Improved Slender-set Linear Cryptanalysis

Guo-Qiang Liu[1] Chen-Hui Jin[1] Chuan-Da Qi[2]

[1]Information Science Technology Institute Zhengzhou, Henan, China

[2]Xinyang Normal University Xinyang, Henan, China

FSE 2014

Outline

1. Introduction
   - Description of PRESENT-like Cipher
   - Previous Work
2. Our Contributions
   - Main Techniques
   - Experiments
3. Conclusion

Outline

1. Introduction
   - Description of PRESENT-like Cipher
   - Previous Work
2. Our Contributions
   - Main Techniques
   - Experiments
3. Conclusion

The Block Cipher Maya

- PRESENT is a lightweight SPN block cipher proposed at CHES 2007.
- Gomathisankaran *et al.* presented a PRESENT-like cipher with secret S-boxes which is named Maya.

The Block Cipher Maya

A typical example of the PRESENT-like cipher with secret S-boxes

- Block Size: 64 bit
- S-box: 16 secret and key-dependent 4-bit S-boxes
- P-box: Public or secret bit-wise permutation of 64-bit
- Round: 16 rounds

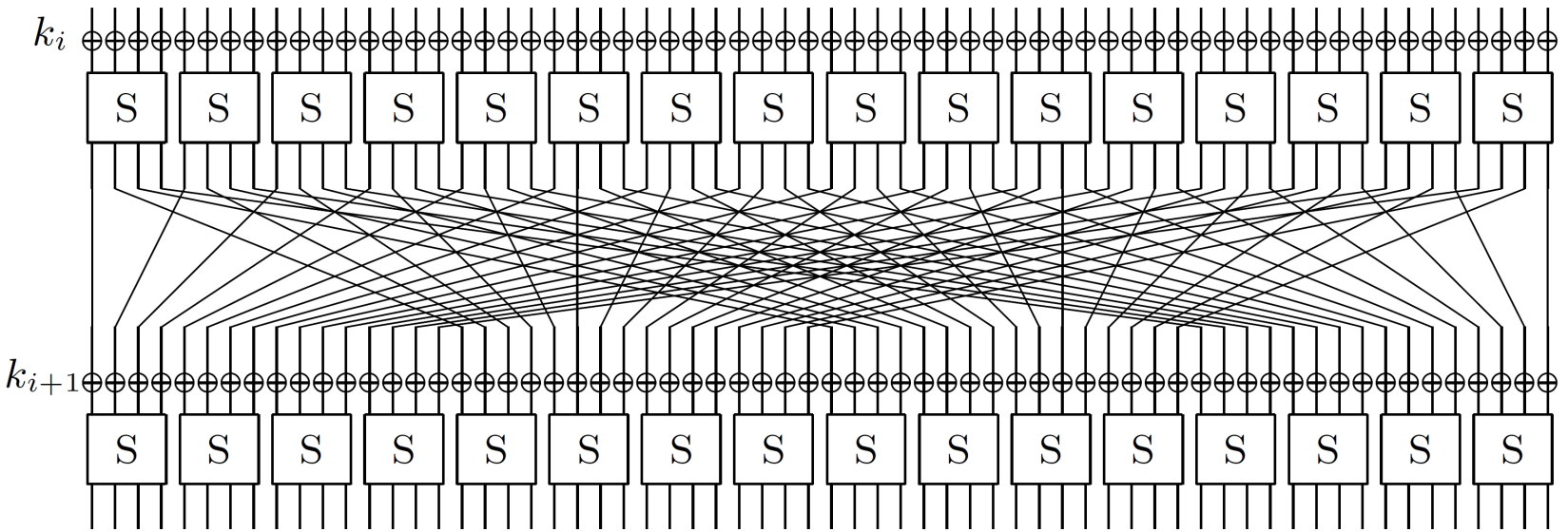

Figure: Two rounds PRESENT-like cipher

Outline

1. Introduction
   - Description of PRESENT-like Cipher
   - Previous Work
2. Our Contributions
   - Main Techniques
   - Experiments
3. Conclusion

Some Basic Notations

- The inner product on F_2^n is denoted by $\langle \cdot, \cdot \rangle$, that is

$$\langle (a_0, a_1, \cdots, a_{n-1}), (b_0, b_1, \cdots, b_{n-1}) \rangle = \sum_{i=0}^{n-1} a_i b_i$$

- The Walsh of H at the pair $(\alpha, \beta) \in F_2^n \times F_2^m$ is defined by

$$\hat{H}(\alpha, \beta) = \sum_{x \in F_2^n} (-1)^{\langle \beta, H(x) \rangle + \langle \alpha, x \rangle}$$

Slender-set Attack

- In 2013, Borghoff *et al.* introduced the slender-set differential and linear cryptanalysis on PRESENT-like ciphers with key-dependent secret S-boxes.
  [Journal of Cryptology 2013]

Borghoff's Work on Slender-set Linear Cryptanalysis

Recover the secret S-box by looking at Fourier transform for a group of output masks and every input value for a given S-box.

- Focus on the improvements of slender-set linear cryptanalysis.

Slender-set Attack

- In 2013, Borghoff *et al.* introduced the slender-set differential and linear cryptanalysis on PRESENT-like ciphers with key-dependent secret S-boxes.
  [Journal of Cryptology 2013]

Borghoff's Work on Slender-set Linear Cryptanalysis

Recover the secret S-box by looking at Fourier transform for a group of output masks and every input value for a given S-box.

- Focus on the improvements of slender-set linear cryptanalysis.

Description of Slender-set Linear Cryptanalysis

We denote that

$$F : F_2^4 \times F_2^{60} \rightarrow F_2^{64} \quad and \quad F(x, y) = c$$

where the function F is the encryption function that starts after the first layer of S-boxes

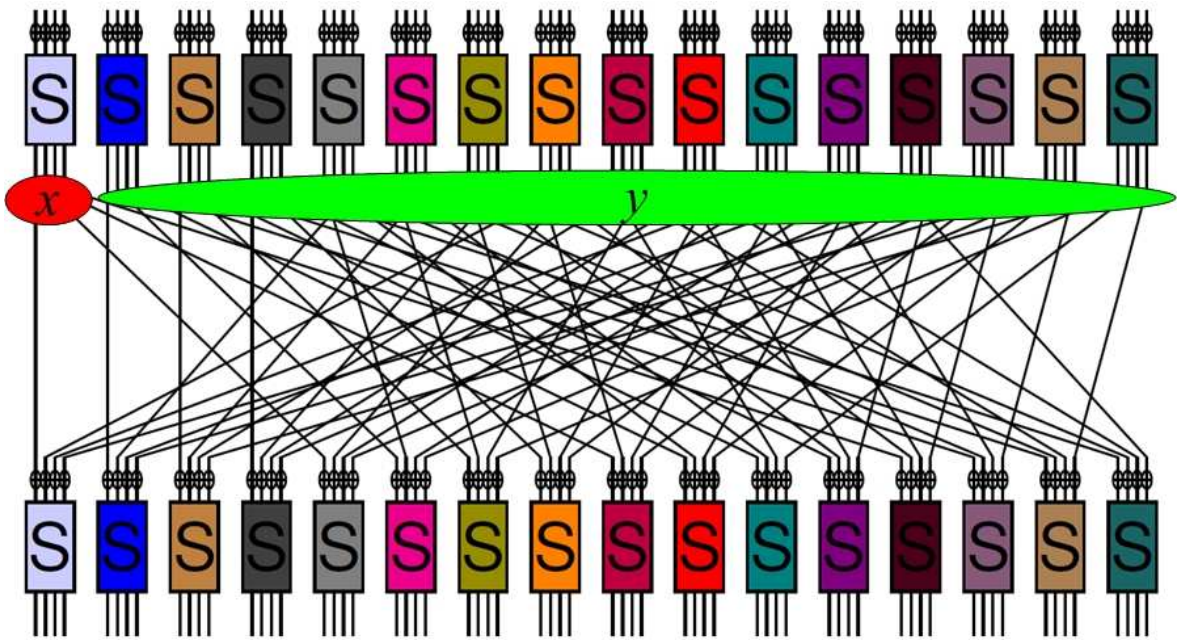

Figure: The function F

Description of Slender-set Linear Cryptanalysis

We denote the corresponding function by

$$T_x : F_2^{60} \to F_2^{64} \quad \textit{and} \quad T_x(y) = F(x,y)$$

and we look at

$$\hat{T}_x(0,\beta) = \sum_{y \in F_2^{60}} (-1)^{\langle \beta, T_x(y) \rangle} = \sum_{y \in F_2^{60}} (-1)^{\langle \beta, F(x,y) \rangle}$$

Lemma 1. [7]

With the notation from above, it holds that

$$2^4 \hat{T}_\lambda(0,\beta) = \sum_{\alpha_1 \in F_2^4} (-1)^{\langle \alpha_1, \lambda \rangle} \hat{F}((\alpha_1, 0), \beta)$$

Description of Slender-set Linear Cryptanalysis

Now we denote the whole encryption function by E.

$$E : F_2^4 \times F_2^{60} \rightarrow F_2^{64} \quad \textit{and} \quad E(x, y) = c$$

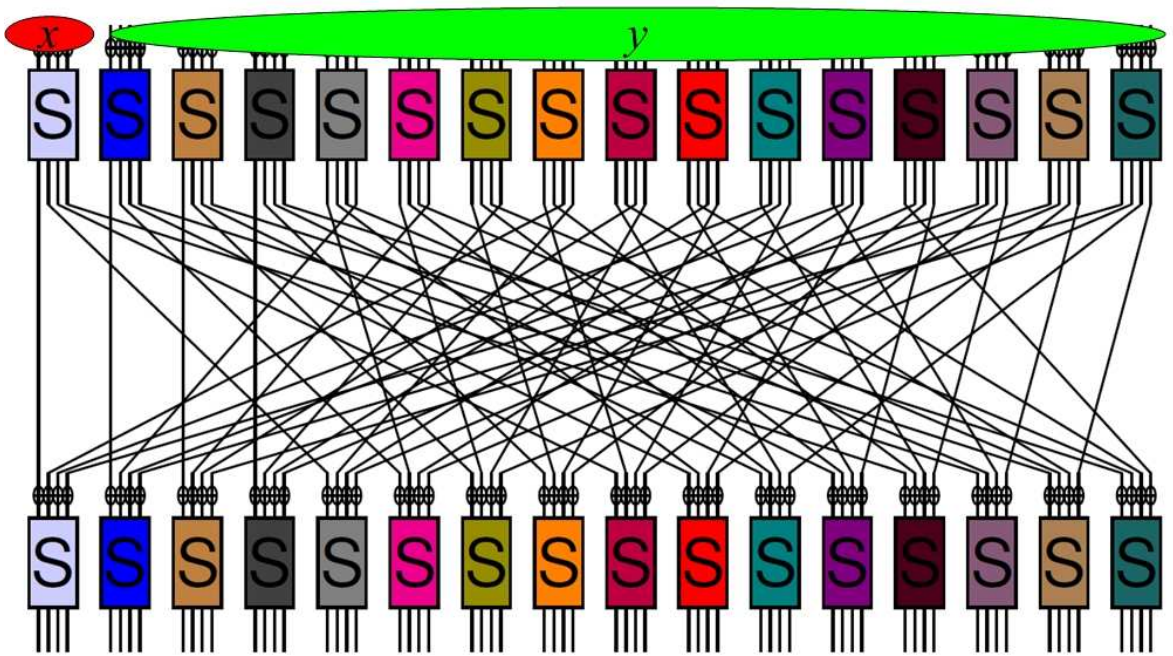

Figure: The function E

Description of Slender-set Linear Cryptanalysis

They define the function corresponding to fixing x as T'_x , that is

$$T'_x : F_2^{60} \to F_2^{64} \quad \textit{and} \quad T'_x(y) = E(x, y)$$

Lemma 2. [7]

With the notation from above, the bias of $\langle \beta, T'_x(y) \rangle$ is equal to the bias of $\langle \beta, T_{S(x)}(y) \rangle$. That is

$$\hat{T}'_x(0, \beta) = \hat{T}_{S(x)}(0, \beta)$$

Description of Slender-set Linear Cryptanalysis

- An important equation can be derived from Lemma 1 and Lemma 2.

An Important Equation in Borghoff's Aattack

$$\hat{T}'_x(0,\beta) = \hat{T}_{S(x)}(0,\beta) = 2^{-4} \sum_{\xi \in F_2^4} (-1)^{\langle \xi, S(x) \rangle} \hat{F}((\xi,0),\beta)$$
$$\approx 2^{-4}(-1)^{\langle \alpha, S(x) \rangle} \hat{F}((\alpha,0),\beta)$$

Explanation of This Equation

- For a given mask β , there is exactly one mask α such that $\hat{F}((\alpha,0),\beta)$ is higher while for any $\xi \neq \alpha$ the value $\hat{F}((\xi,0),\beta)$ is close to zero.
- As P is a m-bit permutation, the value of $\hat{F}((\alpha,0),\beta)$ is higher while for any $wt(\alpha) = 1$.

Description of Slender-set Linear Cryptanalysis

- An important equation can be derived from Lemma 1 and Lemma 2.

An Important Equation in Borghoff's Aattack

$$\hat{T}'_x(0,\beta) = \hat{T}_{S(x)}(0,\beta) = 2^{-4} \sum_{\xi \in F_2^4} (-1)^{\langle \xi, S(x) \rangle} \hat{F}((\xi,0),\beta)$$
$$\approx 2^{-4}(-1)^{\langle \alpha, S(x) \rangle} \hat{F}((\alpha,0),\beta)$$

Explanation of This Equation

- For a given mask β , there is exactly one mask α such that $\hat{F}((\alpha,0),\beta)$ is higher while for any $\xi \neq \alpha$ the value $\hat{F}((\xi,0),\beta)$ is close to zero.
- As P is a m-bit permutation, the value of $\hat{F}((\alpha,0),\beta)$ is higher while for any $wt(\alpha) = 1$.

Description of Slender-set Linear Cryptanalysis

By This Method

Borghoff *et al.* could partition the values of x into two equally-sized sets V_0 and V_1 depending on the sign of $\hat{T}'_x(0, \beta)$, where $V_\gamma = \{x | \langle \alpha, S(x) \rangle = \gamma\}, \gamma = 0, 1$.

The Steps of Borghoff's Attack

Step 1

Let the output mask $\beta = 0^{4j}||b||0^{60-4j}$, $0 \leq j \leq 15$. For every leftmost input $0 \leq x \leq 15$ and for every $1 \leq b \leq 15$, estimate the value of the counter $\hat{T}'_x(0, \beta)$.

Example of Step 1

Let the output mask $\beta = 0^{4j}||b||0^{60-4j}$, $0 \leq j \leq 15$. For different b, j and x, we estimate $\hat{T}'_x(0, \beta)$ as following.

- $\beta = 0x1$ {-554, -364, 170, -166, 352, -776, -686, -228, 222, -638, -774, -64, 44, -560, 530, 416}
  $\beta = 0x2$ {-810, 830, 1974, -654, 1584, 2286, 2118, -1328, -990, -1020, -334, 2270, 1880, -1182, -702, 2040}
  ...
  $\beta = 0xF000000000000000$ {-402, 28, -502, -542, -144, -408, 10, -136, 164, 76, 16, 712, 262, -246, 116, -158}

The Steps of Borghoff's Attack

Step 2

After $W_\beta = (\hat{T}'_0(0,\beta), \hat{T}'_1(0,\beta), \cdots, \hat{T}'_{15}(0,\beta))$ being retrieved,we identify the three longest vectors using the Euclidean norm as a metric, as Borghoff *et al*. assume that these vectors contain the most reliable information.

Step 3

We transform each of these vectors into a binary vector such that the eight highest counter values correspond to '1'-bits and the remaining correspond to '0'-bits. We take a majority vote among these three binary vectors to find a correct coordinate function of secret S-box.

The Steps of Borghoff's Attack

Step 2

After $W_\beta = (\hat{T}'_0(0,\beta), \hat{T}'_1(0,\beta), \cdots, \hat{T}'_{15}(0,\beta))$ being retrieved,we identify the three longest vectors using the Euclidean norm as a metric, as Borghoff *et al*. assume that these vectors contain the most reliable information.

Step 3

We transform each of these vectors into a binary vector such that the eight highest counter values correspond to '1'-bits and the remaining correspond to '0'-bits. We take a majority vote among these three binary vectors to find a correct coordinate function of secret S-box.

Example of Step 2 and Step 3

- The three longest vectors were these:

  $(-3138, -2218, -3156, 3146, -2486, 1784, -2974, -3452, 1392, 1602, 2850, 3198, -3100, 2796, -3458, 1708)$

  $(-2558, -1768, -2022, 2798, -1754, 2538, -1808, -2440, 2784, 2694, 2424, 3378, -2576, 2378, -2658, 2424)$

  $(3046, 1842, 1730, -2982, 1952, -1600, 2116, 2930, -2426, -2742, -2036, -2440, 2918, -1764, 3112, -1670)$

- After transforming these vectors into binary vectors as described, one gets

  $(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$

  $(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$

  $(1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0)$

Example of Step 2 and Step 3

- The three longest vectors were these:

$$(-3138, -2218, -3156, 3146, -2486, 1784, -2974, -3452, 1392, 1602, 2850, 3198, -3100, 2796, -3458, 1708)$$

$$(-2558, -1768, -2022, 2798, -1754, 2538, -1808, -2440, 2784, 2694, 2424, 3378, -2576, 2378, -2658, 2424)$$

$$(3046, 1842, 1730, -2982, 1952, -1600, 2116, 2930, -2426, -2742, -2036, -2440, 2918, -1764, 3112, -1670)$$

- After transforming these vectors into binary vectors as described, one gets

$$(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$$

$$(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$$

$$(1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0)$$

The Steps of Borghoff's Attack

We get the coordinate functions of secret S-boxes by majority vote among these three binary vectors as following:

$$(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$$

Step 4

We recover the 4-bit secret S-boxes based on four linearly independent coordinate functions of secret S-boxes.

The Steps of Borghoff's Attack

We get the coordinate functions of secret S-boxes by majority vote among these three binary vectors as following:

$$(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$$

Step 4

We recover the 4-bit secret S-boxes based on four linearly independent coordinate functions of secret S-boxes.

Outline

1. Introduction
   - Description of PRESENT-like Cipher
   - Previous Work
2. Our Contributions
   - Main Techniques
   - Experiments
3. Conclusion

The First Improvement

Question 1?

According to

$$\hat{T}'_x(0,\beta) \approx 2^{-4}(-1)^{\langle \alpha, S(x)\rangle}\hat{F}((\alpha,0),\beta)$$

The sign of $\hat{F}((\alpha,0),\beta)$ might be opposite for the same α and the different β. This will cause opposite sign of $\hat{T}'_x(0,\beta)$ for the same α and the different β.

The symbol '0' represents the same partition in one vector, and may stand for different partitions in different vectors.

$$(0,0,0,1,0,1,0,0,1,1,1,1,0,1,0,1)$$

$$(0,0,0,1,0,1,0,0,1,1,1,1,0,1,0,1)$$

$$(1,1,1,0,1,0,1,1,0,0,0,0,1,0,1,0)$$

The First Improvement

Question 1?

According to

$$\hat{T}'_x(0,\beta) \approx 2^{-4}(-1)^{\langle \alpha, S(x)\rangle}\hat{F}((\alpha,0),\beta)$$

The sign of $\hat{F}((\alpha,0),\beta)$ might be opposite for the same α and the different β. This will cause opposite sign of $\hat{T}'_x(0,\beta)$ for the same α and the different β.

The symbol '0' represents the same partition in one vector, and may stand for different partitions in different vectors.

$$(0,0,0,1,0,1,0,0,1,1,1,1,0,1,0,1)$$

$$(0,0,0,1,0,1,0,0,1,1,1,1,0,1,0,1)$$

$$(1,1,1,0,1,0,1,1,0,0,0,0,1,0,1,0)$$

Technique to Support Consistency of Partitions

Our Idea

We consider to partition x by the difference of $\hat{T}_i'(0, \beta)$ and $\hat{T}_k'(0, \beta)$, where $0 \leq i \leq 15$. Without loss of generality, we let $k = 0$ in our paper.

Technique to Support Consistency of Partitions

Relative Distance

We compute the *relative distance*

$$\mathbb{R}_{\beta}^{(i)} = \hat{T}_0'(0,\beta) - \hat{T}_i'(0,\beta)$$

between $\hat{T}_0'(0,\beta)$ and $\hat{T}_i'(0,\beta)$.

- We transform each of these *relative distance* vectors $(\mathbb{R}_{\beta}^{(0)}, \mathbb{R}_{\beta}^{(1)}, \cdots, \mathbb{R}_{\beta}^{(15)})$ into binary vectors $(\mathbb{B}_{\beta}^{(0)}, \mathbb{B}_{\beta}^{(1)}, \cdots, \mathbb{B}_{\beta}^{(15)})$ such that the eight highest values correspond to ”1”-bits and the remaining values correspond to ”0”-bits.

Example of Our First Improvement

we obtained the vectors described above. The *relative distance* vectors are these for the three longest vectors:

$$(0, 920, -18, 6284, 652, 4922, 164, -314, 4530, 4740, 5988, 6336, 38, 5934, -320, 4846)$$

$$(0, 790, 536, 5356, 804, 5096, 750, 118, 5342, 5252, 4982, 5936, -18, 4936, -100, 4982)$$

$$(0, -1204, -1316, -6028, -1094, -4646, -930, -116, -5472, -5788, -5082, -5486, -128, -4810, 66, -4716)$$

We transform the *relative distance* vector into binary vectors as follows:

$$(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$$

$$(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$$

$$(0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$$

Example of Our First Improvement

we obtained the vectors described above. The *relative distance* vectors are these for the three longest vectors:

$$(0, 920, -18, 6284, 652, 4922, 164, -314, 4530, 4740, 5988, 6336, 38, 5934, -320, 4846)$$

$$(0, 790, 536, 5356, 804, 5096, 750, 118, 5342, 5252, 4982, 5936, -18, 4936, -100, 4982)$$

$$(0, -1204, -1316, -6028, -1094, -4646, -930, -116, -5472, -5788, -5082, -5486, -128, -4810, 66, -4716)$$

We transform the *relative distance* vector into binary vectors as follows:

$$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1})$$

$$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1})$$

$$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1})$$

The Second Improvement

Question 2?

Borghoff *et al.* recovered the ”true” vectors by majority vote method based on three longest vectors, which will lose some information from the candidate binary vectors.

Our Idea

We consider to make full use of information from the 240 output low-weight masks $\beta = 0^{4j}||b||0^{60-4j}, 0 \leq j \leq 15, 1 \leq b \leq 15$ of the S-box layer in the last round instead of three longest vectors by a new voting method.

The Second Improvement

Question 2?

Borghoff *et al.* recovered the ”true” vectors by majority vote method based on three longest vectors, which will lose some information from the candidate binary vectors.

Our Idea

We consider to make full use of information from the 240 output low-weight masks $\beta = 0^{4j}||b||0^{60-4j}, 0 \leq j \leq 15, 1 \leq b \leq 15$ of the S-box layer in the last round instead of three longest vectors by a new voting method.

The Definition of Vectors Distances

How to measure the degree of similarity between two vectors?

Vectors Distances

We define the distances between two binary vectors B_i and B_j by

$$\mathbb{D}_{B_i,B_j} = wt(\overline{B_i \oplus B_j})$$

where $1 \leq i, j \leq 240$ and $wt(\overline{B_i \oplus B_j})$ is the *Hamming weight* of $\overline{B_i \oplus B_j}$.

The Definition of Vectors Distances

How to measure the degree of similarity between two vectors?

Vectors Distances

We define the distances between two binary vectors B_i and B_j by

$$\mathbb{D}_{B_i,B_j} = wt(\overline{B_i \oplus B_j})$$

where $1 \leq i, j \leq 240$ and $wt(\overline{B_i \oplus B_j})$ is the *Hamming weight* of $\overline{B_i \oplus B_j}$.

Which Vectors Can be Seem as Similar Vectors

Which Vectors is Similar

- For two binary vectors B_i and B_j, they will be similar to each other when $wt(\overline{B_i \oplus B_j})$ approximate to 16.
- The expectation of $wt(\overline{\alpha \oplus \beta})$ is equal to 8.533 for two random vectors.

- Thus we believe the vectors B_i and B_j would be closer to each other when $wt(\overline{B_i \oplus B_j}) > 8$.

The Definition of Similarity Degree

We assume that the binary vectors corresponding the same α are similar to each other. How to partition the 240 binary vectors?

Similarity Degree

The *similarity degree* of B_i and B_j is defined by

$$\mathbb{S}_{B_i,B_j} = g(B_i,B_j) + \sum_{\substack{1\le k\le 240\\ k\neq i,k\neq j}} (f(B_i,B_k)+f(B_j,B_k))$$

where the function $g(B_i,B_j) = \begin{cases} \xi, & \text{if } wt(\overline{B_i \oplus B_j}) \geq t \\ 0, & \text{others} \end{cases}$ and

$f(B_i,B_j) = \begin{cases} \tau, & \text{if } wt(\overline{B_i \oplus B_j}) \geq t \\ 0, & \text{others} \end{cases}$.

Explanation of Similarity Degree

The Meaning of Similarity Degree

The higher $\mathbb{S}_{B_i,B_j}$, the higher possibility for two vectors B_i and B_j in the same partition.

Benefit of Similarity Degree

The *similarity degree* of B_i and B_j considers not only the relationship between B_i and B_j but also the relationship from other B_k, which can help us to collect all the correlation between 240 candidate binary vectors synthetically with suitable value of t, ξ, τ. (We let $\tau = 1, \xi = 2, t = 10$ in our experiment)

Explanation of Similarity Degree

The Meaning of Similarity Degree

The higher $\mathbb{S}_{B_i,B_j}$, the higher possibility for two vectors B_i and B_j in the same partition.

Benefit of Similarity Degree

The *similarity degree* of B_i and B_j considers not only the relationship between B_i and B_j but also the relationship from other B_k, which can help us to collect all the correlation between 240 candidate binary vectors synthetically with suitable value of t, ξ, τ. (We let $\tau = 1, \xi = 2, t = 10$ in our experiment)

New Voting Method

After we partition four parts $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ based on magnitude of *similarity degree*, then we propose a new voting method as following.

New Voting Method

For a given part Φ_l, $1 \leq l \leq 4$, for each $0 \leq x \leq 15$, we compute

$$v_{l,x} = \sum_{\alpha \in \Phi_l} \left(\sum_{\beta \in \Phi_l, \beta_x = 0} \mathbb{S}_{\alpha,\beta} - \sum_{\beta \in \Phi_l, \beta_x = 1} \mathbb{S}_{\alpha,\beta} \right)$$

and transform the vectors $(v_{l,0}, v_{l,1}, \cdots, v_{l,15})$ into a binary vector such that the eight highest counter values correspond to '1'-bits and the remaining correspond to '0'-bits.

New Voting Method

By This New Voting Method

We can get four candidates coordinate functions of secret S-box.

The Third Improvement

Question 3?

The candidate vectors we found might not be complete the correct ones. How to find all correct coordinate functions of secret S-box with lower data complexity?

Our Idea

we consider a method of constructing all correct coordinate functions of secret S-box by using pruning search algorithm.

The Third Improvement

Question 3?

The candidate vectors we found might not be complete the correct ones. How to find all correct coordinate functions of secret S-box with lower data complexity?

Our Idea

we consider a method of constructing all correct coordinate functions of secret S-box by using pruning search algorithm.

The Third Improvement

Balance Filter

The function consists of 4 parallel applications of coordinate functions must be balanced.

The probability of 4 random vectors passing this filter is equal to

$$\frac{15!}{(C_{15}^{7})^4} \approx 2^{-10.36}$$

by which, many wrong candidate coordinate functions may be found by this filter.

Pruning Search Algorithm

In order to reduce the complexity, we change a few bits of candidate vectors instead of discarding while fail to pass the balance filter.

Outline

1. Introduction
   - Description of PRESENT-like Cipher
   - Previous Work
2. Our Contributions
   - Main Techniques
   - Experiments
3. Conclusion

The Complexity on the Cipher Maya

Borghoff's Linear Attack

They can recover part of coordinate functions of the secret S-box on 10 rounds Maya with 2^{25} data complexity. But the success rate was not presented.

Table: The data and time complexity and success rate of recovering all four correct coordinate functions on 10 to 16 rounds Maya cipher in this paper

Rounds	10	11	12	13	14	15	16
Data complexity	$2^{24.0}$	$2^{26.3}$	$2^{27.9}$	$2^{29.5}$	2^{31}	$2^{34.2}$	2^{36}
Time complexity	$2^{17.8}$	$2^{18.7}$	$2^{15.2}$	$2^{16.7}$	$2^{18.9}$	$2^{18.7}$	$2^{18.9}$
Success rate	94.0%	89.5%	90.0%	93.0%	91.5%	88.5%	87.5%

The Complexity on the Cipher Maya

Borghoff's Linear Attack

They can recover part of coordinate functions of the secret S-box on 10 rounds Maya with 2^{25} data complexity. But the success rate was not presented.

Table: The data and time complexity and success rate of recovering all four correct coordinate functions on 10 to 16 rounds Maya cipher in this paper

Rounds	10	11	12	13	14	15	16
Data complexity	$2^{24.0}$	$2^{26.3}$	$2^{27.9}$	$2^{29.5}$	2^{31}	$2^{34.2}$	2^{36}
Time complexity	$2^{17.8}$	$2^{18.7}$	$2^{15.2}$	$2^{16.7}$	$2^{18.9}$	$2^{18.7}$	$2^{18.9}$
Success rate	94.0%	89.5%	90.0%	93.0%	91.5%	88.5%	87.5%

Conclusion

- First, we present a new technique to support consistency of partitions of the input to the secret S-box of the first S-box layer.
- Our second new idea is that we propose a new method to get the coordinate functions of secret S-boxes efficiently based on more information.
- The third new technique is that we present a method of constructing the correct coordinate functions by pruning search algorithm

The End

Thanks for your attention!