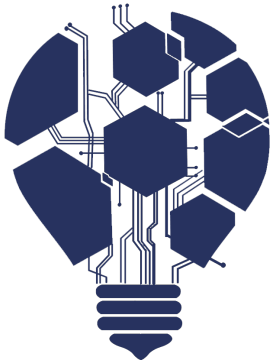
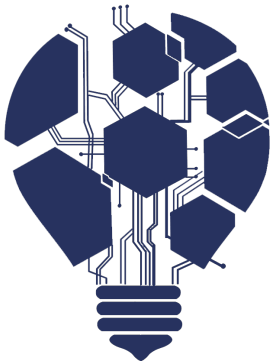
Implicit WBC implementation : ARX cipher

25.01.13

류지은





White-Box Implementation

A. Ranea, et al., Implicit white-box implementations: White-boxing ARX ciphers, CRYPTO 2022 [4]

A. Biryukov, et al., Cryptanalysis of ARX-based White-box Implementations, TCHES 2023 [2]

Implicit White-Box Implementations: White-Boxing ARX Ciphers

Adrián Ranea[1][0000−0002−8697−7423], Joachim Vandersmissen[2], and Bart Preneel[1][0000−0003−2005−9651]

[1] imec-COSIC, KU Leuven, Belgium
firstname.lastname@esat.kuleuven.be
[2] atsec information security
joachim@atsec.com

Abstract. Since the first white-box implementation of AES published twenty years ago, no significant progress has been made in the design of secure implementations against an attacker with full control of the device. Designing white-box implementations of existing block ciphers is a challenging problem, as all proposals have been broken. Only two white-box design strategies have been published this far: the CEJO framework, which can only be applied to ciphers with small S-boxes, and self-equivalence encodings, which were only applied to AES.
In this work we propose implicit implementations, a new design of white-box implementations based on implicit functions, and we show that current generic attacks that break CEJO or self-equivalence implementations are not successful against implicit implementations. The generation and the security of implicit implementations are related to the self-equivalences of the non-linear layer of the cipher, and we propose a new method to obtain self-equivalences based on the CCZ-equivalence. We implemented this method and many other functionalities in a new open-source tool `BoolCrypt`, which we used to obtain for the first time affine, linear, and even quadratic self-equivalences of the permuted modular addition. Using the implicit framework and these self-equivalences, we describe for the first time a practical white-box implementation of a generic Addition-Rotation-XOR (ARX) cipher, and we provide an open-source tool to easily generate implicit implementations of ARX ciphers.

Keywords: White-box cryptography · Self-equivalence · Implicit implementation · ARX

IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 3, pp. 97–135. DOI:10.46586/tches.v2023.i3.97-135

Cryptanalysis of ARX-based White-box Implementations*

Alex Biryukov, Baptiste Lambin and Aleksei Udovenko

University of Luxembourg, Esch-sur-Alzette, Luxembourg
firstname.lastname@uni.lu

Abstract. At CRYPTO'22, Ranea, Vandersmissen, and Preneel proposed a new way to design white-box implementations of ARX-based ciphers using so-called *implicit* functions and quadratic-affine encodings. They suggest the Speck block-cipher as an example target.
In this work, we describe practical attacks on the construction. For the implementation without one of the external encodings, we describe a simple algebraic key recovery attack. If both external encodings are used (the main scenario suggested by the authors), we propose optimization and inversion attacks, followed by our main result - a multiple-step round decomposition attack and a decomposition-based key recovery attack.
Our attacks only use the white-box round functions as oracles and do not rely on their description. We implemented and verified experimentally attacks on white-box instances of Speck-32/64 and Speck-64/128. We conclude that a single ARX-round is too weak to be used as a white-box round.

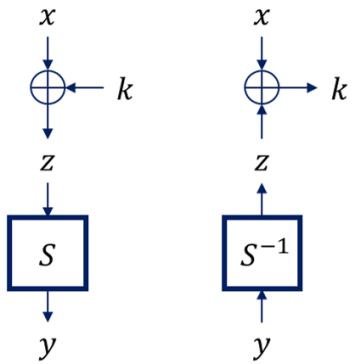
Keywords: White-box cryptography · Cryptanalysis · Algebraic attacks · Decomposition attacks

White-Box Implementation

초기 화이트박스 구현 아이디어

S가 공개되어 있고 공격자가 입력 x를 볼 수 있으므로

블록암호의 비선형 요소인 S-layer를 그냥 저장하면 $S(x \oplus k)$에서 k가 그냥 노출되고, k가 고정된 테이블 $S_k(x)$로 저장해도 키를 추출할 수 있다.
이때 해당 테이블에 인코딩 (A, B)를 적용한 $B \circ S_k \circ A$를 테이블로 저장하면 k를 확정할 수 없다.



CEJO 화이트박스 구현[3]

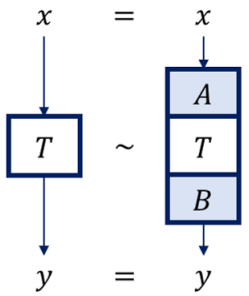
only small encoding

비선형 인코딩의 크기를 키우려면 너무 많은 메모리가 필요하고,
작은 크기의 비선형 인코딩을 사용하면 BGE attack과 같은 키 추출 공격이 가능하다[1].

Self-equivalence 화이트박스 구현[5]

only self-equivalence encoding ; affine encoding을 추가해도 결국 SE set으로 reduction됨

CEJO 화이트박스 구현과 달리 전체 라운드 함수 T에 큰 인코딩을 적용할 수 있다.
S-box를 만들 수 있는 $x \mapsto x^d$ 같은 power function은 x의 비트 길이가 길 때 큰 affine self-equivalence를 가진다고 알려진 함수 중 하나이다. 그러나 S-layer의 인코딩 크기는 layer를 구성하는 S-box의 인코딩에 대한 크기로 reduction 되고, 일반적으로 S-box가 작아서 메모리 문제 power function의 인코딩 크기도 작다는 문제가 있다.



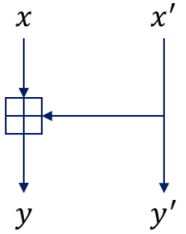
Implicit 화이트박스 구현[4]

기존 CEJO나 self-equivalence 구현에 수행된 일반적인 공격들에 저항성을 가진다.

White-Box Implementation

ARX cipher에서 주로 사용되는 비선형 함수인 n-bit modular addition

$$(x, x') \mapsto (x \boxplus x', x') = (x + x' \bmod 2^n, x')$$



은 2011년 Ernst Schulte-Geers[6]에 의해 이차 함수와 CCZ-equivalence라는 것이 밝혀졌다.
CCZ-equivalence는 두 함수의 graph 속성을 보존하며,
따라서 이차함수의 self-equivalence set이 크면
modular addition의 self-equivalence set 역시 클 것으로 예측할 수 있다.

단, 저자들이 구한 self-equivalence 구조가 sparse shape과 low-entropy constant vector를 가진다는 문제가 있어, 이를 해결하기 위해 implicit implementation을 적용한다. Implicit implementation은 큰 self-equivalence 인코딩을 사용하여 일반적으로 self-equivalence 구현에 수행되는 구조 공격을 막는다.

Preliminaries

Implicit white-box implementation을 위해 알아 둘 주요 사전 개념에는 *implicit function*, *self-equivalence*, *graph automorphism*이 있다.

Implicit function

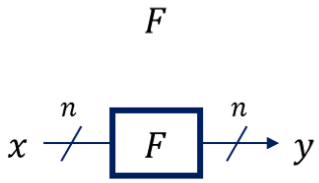
Let $x, y \in \mathbb{F}_2^n$ and F be a n-bit function s.t. $y = F(x)$. A $(2n, m)$-bit function P is called an *implicit function* of F if it is satisfies

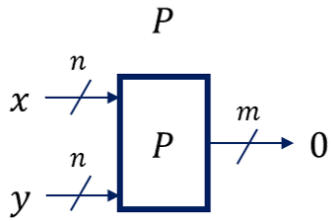
$$P(x, y) = 0^m \iff y = F(x).$$

Implicit function 자명한 예로,

$$P(x, y) = y \oplus F(x) = 0^n \iff F(x) = y$$

이다.





Preliminaries

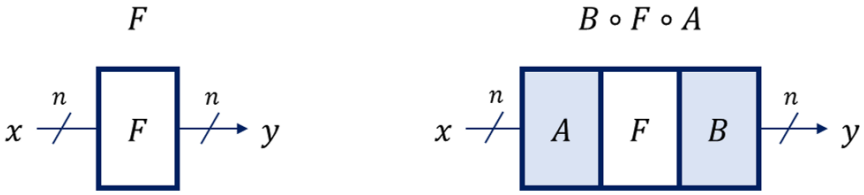
Self-equivalence

A *self-equivalence* of a function F is a pair of permutations (A, B) s.t.

$$F(x) = (B \circ F \circ A)(x).$$

If (A, B) is a self-equivalence of F, A is right self-equivalence of F and B is left self-equivalence of F.
Moreover, (A, B) is an affine-quadratic[a] self-equivalence, if A is affine and B is quadratic.

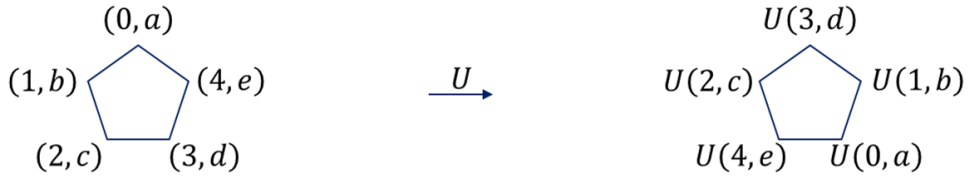
[a]선형 인코딩은 cryptanalysis에 취약하고, 3차 이상의 인코딩은 방정식 형태로 저장하는 음함수 구현 방식상 메모리 한계가 있음



Graph automorphism

Let Γ_F is a graph of F s.t. $\Gamma_F := \{(x, y) : y = F(x), x \in \mathbb{F}_2^n\}$. A *graph automorphism* of F is a permutation U s.t.

$$\Gamma_F = U(\Gamma_F).$$



$$\Gamma_F = \{(0,a), (1,b), (2,c), (3,d), (4,e)\}$$

$$\Gamma_F = \{U(3,d), U(2,c), U(4,e), U(0,a), U(1,b)\} = \{(0,a), (1,b), (2,c), (3,d), (4,e)\}$$

Preliminaries

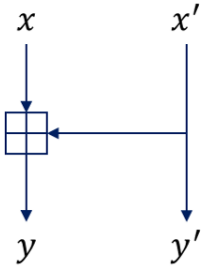
Speck

Speck is a family of lightweight block ciphers publicly released by the NSA in June 2013. Speck is an add-rotate-xor(ARX) cipher[a].

[a] Wikipedia, Speck (cipher), https://en.wikipedia.org/wiki/Speck_(cipher)

Modular addition $\boxplus$을 비선형 레이어 S라고 하면

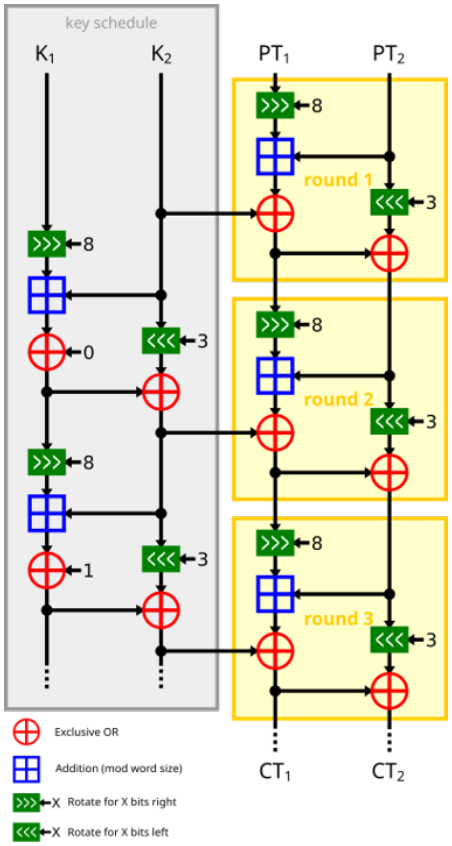
$$S(x, x') = (x \boxplus x', x') = (y, y')$$

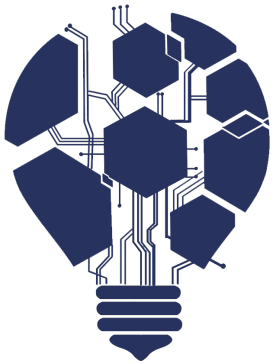


이고, $k^{(i)}$가 라운드 키, $L^{(i)}$가 선형 레이어라고 할 때 i번째 라운드 함수 $E^{(i)}$는

$$E^{(i)}_{k^{(i)}}(x, x') = (L^{(i)} \circ S)(x, x')$$

이다. Speck 라운드의 첫 선형 레이어는 이전 라운드의 마지막 부분으로 이동 가능





Explicit White-Box Implementation

키 k를 사용하는 r 라운드로 구성된 암호화 함수 E_k가 다음과 같다고 하자. 편의상 라운드 키 표기 생략

$$E_k = E^{(r)} \circ E^{(r-1)} \circ \cdots \circ E^{(2)} \circ E^{(1)}$$

임의의 라운드 함수 $E^{(i)}$는

$$E^{(i)} = L^{(i)} \circ S$$

이고, 비선형 함수 S는 다음과 같은 affine-quadratic self-equivalence $(A, B) \in SE(S)$를 가진다. $B \circ S \circ A$의 음함수를 생각할 때, x에 대하여 affine, y에 대하여 quadratic 인코딩을 붙여 y에 대한 2차 연립 방정식을 푸는 문제를 만드는 것처럼 보일 수 있으나, 뒤에서 인코딩된 라운드 함수를 구성하는 과정 중 B가 x에 대한 인코딩에 사용됨

$$S = B^{(i)} \circ S \circ A^{(i)}$$

특히, $E^(i)$에 대한 self-equivalence set의 크기는 S에 대한 self-equivalence set의 크기와 같으므로 $(A, B) \in SE(S)$로부터 $E^{(i)}$의 self-equivalence set을 표현할 수 있다[5].

이를 통해 다음과 같이 라운드 함수 $E^{(i)}$의 self-equivalence를 구한다.

$$\begin{aligned} E^{(i)} &= L^{(i)} \circ S \\ &= (L^{(i)} \circ B^{(i)} \circ (L^{(i)})^{-1}) \circ L^{(i)} \circ S \circ (A^{(i)}) \\ &= \widehat{B^{(i)}} \circ E^{(i)} \circ \widehat{A^{(i)}} \end{aligned}$$

즉,

$$(\widehat{A^{(i)}}, \widehat{B^{(i)}}) = (A^{(i)},\ L^{(i)} \circ B^{(i)} \circ (L^{(i)})^{-1}) \in SE(E^{(i)}).$$

Explicit White-Box Implementation

한편, affine permutation $C^{(i+1)}$에 대하여

$$E^{(i)} = \widehat{B^{(i)}} \circ (C^{(i+1)})^{-1} \circ C^{(i+1)} \circ E^{(i)} \circ \widehat{A^{(i)}}$$

이고, 연속된 두 라운드에 대하여 다음이 성립한다.

$$E^{(i)} \circ E^{(i-1)} = (\widehat{B^{(i)}} \circ (C^{(i+1)})^{-1} \circ \left[C^{(i+1)} \circ E^{(i)} \circ \widehat{A^{(i)}}) \circ (\widehat{B^{(i-1)}} \circ (C^{(i)})^{-1}\right] \circ C^{(i)} \circ E^{(i-1)} \circ \widehat{A^{(i-1)}})$$

인코딩된 라운드 함수 $\overline{E^{(i)}}$를 다음과 같이 정의하면,

$$\overline{E^{(i)}} = C^{(i+1)} \circ E^{(i)} \circ \widehat{A^{(i)}} \circ \widehat{B^{(i-1)}} \circ (C^{(i)})^{-1}$$

인코딩된 암호화 함수는 다음과 같다.

$$\begin{aligned}
\overline{E_k} &= \overline{E^{(r)}} \circ \overline{E^{(r-1)}} \circ \cdots \circ \overline{E^{(2)}} \circ \overline{E^{(1)}} \\
&= \left(C^{(r+1)} \circ E^{(r)} \circ \widehat{A^{(r)}} \circ \widehat{B^{(r-1)}} \circ (C^{(r)})^{-1}\right) \circ \left(C^{(r)} \circ E^{(r-1)} \circ \widehat{A^{(r-1)}} \circ \widehat{B^{(r-2)}} \circ (C^{(r-1)})^{-1}\right) \\
&\quad \circ \cdots \circ \left(C^{(3)} \circ E^{(2)} \circ \widehat{A^{(2)}} \circ \widehat{B^{(1)}} \circ (C^{(2)})^{-1}\right) \circ \left(C^{(2)} \circ E^{(1)} \circ \widehat{A^{(1)}} \circ \widehat{B^{(0)}} \circ (C^{(1)})^{-1}\right) \\
&= \left(\widehat{B^{(r)}} \circ (C^{(r+1)})^{(-1)}\right)^{-1} \circ \left(\left(\widehat{B^{(r)}} \circ (C^{(r+1)})^{(-1)}\right) \circ C^{(r+1)} \circ E^{(r)} \circ \widehat{A^{(r)}}\right) \\
&\quad \circ \left(\widehat{B^{(r-1)}} \circ (C^{(r)})^{-1} \circ C^{(r)} \circ E^{(r-1)} \circ \widehat{A^{(r-1)}}\right) \circ \cdots \circ \left(\widehat{B^{(1)}} \circ (C^{(2)})^{-1} \circ C^{(2)} \circ E^{(1)} \circ \widehat{A^{(1)}}\right) \circ \widehat{B^{(0)}} \circ (C^{(1)})^{-1} \\
&= \left(\widehat{B^{(r)}} \circ (C^{(r+1)})^{(-1)}\right)^{-1} \circ E^{(r)} \circ E^{(r-1)} \circ \cdots \circ E^{(1)} \circ \left(\widehat{B^{(0)}} \circ (C^{(1)})^{-1}\right)
\end{aligned}$$

Implicit White-Box Implementation

따라서

$$\overline{E_k} = \left(B^{(r)} \circ (C^{(r+1)})^{(-1)}\right)^{-1} \circ E_k \circ \left(\widehat{B^{(0)}} \circ (C^{(1)})^{-1}\right)$$

이고, $\overline{E_k}$를 이용하여 음함수 구현을 수행한다. 즉, 음함수 구현은 E_k와 functionally equivalent하지 않음

인코딩된 라운드 함수

$$\begin{aligned}\overline{E^{(i)}} &= C^{(i+1)} \circ E^{(i)} \circ \widehat{A^{(i)}} \circ \widehat{B^{(i-1)}} \circ (C^{(i)})^{-1} \\ &= C^{(i+1)} \circ L^{(i)} \circ S \circ \widehat{A^{(i)}} \circ \widehat{B^{(i-1)}} \circ (C^{(i)})^{-1}\end{aligned}$$

를 음함수로 표현하면 다음과 같다.

$$P^{(i)} = T \circ \left(Id, (L^{(i)})^{-1}\right) \circ \left(\widehat{A^{(i)}}, Id\right) \circ \left(\widehat{B^{(i-1)}}, Id\right) \circ \left((C^{(i)})^{-1}, (C^{(i+1)})^{-1}\right)$$

상기 식에서 T는 S의 음함수 방정식이다. 이때 함수 A, B에 대하여 $(A, B)(x, y) := (A(x), B(y))$이다.

나아가 $(U^{(i)}, V^{(i)})$가 T의 self-equivalence라고 하면

$$T = V^{(i)} \circ T \circ U^{(i)}$$

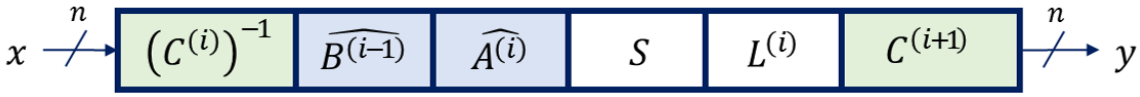
이므로,

$$P^{(i)} = V^{(i)} \circ T \circ U^{(i)} \circ \left(Id, (L^{(i)})^{-1}\right) \circ \left(\widehat{A^{(i)}}, Id\right) \circ \left(\widehat{B^{(i-1)}}, Id\right) \circ \left((C^{(i)})^{-1}, (C^{(i+1)})^{-1}\right)$$

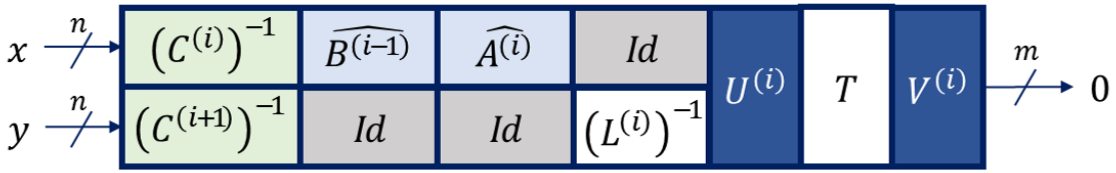
이다.

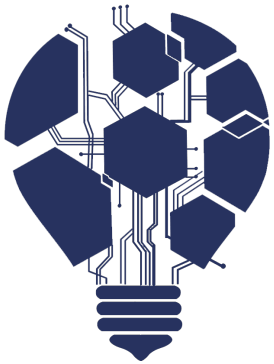
Implicit White-Box Implementation

$$\overline{E^{(i)}} = \left(C^{(i)}\right)^{-1} \circ \widehat{B^{(i-1)}} \circ \widehat{A^{(i)}} \circ E^{(i)} \circ C^{(i+1)}$$



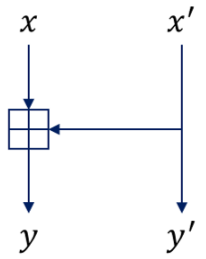
$$P^{(i)} = V^{(i)} \circ T \circ U^{(i)} \circ \left(Id, \left(L^{(i)}\right)^{-1}\right) \circ \left(\widehat{A^{(i)}}, Id\right) \circ \left(\widehat{B^{(i-1)}}, Id\right) \circ \left(\left(C^{(i)}\right)^{-1}, \left(C^{(i+1)}\right)^{-1}\right)$$





CCZ-equivalence

ARX cipher에서 S-layer를 구성하는 modular addition은 y에 대한 n차 방정식으로 표현된다.



$$(x, x') \mapsto (y, y') = (x \boxplus x', x') = (x + x' \bmod 2^n, x')$$

이는 S-layer를 implicit white-box로 구현할 수 없게 만들기 때문에 메모리 문제, modular addition을 저차 방정식으로 표현할 수 있도록 변환해야 한다.

Schulte-Geers는 2011년 modular addition과 이차 함수 Q가 CCZ-equivalent함을 증명했다[6]. Q는 다음 장에서 설명
따라서 S-layer는 해당 이차 함수를 활용하여 저차 implicit white-box로 구현 가능하다.

CCZ-equivalent

A function F is *CCZ-equivalent* to a function G if the graph of F can be transformed to the graph of G through an affine permutation, that is, if there exists an affine permutation $Z_{G,F}$ such that

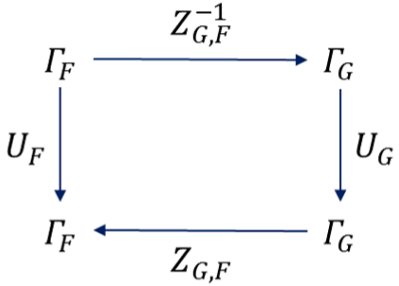
$$\Gamma_F = Z_{G,F}(\Gamma_G).$$

위 CCZ-equivalent인 함수에 대하여 $\Gamma_G = Z_{G,F}^{-1}(\Gamma_F)$ 역시 성립힌다. 나아가 U_G가 G의 graph automorphism일 때 다음이 성립힌다.

$$U_F(\Gamma_F) = \Gamma_F = Z_{G,F}(U_G(\Gamma_G)) = Z_{G,F}(U_G(Z_{G,F}^{-1}(\Gamma_F))) = (Z_{G,F} \circ U_G \circ Z_{G,F}^{-1})(\Gamma_F)$$

$$\text{i.e., } U_F(\Gamma_F) = (Z_{G,F} \circ U_G \circ Z_{G,F}^{-1})(\Gamma_F)$$

즉, 고차 함수 F의 그래프와 그에 대한 graph automorphism U를 저차 함수 G의 그래프와 그에 대한 graph automorphism 및 선형 함수 Z로 표현할 수 있다.



F, G graph의 commutative diagram

CCZ-equivalence

Modular addition $\boxplus$과 CCZ-equivalent한 이차 함수 Q는 $x, x' \in \mathbb{F}_2^n$에 대하여 다음과 같다.

$$Q(x, x') = (0,\ x_0x'_0,\ x_0x'_0 \oplus x_1x'_1,\ \cdots,\ x_0x'_0 \oplus \cdots \oplus x_{n-2}x'_{n-2})$$

Theorem

The linear mapping

$$Z : (\mathbb{F}_2^n)^3 \to (\mathbb{F}_2^n)^3;\ (x, x', y) \mapsto Z(x, x', y) = (x \oplus y, x' \oplus y, x \oplus x' \oplus y)$$

maps the graph of Γ_Q bijectively onto the graph of $\Gamma_{\boxplus}$, $\Gamma_{\boxplus} = Z(\Gamma_Q)$, i.e., Γ_Q is CCZ-equivalent of $\Gamma_{\boxplus}$.

Proof

$c = c(x, x')$이 x, x'의 modular addition에서 발생하는 carry vector일 때,
$(\tilde{x}, \tilde{x'}) = \beta(x, x') := (x \oplus c(x, x'),\ x' \oplus c(x, x'))$이라고 하자.

$\Gamma_{\boxplus} := \{(x, x', x \boxplus x') : x, x' \in \mathbb{F}_2^n\}$이고 $x \boxplus x' = x \oplus x' \oplus c(x, x')$이므로, $\Gamma_{\boxplus}$를 다음과 같이 표현 가능하다.

$$\Gamma_{\boxplus} := \{(x \oplus c(x, x') \oplus c(x, x'), x' \oplus c(x, x') \oplus c(x, x'), x \oplus x' \oplus c(x, x')) : x, x' \in \mathbb{F}_2^n\}$$

CCZ-equivalence

또한, $c(x, x') = Q(x \oplus c(x, x'),\ x' \oplus c(x, x'))$이다.

proof

x, x'의 carry vertor $c = c(x, x')$에 대하여, $c_0 = 0$은 자명하다.

$0 < j < n-1$에 대하여, $(x \boxplus x')_j = x_j \oplus x'_j \oplus c_j$이고, $c_{j+1} = x_j x'_j \oplus (x_j \oplus x'_j)c_j$이므로,

$$\begin{aligned} c_{j+1} \oplus c_j &= (x_j x'_j \oplus (x_j \oplus x'_j)c_j) \oplus c_j \\ &= x_j x'_j \oplus (x_j c_j \oplus x'_j c_j) \oplus c_j^2 \qquad \because c_j^2 = c_j \\ &= (x_j \oplus c_j)(x'_j \oplus c_j) \end{aligned}$$

즉, $c_{j+1} \oplus c_j = (x_j \oplus c_j)(x'_j \oplus c_j)$이다.

다음으로,

$$\begin{pmatrix} 0 \\ c_1 \oplus c_0 \\ c_2 \oplus c_1 \\ c_3 \oplus c_2 \\ \vdots \\ c_{n-2} \oplus c_{n-3} \\ c_{n-1} \oplus c_{n-2} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_0 \oplus c_1 \\ c_1 \oplus c_2 \\ c_2 \oplus c_3 \\ \vdots \\ c_{n-3} \oplus c_{n-2} \\ c_{n-2} \oplus c_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix}$$

CCZ-equivalence

그리고

$$\begin{pmatrix} 0 \\ (x_0 \oplus c_0)(x'_0 \oplus c_0) \\ (x_1 \oplus c_1)(x'_1 \oplus c_1) \\ (x_2 \oplus c_2)(x'_2 \oplus c_2) \\ \vdots \\ (x_{n-3} \oplus c_{n-3})(x'_{n-3} \oplus c_{n-3}) \\ (x_{n-2} \oplus c_{n-2})(x'_{n-2} \oplus c_{n-2}) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} (x_0 \oplus c_0)(x'_0 \oplus c_0) \\ (x_1 \oplus c_1)(x'_1 \oplus c_1) \\ (x_2 \oplus c_2)(x'_2 \oplus c_2) \\ (x_3 \oplus c_3)(x'_3 \oplus c_3) \\ \vdots \\ (x_{n-2} \oplus c_{n-2})(x'_{n-2} \oplus c_{n-2}) \\ (x_{n-1} \oplus c_{n-1})(x'_{n-1} \oplus c_{n-1}) \end{pmatrix}$$

에 대하여, $c_{j+1} \oplus c_j = (x_j \oplus c_j)(x'_j \oplus c_j)$이므로

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} (x_0 \oplus c_0)(x'_0 \oplus c_0) \\ (x_1 \oplus c_1)(x'_1 \oplus c_1) \\ (x_2 \oplus c_2)(x'_2 \oplus c_2) \\ (x_3 \oplus c_3)(x'_3 \oplus c_3) \\ \vdots \\ (x_{n-2} \oplus c_{n-2})(x'_{n-2} \oplus c_{n-2}) \\ (x_{n-1} \oplus c_{n-1})(x'_{n-1} \oplus c_{n-1}) \end{pmatrix}$$

CCZ-equivalence

$$\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} (x_0 \oplus c_0)(x'_0 \oplus c_0) \\ (x_1 \oplus c_1)(x'_1 \oplus c_1) \\ \vdots \\ (x_{n-2} \oplus c_{n-2})(x'_{n-2} \oplus c_{n-2}) \\ (x_{n-1} \oplus c_{n-1})(x'_{n-1} \oplus c_{n-1}) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} (x_0 \oplus c_0)(x'_0 \oplus c_0) \\ (x_1 \oplus c_1)(x'_1 \oplus c_1) \\ \vdots \\ (x_{n-2} \oplus c_{n-2})(x'_{n-2} \oplus c_{n-2}) \\ (x_{n-1} \oplus c_{n-1})(x'_{n-1} \oplus c_{n-1}) \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 1 & 1 & 1 & \cdots & 0 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} (x_0 \oplus c_0)(x'_0 \oplus c_0) \\ (x_1 \oplus c_1)(x'_1 \oplus c_1) \\ \vdots \\ (x_{n-2} \oplus c_{n-2})(x'_{n-2} \oplus c_{n-2}) \\ (x_{n-1} \oplus c_{n-1})(x'_{n-1} \oplus c_{n-1}) \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ (x_0 \oplus c_0)(x'_0 \oplus c_0) \\ (x_0 \oplus c_0)(x'_0 \oplus c_0) \oplus (x_1 \oplus c_1)(x'_1 \oplus c_1) \\ \vdots \\ (x_0 \oplus c_0)(x'_0 \oplus c_0) \oplus \cdots \oplus (x_{n-3} \oplus c_{n-3})(x'_{n-3} \oplus c_{n-3}) \\ (x_0 \oplus c_0)(x'_0 \oplus c_0) \oplus \cdots \oplus (x_{n-2} \oplus c_{n-2})(x'_{n-2} \oplus c_{n-2}) \end{pmatrix} = Q(x \oplus c, x' \oplus c)$$

따라서, $c(x, x') = Q(x \oplus c(x, x'),\ x' \oplus c(x, x'))$이다. □

CCZ-equivalence

따라서,

$$\begin{aligned}\Gamma_{\boxplus} :=& \{(x \oplus c(x,x') \oplus c(x,x'), x' \oplus c(x,x') \oplus c(x,x'), x \oplus x' \oplus c(x,x')) : x, x' \in \mathbb{F}_2^n\} \\ =& \{(x \oplus c(x,x') \oplus Q(x \oplus c(x,x'), x' \oplus c(x,x')), \\ & x' \oplus c(x,x') \oplus Q(x \oplus c(x,x'), x' \oplus c(x,x')), \\ & x \oplus x' \oplus Q(x \oplus c(x,x'), x' \oplus c(x,x'))) : x, x' \in \mathbb{F}_2^n\}\end{aligned}$$

이다. β의 정의에 의해

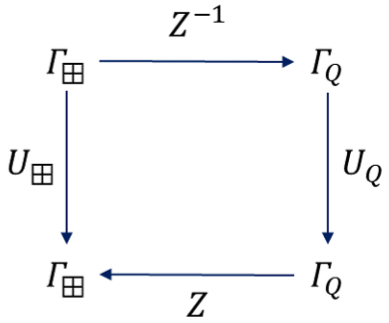
$$\begin{aligned}\Gamma_{\boxplus} :=& \{(\tilde{x} \oplus Q(\tilde{x}, \tilde{x'}),\ \tilde{x'} \oplus Q(\tilde{x}, \tilde{x'}),\ \tilde{x} \oplus \tilde{x'} \oplus Q(\tilde{x}, \tilde{x'})) : \tilde{x}, \tilde{x'} \in \mathbb{F}_2^n\} \\ =& \{Z(\tilde{x}, \tilde{x'}, Q(\tilde{x}, \tilde{x'})) : \tilde{x}, \tilde{x'} \in \mathbb{F}_2^n\} \\ =& Z(\Gamma_Q)\end{aligned}$$

결론적으로 Γ_Q는 $\Gamma_{\boxplus}$의 CCZ-equivalent이다. □

Implicit function of S-layer

Modular addition $\boxplus$의 graph automorphism $U_{\boxplus}$와 이차 함수 Q의 graph automorphism U_Q에 대하여 다음이 성립한다.

$$U_{\boxplus} = Z \circ U_Q \circ Z^{-1}$$



그리고 $\boxplus$의 어떤 음함수 $P_{\boxplus}$에 대하여 $U_{\boxplus}$가 affine permutation일 때, $P_{\boxplus} \circ U_{\boxplus}^{-1}$ 역시 $\boxplus$의 음함수이다.

Proof

$P_{\boxplus}$가 modular addition $\boxplus$의 음함수이므로,

$$P_{\boxplus}(x, x', y) = 0 \iff y = \boxplus(x, x')$$

이고, affine permutation $U_{\boxplus}$에 대하여

$$\begin{aligned}
U_{\boxplus}(\Gamma_{\boxplus}) &= \{U_{\boxplus}(x, x', y) : P_{\boxplus}(x, x', y) = 0, x, x', y \in \mathbb{F}_2^n\} \\
&= \left\{(\hat{x}, \hat{x'}, \hat{y}) : U_{\boxplus}^{-1}(\hat{x}, \hat{x'}, \hat{y}) = (x, x', y), P_{\boxplus}(x, x', y) = 0, x, x', y \in \mathbb{F}_2^n\right\} \\
&= \left\{(\hat{x}, \hat{x'}, \hat{y}) : (P_{\boxplus} \circ U_{\boxplus}^{-1})(\hat{x}, \hat{x'}, \hat{y}) = 0, \hat{x}, \hat{x'}, \hat{y} \in \mathbb{F}_2^n\right\} \\
&= \{(x, x', y) : (P_{\boxplus} \circ U_{\boxplus}^{-1})(x, x', y) = 0, x, x', y \in \mathbb{F}_2^n\} \\
&= \{(x, x', \boxplus(x, x')) : x, x' \in \mathbb{F}_2^n\}
\end{aligned}$$

이므로, $(P_{\boxplus} \circ U_{\boxplus}^{-1})(x, x', y) = 0$는 $\boxplus$의 음함수이다. □

Implicit function of S-layer

$P_{\boxplus} = P_Q \circ U_Q \circ Z^{-1}$이므로

$$\begin{aligned} P_{\boxplus} \circ U_{\boxplus}^{-1} &= (P_Q \circ U_Q \circ Z^{-1}) \circ (Z \circ U_Q \circ Z^{-1})^{-1} \\ &= (P_Q \circ U_Q \circ Z^{-1}) \circ (Z \circ U_Q^{-1} \circ Z^{-1}) \\ &= P_Q \circ Z^{-1} \end{aligned}$$

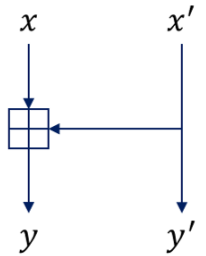
가 성립하여 $P_Q \circ Z^{-1}$ 역시 $\boxplus$의 음함수가 된다.

$P_Q(x, x', y)$가 Q의 자명한 음함수 $y \oplus Q(x, x')$라고 하면,

$$\begin{aligned} (P_Q \circ Z^{-1})(x, x', y) &= Q(x \oplus y,\ x' \oplus y,\ x \oplus x' \oplus y) \\ &= x \oplus x' \oplus y \oplus Q(x \oplus y,\ x' \oplus y) \end{aligned}$$

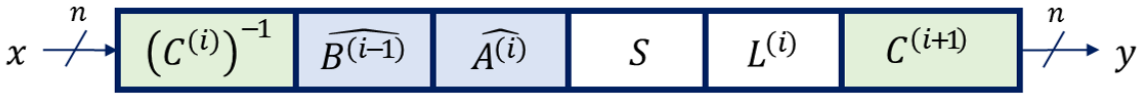
이로부터 다음과 같은 ARX cipher S-layer의 음함수 표현을 얻는다.

$$T(x, x', y, y') = (x \oplus x' \oplus y \oplus Q(x \oplus y,\ x' \oplus y),\ x' \oplus y')$$

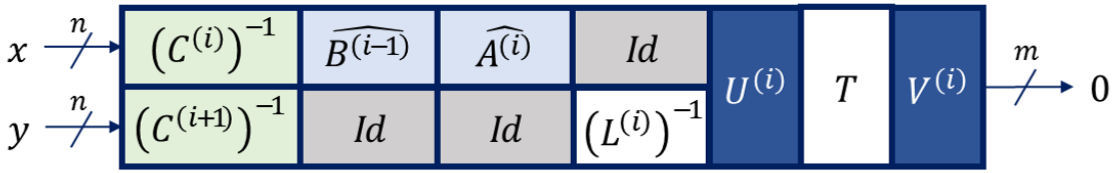


Implicit function of S-layer

$$\overline{E^{(i)}} = \left(C^{(i)}\right)^{-1} \circ \widehat{B^{(i-1)}} \circ \widehat{A^{(i)}} \circ E^{(i)} \circ C^{(i+1)}$$



$$P^{(i)} = V^{(i)} \circ T \circ U^{(i)} \circ \left(Id, \left(L^{(i)}\right)^{-1}\right) \circ \left(\widehat{A^{(i)}}, Id\right) \circ \left(\widehat{B^{(i-1)}}, Id\right) \circ \left(\left(C^{(i)}\right)^{-1}, \left(C^{(i+1)}\right)^{-1}\right)$$



감사합니다

Q&A

주 참고 문헌 : [4]

[1] Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi.
Cryptanalysis of a white box aes implementation.
In *Selected Areas in Cryptography: 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers 11*, pages 227–240. Springer, 2005.

[2] Alex Biryukov, Baptiste Lambin, and Aleksei Udovenko.
Cryptanalysis of arx-based white-box implementations.
IACR Transactions on Cryptographic Hardware and Embedded Systems, 2023(3):97–135, 2023.

[3] Stanley Chow, Philip Eisen, Harold Johnson, and Paul C Van Oorschot.
White-box cryptography and an aes implementation.
In *Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002 St. John's, Newfoundland, Canada, August 15–16, 2002 Revised Papers 9*, pages 250–270. Springer, 2003.

[4] Adrián Ranea, Joachim Vandersmissen, and Bart Preneel.
Implicit white-box implementations: White-boxing arx ciphers.
In *Annual International Cryptology Conference*, pages 33–63. Springer, 2022.

[5] Adrián Ranea and Bart Preneel.
On self-equivalence encodings in white-box implementations.
Cryptology ePrint Archive, Paper 2020/1325, 2020.

[6] Ernst Schulte-Geers.
On ccz-equivalence of addition mod 2^n.
Designs, Codes and Cryptography, 66:111–127, 2013.