An Efficient Affine Equivalence Algorithm for Multiple S-Boxes and a Structured Affine Layer

Jung Hee Cheon[1], Hyunsook Hong[1], **Joohee Lee**[1], and Jooyoung Lee[2]

[1] Seoul National University (SNU), Seoul, Korea
[2] KAIST, Daejeon, Korea

2016. 08. 12.

Contents

Affine Equivalence Problem and Previous Works

Problem (Affine Equivalence Problem)

For given permutations $F, S : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, *find affine mappings* $A, B : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ *satisfying* $F = B \circ S \circ A$ *if they exist.*

Affine Equivalence Problem and Previous Works

Problem (Affine Equivalence Problem)

For given permutations $F, S : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, *find affine mappings* $A, B : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ *satisfying* $F = B \circ S \circ A$ *if they exist.*

- Naive approach to solve the problem takes $O(n^3 2^{n^2+n})$ times: $\forall A$, to check if $B = F \circ A^{-1} \circ S^{-1}$ is affine and invertible.
- The Affine Equivalence Algorithm proposed by Biryukov et al. in Eurocrypt 2003 recovers both A and B in $O(n^3 2^{2n})$ times.

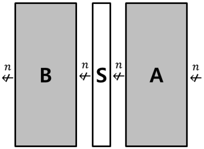
Affine Equivalence Problem and Previous Works

Problem (Affine Equivalence Problem)

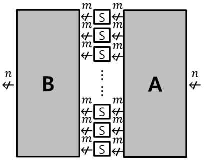
For given permutations $F, S : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, find affine mappings $A, B : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ satisfying $F = B \circ S \circ A$ if they exist.

- Naive approach to solve the problem takes $O(n^3 2^{n^2+n})$ times: $\forall A$, to check if $B = F \circ A^{-1} \circ S^{-1}$ is affine and invertible.
- The Affine Equivalence Algorithm proposed by Biryukov et al. in Eurocrypt 2003 recovers both A and B in $O(n^3 2^{2n})$ times.
- Baek et al. proposed a Specialized Affine Equivalence Algorithm to solve the problem with multiple m-bit S-Boxes in
  - Case 1. With F^{-1} queries: $O(\frac{n}{m} \cdot n^3 \cdot 2^{3m})$ times.
  - Case 2. Without F^{-1} queries: $O(\min\{\frac{n}{m} \cdot n^{m+3} \cdot 2^{2m}, \quad \frac{n}{m} \cdot n^3 \cdot 2^{3m} + n \log n \cdot 2^{n/2}\})$ times.

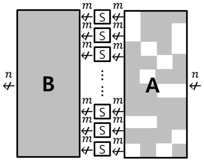
Affine Equivalence Problem and Previous Works



(a) The original problem



(b) Baek et al.'s consideration



(c) Our problem: A with empty $m \times m$ blocks

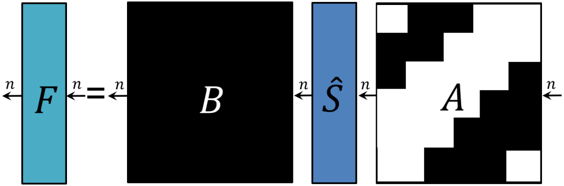
Biryukov et al.'s solution	**Their solution**	**Our solution**
• General	• Used to attack WB implementations	• Does not require to evaluate F^{-1} $\Longrightarrow$ Efficient!
• Used as a module for many known attacks	• Requires several evaluations of F^{-1} $\Longrightarrow$ Complexity mainly depends on this part	• Applicable to attack WB implementation

Look-up table sizes: (a) $n \cdot 2^n =$ (b) $n \cdot 2^n >>$ (c) $\frac{n}{m} \cdot n \cdot 2^{km}$,
where k blocks are filled in each rows in A in (c).

Our Problem

Problem (Our Specialized Affine Equivalence Problem)

Let $F, \hat{S}$ be given n-bit permutations s.t. $\hat{S}$ is a concatenation of m-bit S-Boxes for $n = m \cdot s$. Suppose that there exists a pair of affine maps $A, B : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ s.t. $F = B \circ \hat{S} \circ A$ and A has a certain known structure w.r.t. m.[1] Find A' and B' s.t. $F = B' \circ S \circ A'$ and A' has the same structure with A.



[1]We call it as "*structured*"

Our Problem

Definition (Structured Matrix, Structured Affine Map)

A matrix $L \in \mathbb{Z}_2^{n \times n}$ is called structured w.r.t. m where $n = m \cdot s$, if

1. L is invertible and
2. defining the $s \times s$ matrix M_L as
$$(M_L)_{i,j} = \begin{cases} 0 & \text{if } (i,j)\text{-th } m \times m \text{ block of } L \text{ is zero} \\ 1 & \text{Otherwises} \end{cases}$$
, the rows of M_L are pairwise distinct.

An affine map is called structured w.r.t. m if the linear part of the affine map is structured w.r.t. m.

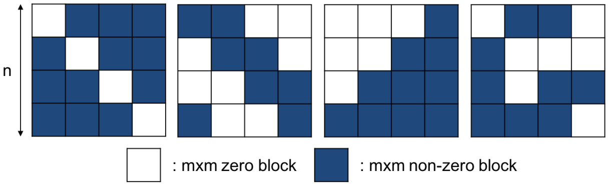
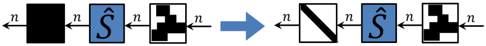


Figure: Examples of structured matrix

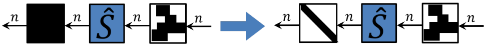
Sketch of Attacks

Step1. WANT:

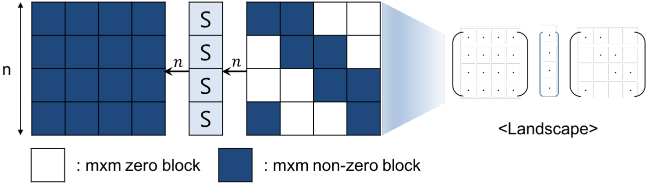


Sketch of Attacks

Step1. WANT:



- Once viewing F in a landscape,



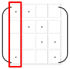
We do differential attacks. That is, fixing $P_1 + P_2 = \begin{pmatrix} \cdot \\ \\ \\ \end{pmatrix}$,
observe $F(P_1) + F(P_2) \in \mathbb{Z}_2^n$.

- Observation:

$$\dim\{F(P_1') + F(P_2') \mid P_1' + P_2' = P_1 + P_2\} = 2m \ (\ll n)$$

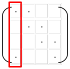
- <u>Observation:</u>

$$\dim\{F(P_1') + F(P_2') \mid P_1' + P_2' = P_1 + P_2\} = 2m \ (\ll n)$$

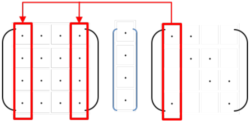
$\implies$ *Why?:* Because of the first column  of A.

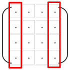
- Observation:

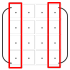
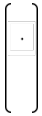
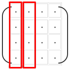
$$\dim\{F(P_1') + F(P_2') \mid P_1' + P_2' = P_1 + P_2\} = 2m \ (\ll n)$$

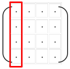
$\implies$ *Why?:* Because of the first column  of A.

Moreover, since the differential activates the first column of A, and the first column of A activates the first and the last column of B depicted as

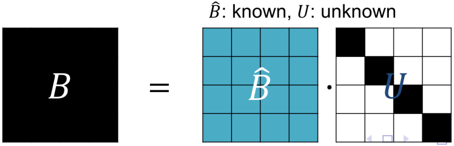


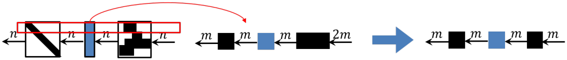
, *we can see the subspace* $\{F(P_1') + F(P_2') \mid P_1' + P_2' = P_1 + P_2\}$ *of* $\mathbb{Z}_2^n$ *is generated by*  *of* B.

- Fixing $P_1 + P_2 =$ , we obtain the column space generated by  of B over $\mathbb{Z}_2$.
- Fixing $P_3 + P_4 =$ , we obtain the column space generated by  of B over $\mathbb{Z}_2$.

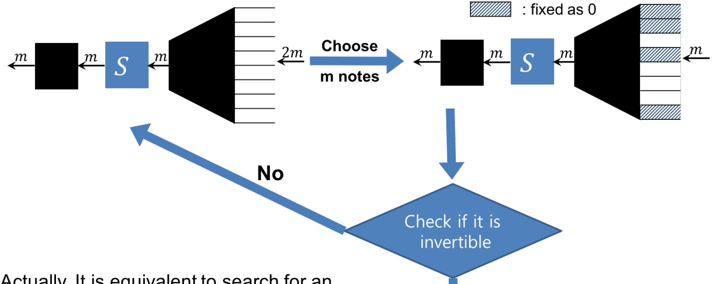
By calculating an intersection of two subspaces over $\mathbb{Z}_2$ obtained as above, we achieve a basis of the column space of  of B.

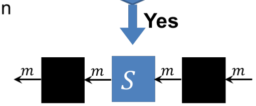
$(\therefore)$ *Repeating this process for* $\binom{n}{m}$ *times, as a result, we can decompose* B *as*



Step2. WANT: 

- Return to bit scale.

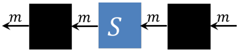




* Actually, It is equivalent to search for an $(m \times m)$ invertible submatrix in $(m \times 2m)$ random full-rank matrix ;

It terminates in 5 iterations in average.

- Apply AEA to solve the affine equivalence problem for



Main Theorem and Comparisons

Theorem (Solving the Specialized Affine Equivalence Problem)

Let $F, \hat{S}$ be given n-bit permutations with the same conditions as in the problem setting. One can solve the specialized affine equivalence problem for F and $\hat{S}$ in time

$$5 \cdot \left(\frac{n}{m} \cdot \log_2 \frac{n}{m} \right) \cdot n^3 + 5 \cdot n^2 \cdot 2^m + n \cdot m^2 \cdot 2^{2m}$$

with $\frac{n}{m}(2n + 5 \cdot 2^m + m + 10)$ chosen plaintexts.

We significantly reduced the complexity of solving affine equivalence problems for the special cases.

- We reduced the main terms of complexity proposed by Baek et al. since we don't need F^{-1} calculations.
- Even with F^{-1} oracle, Baek et al. approach requires $O(\frac{n}{m} \cdot n^3 \cdot 2^{3m})$ time complexity which is larger than ours.

Main Theorem and Comparisons

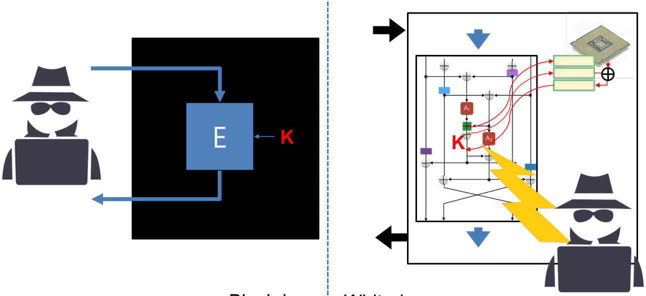
Example. Considering several sample parameters, required work factors to solve our problems are as below.

- **Case 1.** $n = 128, m = 8$
  (a)AEA: 2^{277} , (b)Baek et al. SAEA: 2^{75} , (c)Our Algorithm: 2^{31}
- **Case 2.** $n = 256, m = 8$
  (a)AEA: 2^{536} , (b)Baek et al. SAEA: 2^{110} , (c)Our Algorithm: 2^{34}
- **Case 3.** $n = 256, m = 16$
  (a)AEA: 2^{536} , (b)Baek et al. SAEA: 2^{188} , (c)Our Algorithm: 2^{48}

Application to White-Box Implementations

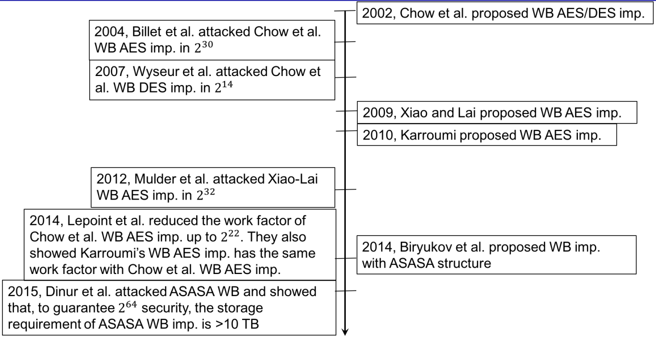
What is “White-Box implementation” ?

- Goal: Obfuscating secret keys in the software



- Applications
  - iOS upgrades
  - Digital Rights Management(DRM):
    Games, recorded music, newspapers, films, magazines

Brief History of White-Box Cryptography



- In this area, it seemed to be hard to construct a WB imp. with a work factor more than 2^{35} and a reasonable storage requirement.
- Baek et al. challenged to resolve this problem, proposed a WB imp. of claimed complexities 2^{75} and 2^{110} with storage requirements 16MB and 64MB, respectively. *However, the construction is vulnerable to our attack algorithm so that they couldn't achieve the security goals.*

Conclusion

- For n-bit permutations F and $\hat{S}$, the complexity of solving an instance of the affine equivalence problem is highly reduced up to

$$5 \cdot \left(\frac{n}{m} \cdot \log_2 \frac{n}{m}\right) \cdot n^3 + 5 \cdot n^2 \cdot 2^m + n \cdot m^2 \cdot 2^{2m},$$

  where $\hat{S}$ is a concatenation of m-bit S-boxes and the input affine layer is structured with respect to m.

- Our algorithm will serve as a useful attack tool for White-Box implementations. Actually, with our methods, we can extract the secret key of White-Box AES implementation proposed by Baek et al. with work factors $2^{32}, 2^{33}$, and 2^{34} for $n = 128, 256$ and 384, respectively, while claimed security were $2^{75}, 2^{110}$, and 2^{117}.

Further Works

- To implement the whole attack algorithms
- Can we generalize our attack method to solve the original Affine Equivalence problems?
- To construct a secure White-Box implementations with an appropriate storage requirement

Thank you for your attention!
Any questions?