbb{X} \subset (\mathbb{F}_{2^n})^m$*, and* $k^i \in \{0, 1, 2, ..., n\}$ *for* $i = 0, ..., m-1$. $\mathbb{X}$ *has the division property* $\mathcal{D}^{n,m}_{\mathbf{k}}$ *- where* $\mathbf{k} = (k^0, k^1, ..., k^{m-1})$ *- if*

$$\bigoplus_{x \in \mathbb{X}} x^{\mathbf{u}} = 0$$

for all $\mathbf{u}$ *such that*

$$\{\mathbf{u} = (u_0, u_1, \ldots, u_{m-1}) \in (\mathbb{F}_{2^n})^m \mid (wt(u_0), ..., wt(u_{m-1})) \not\succeq \mathbf{k}\}$$

(where $wt(\cdot)$ is the Hamming weight - $a \succeq b$ means that $a^i \geq b^i$ for all i)

- Construct input multiset with division property $\mathcal{D}^{n,m}_{\mathbf{k}_0}$
- Propagate the initial division property r rounds to get the output divsion property $\mathcal{D}^{n,m}_{\mathbf{k}_r}$
- Extract useful integral from $\mathcal{D}^{n,m}_{\mathbf{k}_r}$

Bit-based Division Property and Division Trail

Bit-Based Division Property [TM16]: *division property of each bit is treated independently*

Advantage detailed division property, longer distinguishers

Disadvantage time/memory complexity much *higher* than for division property (*upper* bounded by $O(2^n)$ where n is the block size)

$\Rightarrow$ works "only" for ciphers with small block size!

At Asiacrypt 2016, Xiang *et al.* [XZB+16] built an *automatic tool based on mixed integer linear programming (MILP) to study the division property of SPNs with bit-permutation linear layers*

Bit-based Division Property and Division Trail

Bit-Based Division Property [TM16]: *division property of each bit is treated independently*

Advantage detailed division property, longer distinguishers

Disadvantage time/memory complexity much *higher* than for division property (*upper* bounded by $O(2^n)$) where n is the block size)

$\Rightarrow$ works "only" for ciphers with small block size!

At Asiacrypt 2016, Xiang *et al.* [XZB+16] built an *automatic tool based on mixed integer linear programming (MILP) to study the division property of SPNs with bit-permutation linear layers*

MILP Automatic Tool

A MILP model $\mathcal{M}$ consists of

- variables $\mathcal{M}.var$
- linear constraints $\mathcal{M}.con$
- objective function $\mathcal{M}.obj$.

Example:

$$\begin{aligned}
\mathcal{M}.obj &\leftarrow \text{maximize } x + y + 2z \\
\mathcal{M}.con &\leftarrow x + 2y + 3z \leq 4 \\
\mathcal{M}.con &\leftarrow x + y \geq 1 \\
\mathcal{M}.var &\leftarrow x, y, z \text{ as binary.}
\end{aligned}$$

The solution to the model $\mathcal{M}$ is 3, where $(x, y, z) = (1, 0, 1)$.

MILP - Division Trail

Division Trail [XZB+16] *Assume the input multiset to a block cipher has initial division property* $\mathbb{K}_0 \equiv \mathcal{D}^{n,m}_{\mathbf{k}_0}$*, and denote the division property after* i*-round through round function* $R(\cdot)$ *by* $\mathbb{K}_i \equiv \mathcal{D}^{n,m}_{\mathbf{k}_i}$*. We have the following trail of division property propagations:*

$$\mathbb{K}_0 \xrightarrow{R(\cdot)} \mathbb{K}_1 \xrightarrow{R(\cdot)} \cdots \xrightarrow{R(\cdot)} \mathbb{K}_r.$$

Thus, $(\mathbf{k}_0, \mathbf{k}_1, \cdots \mathbf{k}_r)$ *is an* r*-round division trail if* $\mathbf{k}_i$ *can propagate to* $\mathbf{k}_{i+1}$ *for all* $0 \leq i \leq r-1$.

Rule to determine the existence of Zero-sum:

Proposition

Assume $\mathbb{X}$ *is a multiset with division property* $\mathcal{D}^{n,m}_{\mathbf{k}}$*, then* $\mathbb{X}$ *does not have zero-sum property if and only if* $\mathbf{k}$ *contains all the* n *unit vectors.*

MILP - Aided Bit-based Division Property

It follows that we *only need to detect whether* $\mathbf{k}_r$ *contains all unit vectors*:

$\Rightarrow$ by previous Prop., the existence of any vector $\mathbf{v}$ s.t. $wt(\mathbf{v}) \geq 2$ implies that the state satisfies zero-sum property;

$\Rightarrow$ *if* $\mathbf{k}_{r+1}$ *contains all unit vectors*, the division property propagation should stop and *an* r*-round distinguisher can be derived*.

Denote $\mathbf{k}_0 \equiv (k_0^0, \cdots, k_{n-1}^0) \to \cdots \to \mathbf{k}_r \equiv (k_0^r, \cdots, k_{n-1}^r)$ an r-round bit-based division trail.

- The objective function is

$$Min: \; k_0^r + k_1^r + \cdots + k_{n-1}^r$$

$\Rightarrow$ we need *linear inequalities that describe all operations (XOR, S-Box, MC, Copy, ...)*

MILP - Aided Bit-based Division Property

It follows that we *only need to detect whether* $\mathbf{k}_r$ *contains all unit vectors*:

$\Rightarrow$ by previous Prop., the existence of any vector $\mathbf{v}$ s.t. $wt(\mathbf{v}) \geq 2$ implies that the state satisfies zero-sum property;

$\Rightarrow$ *if* $\mathbf{k}_{r+1}$ *contains all unit vectors*, the division property propagation should stop and *an r-round distinguisher can be derived*.

Denote $\mathbf{k}_0 \equiv (k_0^0, \cdots, k_{n-1}^0) \rightarrow \cdots \rightarrow \mathbf{k}_r \equiv (k_0^r, \cdots, k_{n-1}^r)$ an r-round bit-based division trail.

- The objective function is

$$Min: \ k_0^r + k_1^r + \cdots + k_{n-1}^r$$

$\Rightarrow$ we need *linear inequalities that describe all operations (XOR, S-Box, MC, Copy, ...)*

Model S-Box

Given the PRESENT S-Box:

$$(a_{n-1}, \dots, a_1, a_0) \xrightarrow{\text{S-Box}} (b_{n-1}, \dots, b_1, b_0).$$

it can be described by *8 linear inequalities (which is 3 less w.r.t. [XZB+16])*

$$\begin{cases} -a_2 - a_1 + b_3 + b_1 + b_0 \geq -1 \\ -3a_3 - 3a_2 - 3a_1 + b_3 + 2b_2 + b_1 + 2b_0 \geq -5 \\ -2a_3 - a_2 - a_1 - 2a_0 + 5b_3 + 5b_2 + 5b_1 + 2b_0 \geq 0 \\ -a_0 - b_3 - b_2 + 2b_1 - b_0 \geq -2 \\ a_3 + a_2 + a_1 + a_0 - 2b_3 - 2b_2 + b_1 - 2b_0 \geq -1 \\ -a_0 + 2b_3 - b_2 - b_1 - b_0 \geq -2 \\ -a_0 - 2b_3 + b_2 - 2b_1 + b_0 \geq -3 \\ a_3 + a_2 + a_1 + 2a_0 - 2b_2 - 2b_1 - 2b_0 \geq -1 \end{cases}$$

How to Decrease the Algebraic Degree?

PRESENT S-Box:

$$(x_0, x_1, x_2, x_3) \mapsto \text{S-Box}(x_0, x_1, x_2, x_3) = (y_0, y_1, y_2, y_3)$$

where

$$\begin{aligned}
y_3 &= 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1 x_2 \oplus x_0 \cdot (x_1 x_2 \oplus x_1 x_3 \oplus x_2 x_3)\\
y_2 &= 1 \oplus x_2 \oplus x_3 \oplus x_0 x_1 \oplus x_0 x_3 \oplus x_1 x_3 \oplus x_0 \cdot (x_1 x_3 \oplus x_2 x_3)\\
y_1 &= x_1 \oplus x_3 \oplus x_1 x_3 \oplus x_2 x_3 \oplus x_0 \cdot (x_1 x_2 \oplus x_1 x_3 \oplus x_2 x_3)\\
y_0 &= x_0 \oplus x_2 \oplus x_3 \oplus x_1 x_2
\end{aligned}$$

When x_0 is fixed as constant, the degree decreases from 3 to 2.

Number of Rounds of Zero-Sums by the MILP Division Property Tool

Dimension (in bit) of the space X and Y s.t.

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} Y \qquad X \xrightarrow{R^{r}(\cdot)} \text{zero-sum}$$

found by the MILP Division Property Tool (for PHOTON internal permutation) used to set up Zero-Sums

	P_{100}			P_{144}			P_{196}			P_{256}		
	Forward Direction											
#rounds	4	5	6	4	5	6	4	5	6	4	5	6
[Tod15]	12	20	72	12	24	84	12	24	84	12	28	92
Ours	**11**	20	72	**11**	**23**	84	**11**	24	84	**11**	**27**	92
	Backward Direction											
#rounds	3	4	5	3	4	5	3	4	5	3	4	5
Ours	**11**	**19***	**71***	**11**	**23**	**83***	**11**	**23***	**83***	**11**	**27**	**91***

* Partial balanced

Results from MILP automatic tools - Example

Given

$$\mathbb{B} \xleftarrow{R^{-s}} \begin{pmatrix} A & C & C & C & C & C \\ A & C & C & C & C & C \\ A & C & C & C & C & C \\ A & C & C & C & C & C \\ A & C & C & C & C & C \\ aaac & C & C & C & C & C \end{pmatrix}, \begin{pmatrix} A & C & C & C & C & C \\ C & A & C & C & C & C \\ C & C & A & C & C & C \\ C & C & C & A & C & C \\ C & C & C & C & A & C \\ C & C & C & C & C & aaac \end{pmatrix} \xrightarrow{R^r} \mathbb{B}$$

where $\mathbb{B}$ denotes (full/partial) balanced/zero-sum, then

$$\mathbb{B} \xleftarrow{R^{-s}} \begin{pmatrix} A & C & C & C & C & C \\ A & A & C & C & C & C \\ A & C & A & C & C & C \\ A & C & C & A & C & C \\ A & C & C & C & A & C \\ aaac & C & C & C & C & aaac \end{pmatrix} \xrightarrow{R^r} \mathbb{B}$$

Part IV

1-Round Extension: Subspace Trail Cryptanalysis

Observation on X and Y

Goal: using MILP automatic tools based on division property, find subspaces X and Y such that

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} Y \qquad X \xrightarrow{R^{r}(\cdot)} \text{zero-sum}$$

Note: such MILP tools can only found "zero-sum" for which the nibbles - of the input set $X \oplus Y$ - can be active/partial active or constant.

Other more generic cases are not considered, including the ones for which some particular (linear) relationships between the nibble hold.

Idea: use "subspace trail" to extend - for free - the results found by the MILP tools!

Observation on X and Y

Goal: using MILP automatic tools based on division property, find subspaces X and Y such that

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} Y \qquad X \xrightarrow{R^{r}(\cdot)} \text{zero-sum}$$

Note: such MILP tools can only found "zero-sum" for which the nibbles - of the input set $X \oplus Y$ - can be active/partial active or constant.

Other more generic cases are not considered, including the ones for which some particular (linear) relationships between the nibble hold.

Idea: use "subspace trail" to extend - for free - the results found by the MILP tools!

The Space Y - Backward Direction

The space Y found using MILP automatic tools corresponds to a "**column space**" in subspace trail notation [GRR16]

$$\mathcal{C}_i := \langle e_{0,i}, e_{1,i}, ..., e_{n-1,i} \rangle$$

E.g. if $n = 6$ and $i = 0$:

$$C_0 \equiv \begin{pmatrix} x_0 & 0 & 0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 & 0 & 0 \\ x_5 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

General case: given $I \subseteq \{0, 1, ..., n-1\}$ let

$$\mathcal{C}_I := \bigoplus_{i \in I} \mathcal{C}_i$$

The Space Y - Backward Direction

The space Y found using MILP automatic tools corresponds to a "**column space**" in subspace trail notation [GRR16]

$$\mathcal{C}_i := \langle e_{0,i}, e_{1,i}, ..., e_{n-1,i} \rangle$$

E.g. if $n = 6$ and $i = 0$:

$$C_0 \equiv \begin{pmatrix} x_0 & 0 & 0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 & 0 & 0 \\ x_5 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

General case: given $I \subseteq \{0, 1, ..., n-1\}$ let

$$\mathcal{C}_I := \bigoplus_{i \in I} \mathcal{C}_i$$

The Mixed Space $\mathcal{M}$ - Backward Direction

Let the "**mixed space**" defined as

$$\mathcal{M}_i := MC \circ SR(\mathcal{C}_i)$$

E.g. if $n = 6$ and $i = 0$:

$$M_0 \equiv \begin{pmatrix} x_0 & 2x_1 & 8x_2 & 5x_3 & 8x_4 & 2x_5 \\ 2x_0 & 12x_1 & 6x_2 & 2x_3 & x_4 & 5x_5 \\ 12x_0 & 13x_1 & 8x_2 & 8x_3 & 15x_4 & 9x_5 \\ 13x_0 & x_1 & 10x_2 & 3x_3 & 11x_4 & 5x_5 \\ x_0 & 8x_1 & 11x_2 & 14x_3 & 13x_4 & 15x_5 \\ 8x_0 & 8x_1 & 2x_2 & 3x_3 & 3x_4 & 2x_5 \end{pmatrix}$$

As before, given $I \subseteq \{0, 1, ..., n-1\}$ let

$$M_I := \bigoplus_{i \in I} M_i$$

The Mixed Space $\mathcal{M}$ - Backward Direction

Let the "**mixed space**" defined as

$$\mathcal{M}_i := MC \circ SR(\mathcal{C}_i)$$

E.g. if $n = 6$ and $i = 0$:

$$M_0 \equiv \begin{pmatrix} x_0 & 2x_1 & 8x_2 & 5x_3 & 8x_4 & 2x_5 \\ 2x_0 & 12x_1 & 6x_2 & 2x_3 & x_4 & 5x_5 \\ 12x_0 & 13x_1 & 8x_2 & 8x_3 & 15x_4 & 9x_5 \\ 13x_0 & x_1 & 10x_2 & 3x_3 & 11x_4 & 5x_5 \\ x_0 & 8x_1 & 11x_2 & 14x_3 & 13x_4 & 15x_5 \\ 8x_0 & 8x_1 & 2x_2 & 3x_3 & 3x_4 & 2x_5 \end{pmatrix}$$

As before, given $I \subseteq \{0, 1, ..., n-1\}$ let

$$M_I := \bigoplus_{i \in I} M_i$$

Subspace Trail Results

For each $a \in \mathcal{C}_I^\perp$, there exists $b \in \mathcal{M}_I^\perp$ such that

$$R(\mathcal{C}_I \oplus a) = \mathcal{M}_I \oplus b.$$

If

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} \mathcal{C}_I \oplus a$$

then

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} \mathcal{C}_I \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{M}_I \oplus b$$

Subspace Trail Results

For each $a \in \mathcal{C}_I^\perp$, there exists $b \in \mathcal{M}_I^\perp$ such that

$$R(\mathcal{C}_I \oplus a) = \mathcal{M}_I \oplus b.$$

If

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} \mathcal{C}_I \oplus a$$

then

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} \mathcal{C}_I \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{M}_I \oplus b$$

Subspace Trail Results - Add 1 round in the middle

MILP Tool: Given $Y \equiv \mathcal{C}_I \oplus a$ and X such that

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} \mathcal{C}_I \oplus a \qquad X \xrightarrow{R^{r}(\cdot)} \text{zero-sum}$$

then

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} (\mathcal{C}_I \oplus X) \oplus a \xrightarrow{R^{r}(\cdot)} \text{zero-sum}.$$

Subspace Trails: Since $\mathcal{C}_I \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{M}_I \oplus a'$, it follows that

$$\text{zero-sum} \xleftarrow{R^{-(s+1)}(\cdot)} (\mathcal{M}_I \oplus X) \oplus a' \xrightarrow{R^{r}(\cdot)} \text{zero-sum}.$$

Subspace Trail Results - Add 1 round in the middle

MILP Tool: Given $Y \equiv \mathcal{C}_I \oplus a$ and X such that

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} \mathcal{C}_I \oplus a \qquad X \xrightarrow{R^{r}(\cdot)} \text{zero-sum}$$

then

$$\text{zero-sum} \xleftarrow{R^{-s}(\cdot)} (\mathcal{C}_I \oplus X) \oplus a \xrightarrow{R^{r}(\cdot)} \text{zero-sum}.$$

Subspace Trails: Since $\mathcal{C}_I \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{M}_I \oplus a'$, it follows that

$$\text{zero-sum} \xleftarrow{R^{-(s+1)}(\cdot)} (\mathcal{M}_I \oplus X) \oplus a' \xrightarrow{R^{r}(\cdot)} \text{zero-sum}.$$

Our Results

Variants PHOTON	**Security Claim**	**# Rounds**	**Cost Size N**	**Property**
-80/20/16	80	10	2^{40}	Balance
		11	2^{76}	Balance
-128/16/16	128	10	2^{47}	Balance
		11	2^{107}	Balance
		12	$\mathbf{2^{127}}$	**PBalance**
-160/36/36	160	10	2^{48}	Balance
		11	2^{108}	Balance
		12	$\mathbf{2^{159}}$	**PBalance**
-224/32/32	224	10	2^{55}	Balance
		11	2^{124}	Balance
		12	$\mathbf{2^{184}}$	**Balance**

where "PBalance" $\equiv$ Partial Balance (Input or/and Output bits)

(Similar results are given in the paper for less than 10 rounds)

Part V

Final Remarks

Final Remarks

Several Zero-Sums results in the literature, most prominently on Keccak [AM09,BC10]:

- *it seems hard to exploit zero-sum distinguishers to set up an attack on an hash function; however, the inner permutation of a sponge construction must look like a random permutation!*

  Note: Keccak team [BDP+] decided to increase the number of rounds of Keccak (from 18 to 24) in round 2 of the SHA-3 competition to prevent this distinguisher!

Final Remarks

- *zero-sum distinguishers are meaningful since they can not be set up for any arbitrary number of rounds*:

  zero-sum distinguishers proposed in this paper don't work if the number of rounds of PHOTON are increased from 12 to (e.g.) 16.

- there is currently no reason to believe that the security of PHOTON as a hash function is endangered.

Final Remarks

- *zero-sum distinguishers are meaningful since they can not be set up for any arbitrary number of rounds*:

  zero-sum distinguishers proposed in this paper don't work if the number of rounds of PHOTON are increased from 12 to (e.g.) 16.

- there is currently no reason to believe that the security of PHOTON as a hash function is endangered.

Thanks for your attention!

Questions?

Comments?

References I

- J.-P. Aumasson and W. Meier,
  Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi
  CHES 2009

- G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche,
  Note on zero-sum distinguishers of Keccak-f
  `http://keccak.noekeon.org/NoteZeroSum.pdf`

- C. Boura and A. Canteaut,
  A zero-sum property for the KECCAK-f permutation with 18 rounds
  IEEE 2010

References II

L. Grassi, C. Rechberger and S. Rønjom,
Subspace Trail Cryptanalysis and its Applications to AES
IACR Transactions on Symmetric Cryptology 2016

J. Guo, T. Peyrin, and A. Poschmann,
The PHOTON Family of Lightweight Hash Functions
CRYPTO 2011

L. R. Knudsen and V. Rijmen,
Known-Key Distinguishers for Some Block Ciphers
ASIACRYPT 2007

References III

L. Sun, W. Wang, and M. Wang,
MILP-Aided Bit-Based Division Property for Primitives with Non-Bit-Permutation Linear Layers
https://eprint.iacr.org/2016/811.pdf

Y. Todo,
Structural Evaluation by Generalized Integral Property
EUROCRYPT 2015

Y. Todo and M. Morii,
Bit-Based Division Property and Application to Simon Family
FSE 2016

References IV

Z. Xiang, W. Zhang, Z. Bao and D. Lin,
Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers
ASIACRYPT 2016

RSAConference2018

San Francisco | April 16 – 20 | Moscone Center

NOW MATTERS

SESSION ID: CRYP-R04

IMPROVED SECURITY BOUND OF LIGHTMAC_PLUS AND ITS SINGLE-KEY VARIANT

Yusuke Naito

Head Researcher
Mitsubishi Electric Corporation

Improved Security Bound of LightMAC_Plus

Result 1

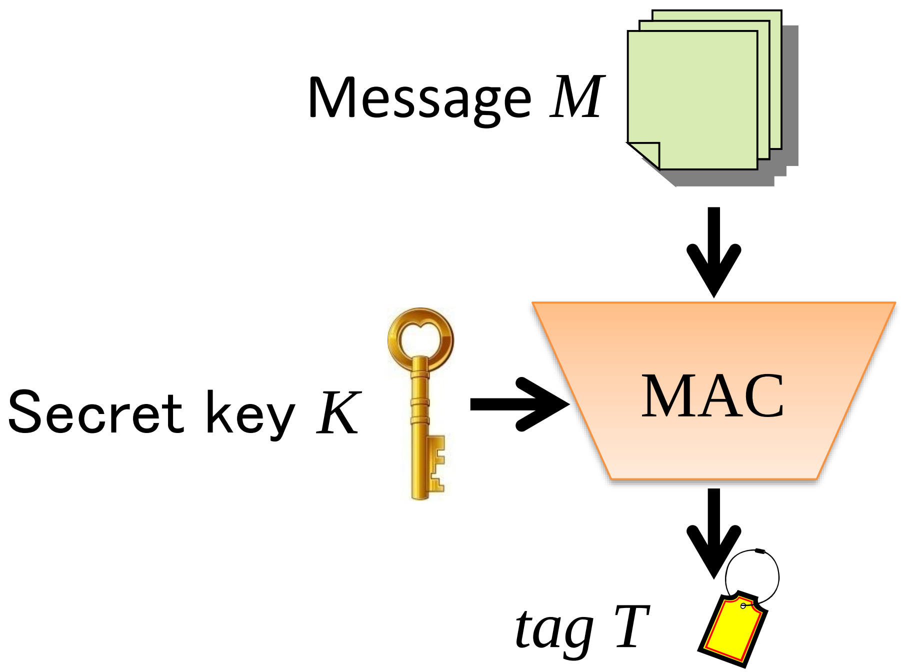
and Its Single-Key Variant

Result 2

Yusuke Naito

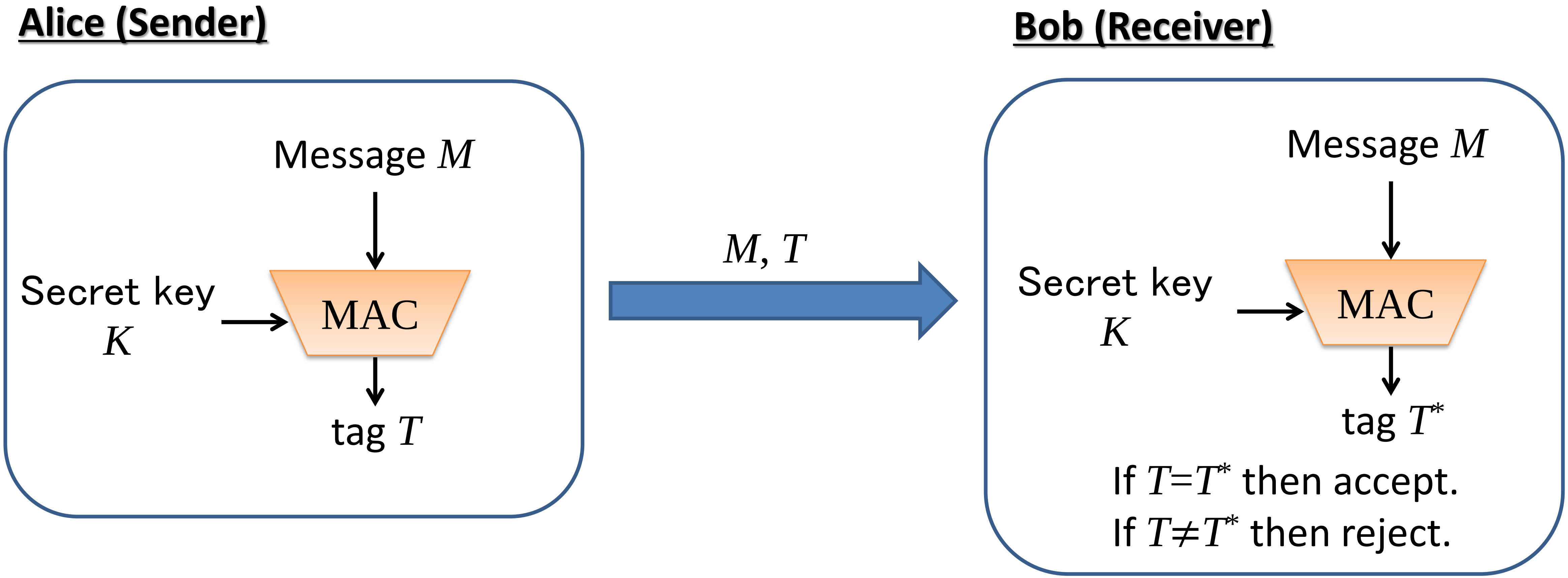
Mitsubishi Electric Corporation

Message Authentication Code (MAC)

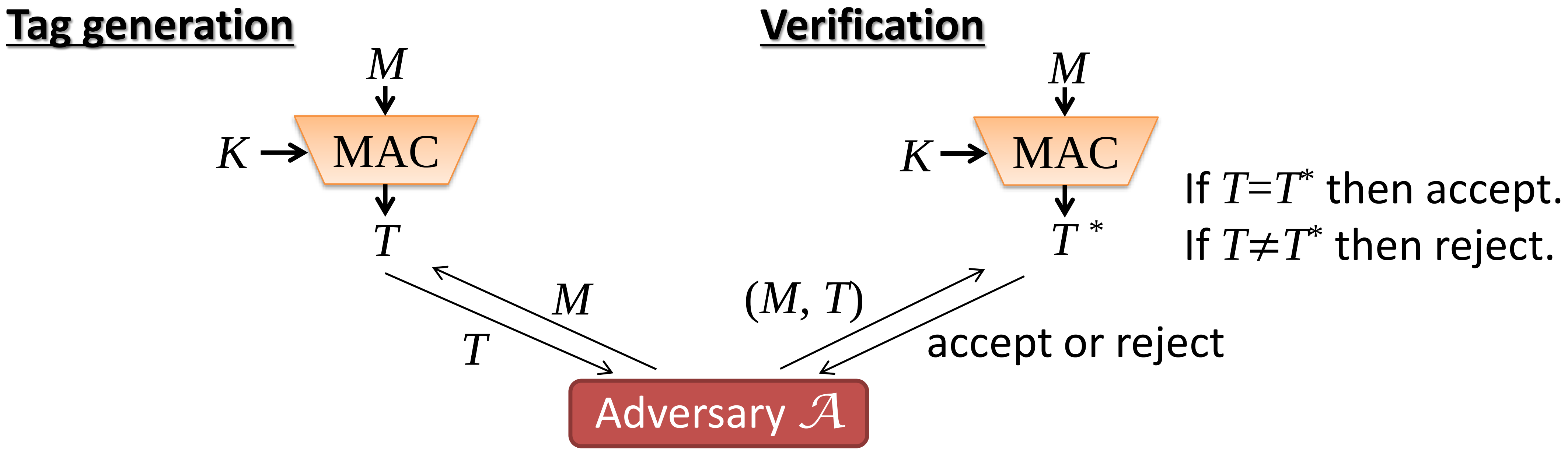


- Symmetric-key primitive.
- Used for integrity check.
- Input: a secret key and a variable-length message.
- Output: a fixed-length value, called tag.

Integrity Check Using MAC



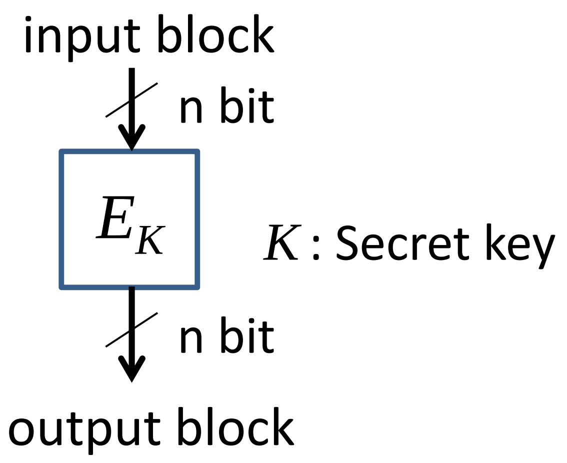
MAC Security (Unforgeability under CPA)



- Adversary $\mathcal{A}$ has access to
  - the tag generation algorithm (tagging queries)
  - the verification algorithm (verification queries).
- The goal of $\mathcal{A}$ is to forge a message and tag, i.e., make a (non-trivial) verification query s.t. accept is returned.
- Designing a MAC, Adv($\mathcal{A}$) = Pr[$\mathcal{A}$ forges] is evaluated.

MAC Design

- Underlying Primitives
  - Blockcipher, Tweakable Blockcipher, Permutation, ...
- Blockcipher
  - Standard: AES, Camellia, CLEFIA, PRESENT, ...
  - Family of permutations indexed by a key.
  - Security: Strong Pseudorandom Permutation
    (Ind. between E_K and a random permutation).
- Security proof of a blockcipher-based MAC
  - E_K is replaced with a random permutation.



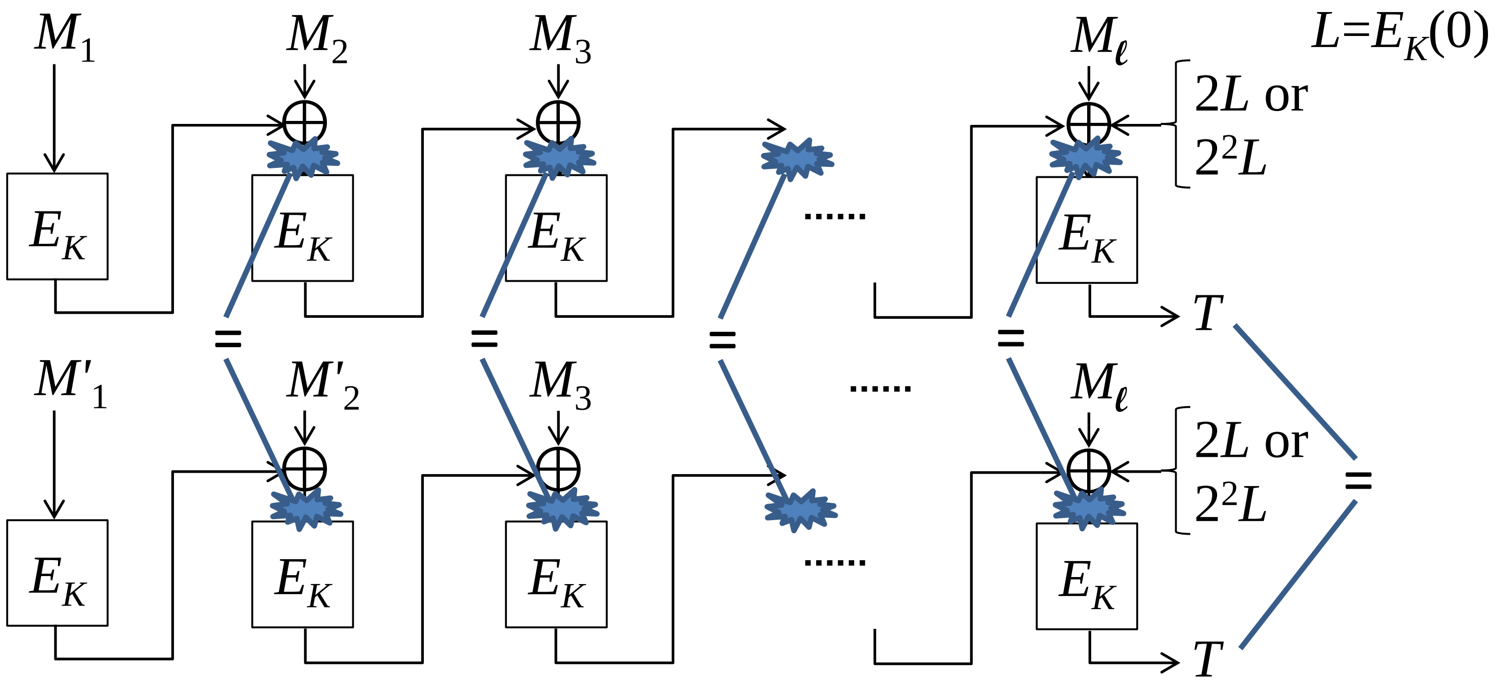
Birthday-Bound Secure MACs

- Many blockcipher-based MACs have been designed to have birthday-bound security.
  - Birthday-bound: $\mathrm{Adv}(\mathcal{A}) \leq O((\ell q)^2/2^n)$ (security up to $q=O(2^{n/2}/\ell)$).
    - ℓ: message length in (n-bit) blocks, i.e., # of blockcipher calls by a query.
    - q: # of (tagging or verification) queries.
  - Birthday-bound secure MACs: CMAC, PMAC, CBC-MAC (with prefix-free messages), ...
- Birthday-bound $O((\ell q)^2/2^n)$ security is not enough (e.g., Sweet32 at CCS 2016),
  - when large amounts of data are processed,
  - when a large number of connections need to be kept secure, or
  - when the block size n is small e.g., $n=64$.
- Designing a beyond-birthday-bound (BBB) secure MAC (i.e., having a better security bound) is an important topic.

How to Design a BBB-secure MAC

- The birthday bound $O((\ell q)^2/2^n)$ comes from collisions in E_K inputs/(n-bit) internal state.

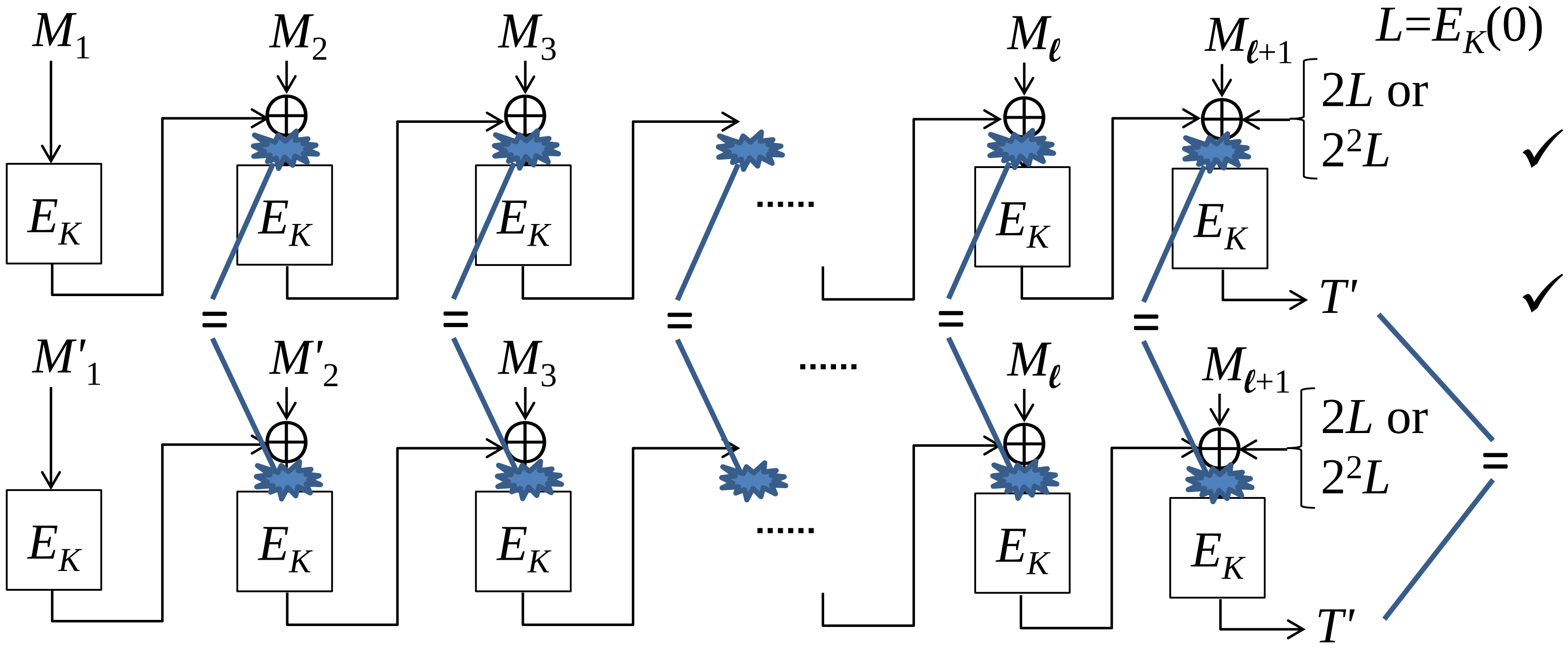
e.g., CMAC:



How to Design a BBB-secure MAC

- The birthday bound $O((\ell q)^2/2^n)$ comes from collisions in E_K inputs/(n-bit) internal state.

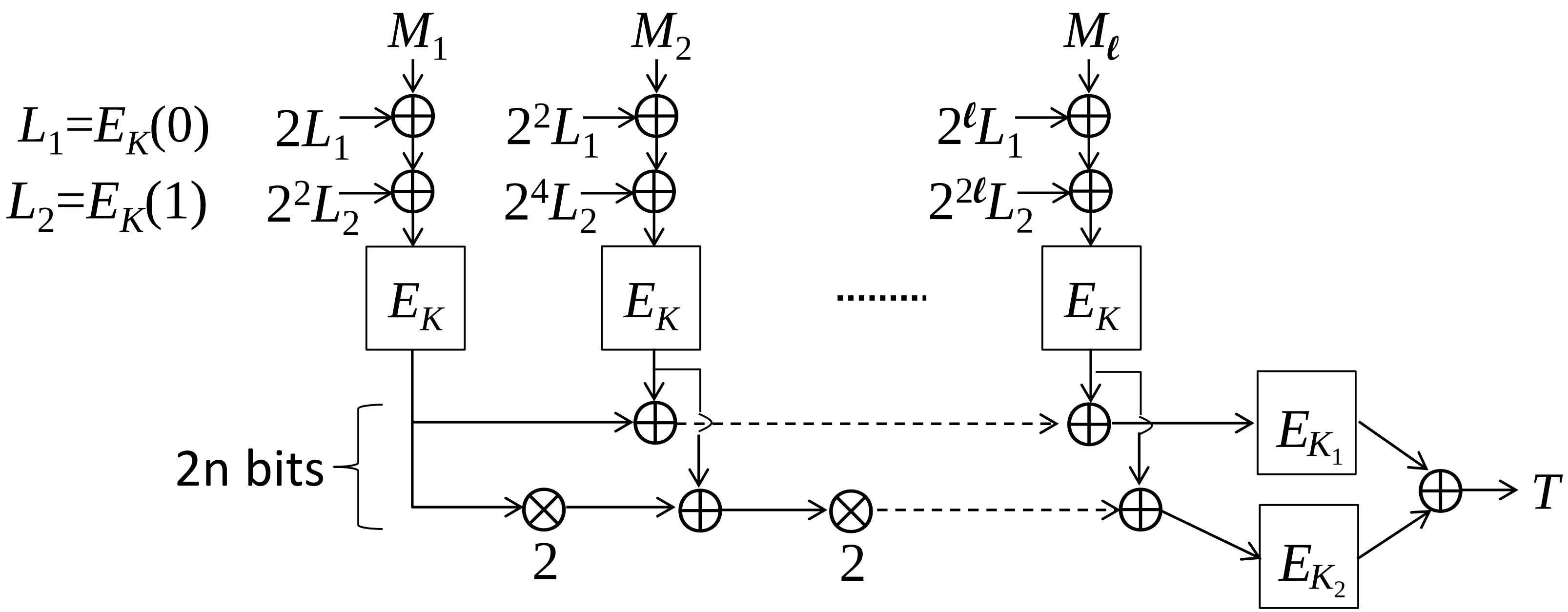
e.g., CMAC:



- ✓ An input collision for E_K triggers a forgery.
- ✓ Since there are ℓq inputs, by the birthday analysis, Collision Prob. = $O((\ell q)^2/2^n)$.

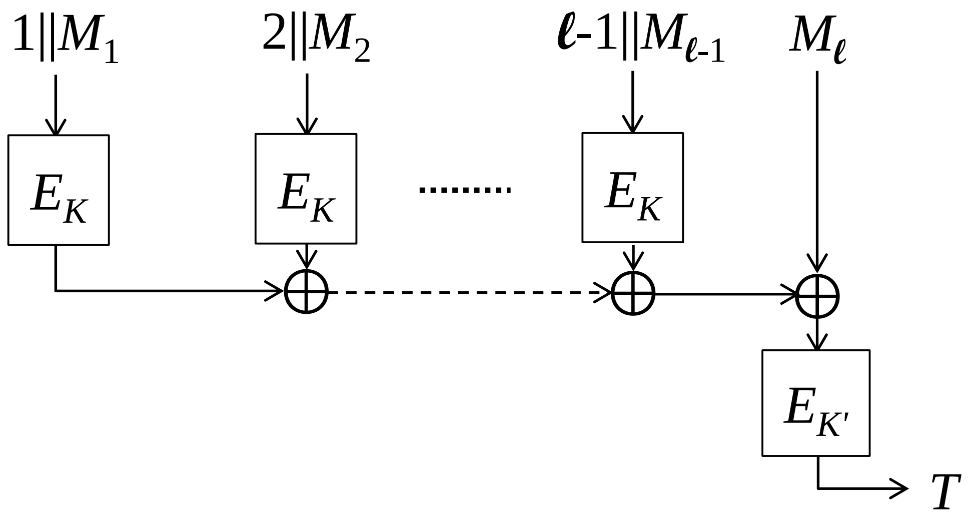
- In order to achieve BBB-security, we need to design a MAC so that the influences of collisions in E_K inputs / internal state are weakened.
- Existing BBB-secure MACs:
  e.g., PMAC_Plus, LightMAC, LightMAC_Plus, LightMAC_Plus2.

PMAC_Plus (Yasuda, CRYPTO 2011)



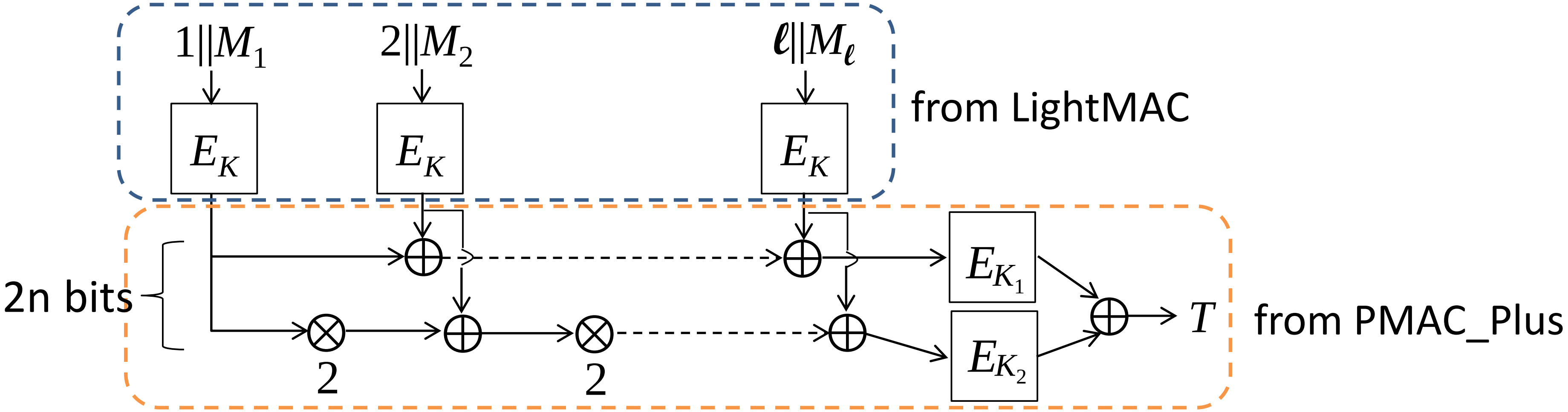
- In order to weaken the collision influences, PMAC_Plus employs
  - double secret masks
    -> the influence of a collision in E_K inputs is weakened,
  - double length (2n bit) internal state
    -> the collision prob. on the internal state is improved.
- Adv($\mathcal{A}$) $\leq O((\ell q)^3/2^{2n})$ (security up to $q=O(2^{2n/3}/\ell)$).
- Security level: $2^{n/2}/\ell$ -> $2^{2n/3}/\ell$.

LightMAC (Luykx et al., FSE 2016)



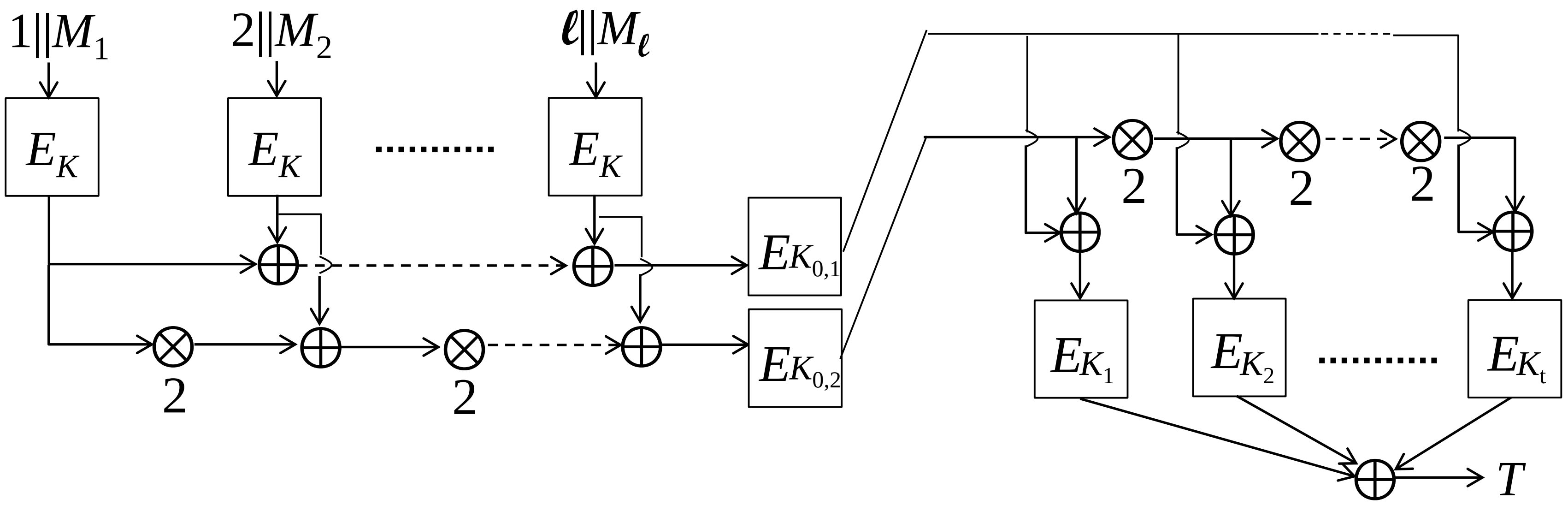
- LightMAC uses the counter-based construction:
  -> the input collision can be avoided for any ℓ,
  -> the message length ℓ can be removed from the security bound.
- By the birthday analysis for the n-bit internal state,
  Adv($\mathcal{A}$) $\leq O(q^2/2^n)$ (security up to $q=O(2^{n/2})$).
- Security level: $2^{n/2}/\ell$ -> $2^{n/2}$.

LightMAC_Plus (Naito, Asiacrypt 2017)



- Combination of LightMAC and PMAC_Plus
- From the LightMAC structure, the message length ℓ can be removed.
- From the PMAC_Plus structure, the collision prob. on the internal state is improved.
- Adv($\mathcal{A}$) $\leq O(q^3/2^{2n})$ (security up to $q=O(2^{2n/3})$).
- Security level: $2^{n/2}/\ell$ -> $2^{2n/3}/\ell$, $2^{n/2}$ -> $2^{2n/3}$.

LightMAC_Plus2 (Naito, Asiacrypt 2017)



- Has the better security bound than LightMAC_Plus.
- The finalization function is modified.
- Adv($\mathcal{A}$) $\leq O(q^{t+1}/2^{tn})$ for $t \leqq 7$ (security up to $q=O(2^{tn/(t+1)})$).
- Security level: $2^{n/2}/\ell$ -> $2^{2n/3}/\ell$, $2^{n/2}$ -> $2^{2n/3}$ -> $2^{tn/(t+1)}$.

Comparison and Question

Comparison

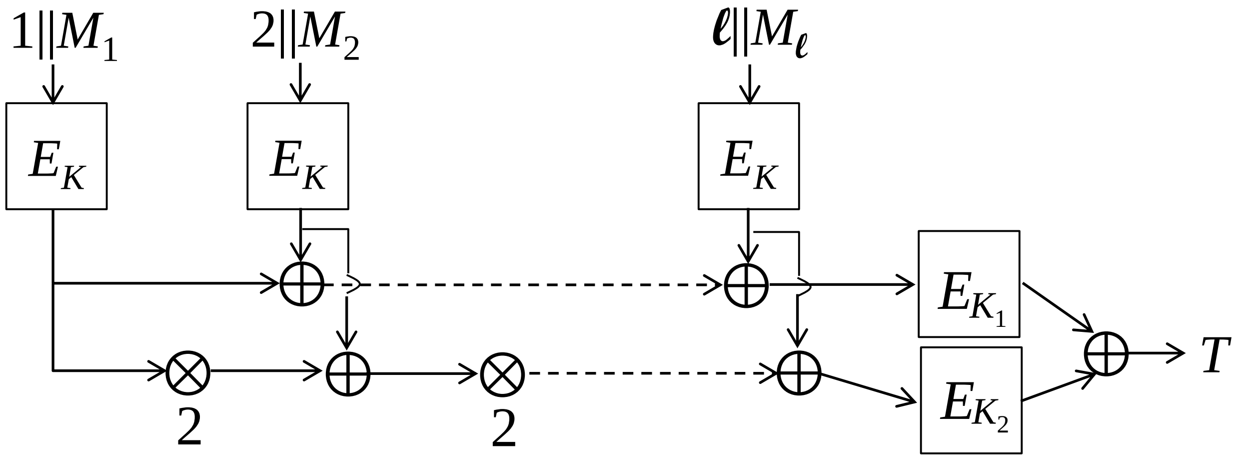
	Security Bound	Security Level	$\|M_i\|$	#BC in FF	Key Size
PMAC_Plus	$(\ell q)^3/2^{2n}$	$2^{2n/3}/\ell$	n	2	3
LightMAC	$q^2/2^n$	$2^{n/2}$	$n - c$	1	2
LightMAC_Plus	$q^3/2^{2n}$	$2^{2n/3}$	$n - c$	2	3
LightMAC_Plus2	$q^{(t+1)}/2^{tn}+q^2/2^{2n}$ $(t \leqq 7)$	$2^{tn/(t+1)}$	$n - c$	$t+2$	$t+3$
	$q^4/2^{3n}+q^2/2^{2n}$ $(t=3)$	$2^{3n/4}$	$n - c$	5	6
	$q^5/2^{4n}+q^2/2^{2n}$ $(t=4)$	$2^{4n/5}$	$n - c$	6	7
	⋮	⋮	⋮	⋮	⋮

- LightMAC_Plus2: Increasing the security level (i.e., increasing t), the efficiency (in the finalization function) is degraded and the key size is increased.

Question

- Can we obtain a highly secure MAC without degrading the efficiency or increasing the key size.

Our Result 1: Improved Security Bound of LightMAC_Plus



- Security bound: $O(q^3/2^{2n})$ -> $O(q_t^2 q_v/2^{2n})$.
  - q_t : # of tagging queries
  - q_v : # of verification queries
  - $q = q_t + q_v$
- If $q_t << q_v$ (e.g., a sender does not send a message frequently) or $q_v << q_t$ (e.g., # of forgery attempts is limited by a system) then LightMAC_Plus becomes a highly secure MAC without degrading the efficiency or increasing the key size.
- e.g., $q_v \leq 2^{n/2}$ -> security up to $2^{3n/4}$ queries; $q_v \leq 2^{n/3}$ -> security up to $2^{5n/6}$ queries; etc.

Comparison

Comparison

	Security Bound	Security Level	$\lvert M_i \rvert$	#BC in FF	Key Size
PMAC_Plus	$(\ell q)^3/2^{2n}$	$2^{2n/3}/\ell$	n	2	3
LightMAC	$q^2/2^n$	$2^{n/2}$	n - c	1	2
LightMAC_Plus	$q^3/2^{2n}$ $q_t^2 q_v/2^{2n}$ (Result 1)	$2^{2n/3}$	n - c	2	3
LightMAC_Plus2	$q^{(t+1)}/2^{tn}+q^2/2^{2n}$ ($t \leqq 7$)	$2^{tn/(t+1)}$	n - c	t+2	t+3
	$q^4/2^{3n}+q^2/2^{2n}$ (t=3)	$2^{3n/4}$	1	5	6
	$q^5/2^{4n}+q^2/2^{2n}$ (t=4)	$2^{4n/5}$	1	6	7
	⋮	⋮	⋮	⋮	⋮

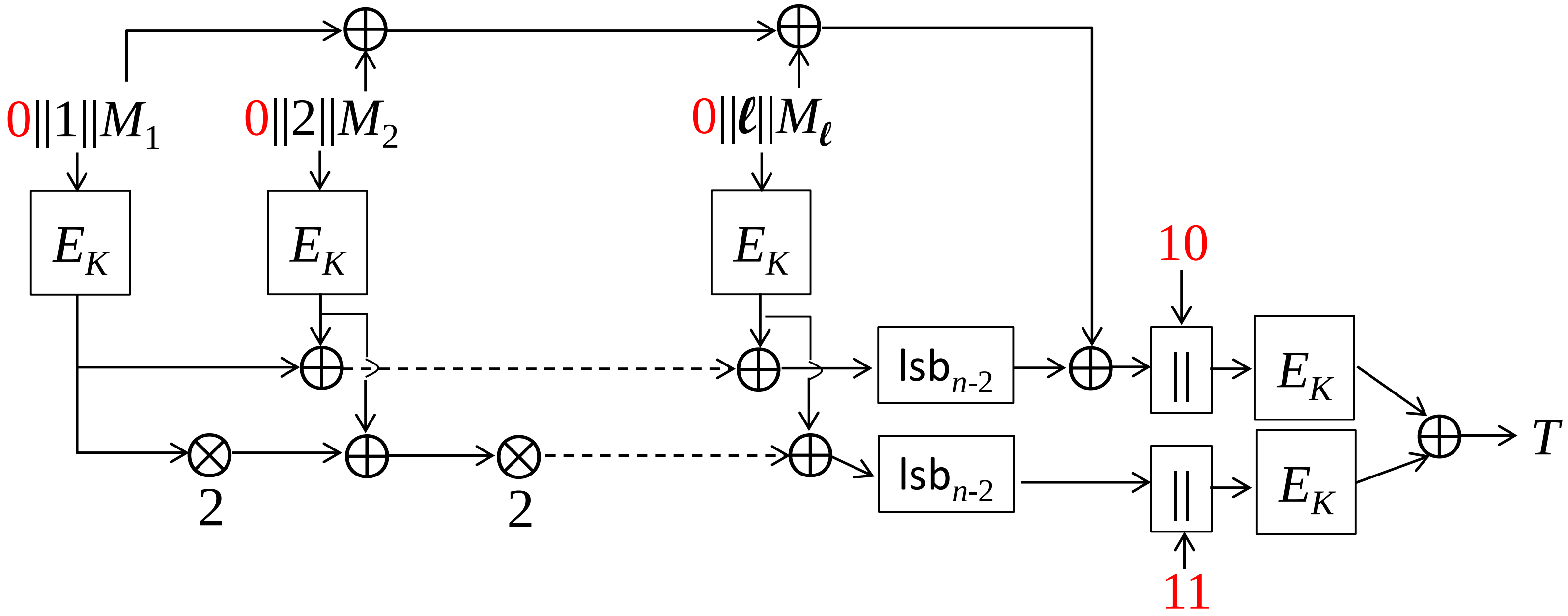
- LightMAC_Plus becomes a highly secure MAC without degrading the efficiency or increasing the key size if $q_t << q_v$ or $q_t >> q_v$ but uses 3 blockcipher keys.

Question

- Can we reduce the key size of LightMAC_Plus while keeping the BBB-security?

Our Result 2: Single-Key Variant of LightMAC_Plus

LightMAC_Plus1k



- In order to reduce the key size, the domain separation technique is used.
  - Hashing: 0
  - Finalization: 10, 11
- The keyed blockciphers with distinct inputs can be seen as distinct keyed blockciphers.
- In order to avoid a forgery with a collision in blockcipher outputs (due to 2-bit truncation), the ZMAC technique is used: XOR of message blocks are input to the internal state.
- Security bound: $O(q_t^2 q_v / 2^{2n})$.

Comparison

	Security Bound	Queries	$\lvert M_i \rvert$	#BC in FF	Key Size
PMAC_Plus	$\ell^3 q^3/2^{2n}$	$2^{2n/3}/\ell$	n	2	3
LightMAC	$q^2/2^n$	$2^{n/2}$	$n - c$	1	2
LightMAC_Plus	$q^3/2^{2n}$	$2^{2n/3}$	$n - c$	2	3
	$q_t^2 q_v/2^{2n}$ (Result 1)				
LightMAC_Plus1k	$q_t^2 q_v/2^{2n}$		$n - c$	2	1 (Result 2)
LightMAC_Plus2	$q^{(t+1)}/2^{tn}+q^2/2^{2n}$ ($t \leqq 7$)	$2^{tn/(t+1)}$	$n - c$	t+2	t+3
	$q^4/2^{3n}+q^2/2^{2n}$ (t=3)	$2^{3n/4}$	1	5	6
	$q^5/2^{4n}+q^2/2^{2n}$ (t=4)	$2^{4n/5}$	1	6	7
	⋮	⋮	⋮	⋮	⋮

Conclusion

for a greener tomorrow

- Result 1: Improved the security bound of LightMAC_Plus:
  - The security bound: $O(q^3/2^{2n})$ -> $O(q_t^2 q_v)/2^{2n}$.
  - If $q_t << q_v$ (e.g., a sender does not send a message frequently) or $q_v << q_t$ (e.g., # of forgery attempts is limited by a system) then LightMAC_Plus becomes a highly secure MAC without degrading the efficiency or increasing the key size.
- Result 2: Proposed LightMAC_Plus1k, the single key variant of LightMAC_Plus:
  - The key size: 3 -> 1.
  - The security bound: $O(q_t^2 q_v)/2^{2n}$.

Thank you for your attention!