An overview on cryptanalysis of ARX ciphers

S. Barbero

ARX operations

- n-bits strings: we essentially play in $\mathbb{F}_2^n$ and use the natural correspondence with $\mathbb{Z}_{2^n}$

$$x = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0) \leftrightarrow x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \cdots + x_1 2 + x_0$$

- Addition : $\boxplus$ is the modular addiction mod 2^n
- Rotation: $\lll_r$ and $\ggg_r$ respectively indicate a constant-distance left-rotation or right-rotation of r bits ($r < n$) of a n-bit word x (when will be clear from the context, we will also use the notations $\overleftarrow{x} = \lll_r$ $\overrightarrow{x} = x \ggg_r$)

$$x \lll_r = (x_{n-r-1}, x_{n-r-2}, \ldots, x_1, x_0, x_{n-1}, \ldots x_{n-r})$$

$$x \ggg_r = (x_{r-1}, \ldots, x_1, x_0, x_{n-1}, x_{n-2}, \ldots, x_r)$$

- XOR: $\oplus$ is the bitwise addition (exclusive OR)

Why ARX?

ARX ciphers are block ciphers with very interesting advantages such as

- fast performance on PCs;
- compact implementation;
- easy algorithms;
- no timing attacks: in many other ciphers analyzing the time taken to execute cryptographic algorithms gives useful informations to the attacker in order to work backwards to the input, since the time of execution can differ based on the input;
- functionally completeness (assuming constants included): every possible logic gate can be realized as a network of gates using ARX operations and constants.

Basic disavantages of ARX

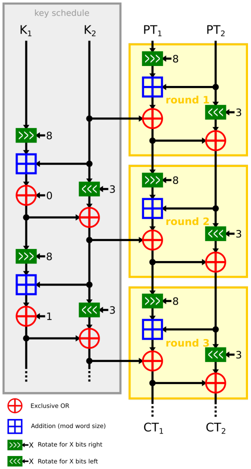
On the other hand we have some disavantages

- not best trade–off in hardware, although there are some different attempts of optimizations for various ARX ciphers;
- it is still not so clear which is their security against some cryptanalytic tools such as linear and differential cryptanalysis,
- it is also still not so clear which is their security against side channel attacks, i.e. attacks based on all hardware informations detected from their implementation (power attacks, electromagnetic attacks, fault attacks...)

Two examples of ARX ciphers: SPECK $2n/mn$

- SPECK is a family of lightweight ARX block ciphers publicly released by the National Security Agency (NSA) in June 2013
- every element of the SPECK family is indicated as SPECK $2n/mn$ where the size of every block is $2n$ with $n \in \{16, 24, 32, 48, 64\}$ and the key size is mn where $m \in \{2, 3, 4\}$ depending on the desired security
- the round function consists of two rotations, adding the right word to the left word, xoring the key into the left word, then xoring the left word into the right word.
- the number of rounds depends on the parameters selected and the key schedule uses the same round function as the main block cipher.

Two examples of ARX ciphers: SPECK $2n/mn$



Three rounds of SPECK

Two examples of ARX ciphers: SPECK $2n/mn$

Round function

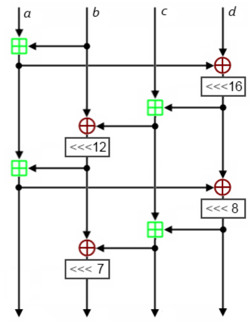
When $i \geq 0$, (x_0, y_0) and $K = (l_{m-2}, \dots, l_0, k_0)$ are respectively the round number, a $2n$ bits plaintext and a $2nm$ bits master key, we have the i–th round output (x_{i+1}, y_{i+1}) from the input (x_i, y_i) and the $i+1$-round key (l_{i+m-1}, k_{i+1}) from (l_{i+m-2}, k_i) as follows

$$\boxed{x_{i+1} = ((x_i \ggg_\alpha) \boxplus y_i) \oplus k_i \quad y_{i+1} = (y_i \lll_\beta) \oplus x_{i+1}}$$

$$\boxed{l_{i+m-1} = ((l_i \ggg_\alpha) \boxplus k_i) \oplus i, \quad k_{i+1} = (k_i \lll_\beta) \oplus l_{i+m-1}}$$

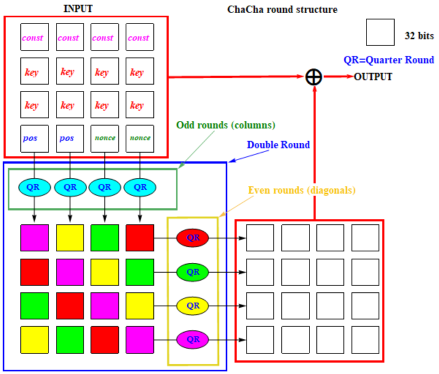
where $(\alpha, \beta) = (7, 2)$ for $n = 16$ and $(\alpha, \beta) = (8, 3)$ for the larger versions. The key schedule uses the same round function to generate the next round key.

Two examples of ARX ciphers:Chacha



The Chacha quarter round function

Two examples of ARX ciphers:Chacha



The Chacha structure

Two examples of ARX ciphers:Chacha

- the input words are placed in an initial matrix

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & t_1 & v_0 & v_1 \end{pmatrix}$$

  - (c_0, c_1, c_2, c_3) are predefined constants
  - $k = (k_0, k_1, ..., k_7)$ is a 256–bit key
  - (v_0, v_1) is a 64–bit nonce,
  - the *i*-th block is the output of the Chacha function, that takes as input the key, the nonce, and a 64-bit counter $t = (t_0, t_1)$ corresponding to the integer i

Two examples of ARX ciphers:Chacha

- A keystream block Z is then defined as

$$Z = X + X^r$$

where "$+$" symbolizes wordwise integer addition, and where $X^r = Round_r(X)$ and r is the round number;

- *Round* is the round function based on the following nonlinear operation (also called the *quarterround function*) $(x_0, x_1, x_2, x_3) \rightarrow (z_0, z_1, z_2, z_3)$ by sequentially computing

$$\begin{aligned} b_0 &= x_0 \boxplus x_1, \quad & b_3 &= (x_3 \oplus b_0) \lll_{16} \\ b_2 &= x_2 \boxplus b_3, \quad & b_1 &= (x1 \oplus b_2) \lll_{12} \\ z_0 &= b_0 \boxplus b_1, \quad & z_3 &= (b_3 \oplus z_0) \lll_{8} \\ z_2 &= b_2 \boxplus z_3, \quad & z_1 &= (b_1 \oplus z_2) \lll_{7} \end{aligned}$$

Two examples of ARX ciphers:Chacha

- in odd numbers of rounds, the nonlinear operation is applied to the columns $(x_0, x_4, x_8, x_{12}), (x_1, x_5, x_9, x_{13}), (x_2, x_6, x_{10}, x_{14}), (x_3, x_7, x_{11}, x_{15})$
- in even numbers of rounds, the nonlinear operation is applied to the diagonals $(x_0, x_5, x_{10}, x_{15}), (x_1, x_6, x_{11}, x_{12}), (x_2, x_7, x_8, x_{13}), (x_3, x_4, x_9, x_{14}),$

Algebraic cryptanalysis

- Deterministic key-recovery attack based on solving a system of equations that represents the encryption (or decryption) function.
- The set of indeterminates of the system consists of variables that represent a plaintext as well as a ciphertext, the internal states of the cipher, and the key used for encryption/decryption.
- Given a plaintext and its corresponding ciphertext, one recovers the key by solving the system of equations.

Algebraic cryptanalysis

- A common approach to mount algebraic attack is to construct the system of equations with coefficients in the binary field $\mathbb{F}_2$.
- In the context of ARX ciphers, bitwise XOR corresponds to addition in $\mathbb{F}_2$ and bitwise rotation is a linear operation over $\mathbb{F}_2$.
- The only nonlinear function over $\mathbb{F}_2$ in ARX ciphers is modular addition.
- In terms of coordinate-wise operation, the computation of the modular addition involves a carry bit. The value of the *i*-th carry bit depends on the $(i-1)$-th input bits and the $(i-1)$-th carry bit.

Algebraic cryptanalysis

Definition

A 3-variable Boolean majority function f_{maj} is defined as

$$f_{\mathrm{maj}}(x, y, z) = \begin{cases} 1 & \mathrm{wt}(x, y, z) \geq 2 \\ 0 & \text{otherwise.} \end{cases}$$

Equivalently, the algebraic normal form of f_{maj} is equal to $f_{\mathrm{maj}}(x, y, z) = xy + xz + yz$.

Algebraic cryptanalysis

Proposition

Let n be a positive integer and let $\varphi : \mathbb{F}_2^n \mapsto \mathbb{Z}_{2^n}$ be the function $\varphi(x) = \sum_{i=0}^{n-1} x_i \cdot 2^i$. where x_i is the value of the i-th coordinate of x. For any $x, y, z \in \mathbb{F}_2^n$ such that $\varphi(z) = \varphi(x) \boxplus \varphi(y)$, we have

$$z_i = x_i + y_i + c_i, \qquad 0 \leq i \leq n-1$$

where $c_0 = 0$ and $c_j = f_{\mathrm{maj}}(x_{j-1}, y_{j-1}, c_{j-1})$ for $j = 1, 2, \ldots, n-1$.

Remark

The degree of the coordinate function in modular addition increases proportionally to the significance of the bit of z.

Linear Cryptanalysis

- Introduced by Matsui in order to attack block ciphers (DES, FEAL) and find an "effective" linear expression which describes a given cipher algorithm obtaining one or more linear relations among the parities of plaintext,ciphertext and the secret key.
- Let P, C and K denote the plaintext, the ciphertext and the key respectively

$$P[i_1, i_2, \dots, i_a] \oplus C[j_1, j_2, \dots, j_b] = K[k_1, k_2, \dots, k_c]$$

  where $i_1, i_2, \dots, i_a, j_1, j_2, \dots, j_b, k_1, k_2, \dots, k_c$ are fixed bit locations.
- the previous equality must hold with a probability $p \neq \frac{1}{2}$ for a randomly given plaintext P and the corresponding ciphertext C, with effectiveness $|p - \frac{1}{2}|$.

Linear Cryptanalysis

Once we succeed in obtaining this equality, we are able to determine one key bit $K[k_1, k_2, \dots, k_c]$ with the following algorithm based on the maximum likelihood method:

- evaluate the number T of plaintext such that the left side of this equation is equal to 0;
- if N is the number of plaintext and $T > \frac{N}{2}$ then guess $K[k_1, k_2, \dots, k_c] = 0$ or $K[k_1, k_2, \dots, k_c] = 1$ respectively when $p > \frac{1}{2}$ or $p < \frac{1}{2}$, otherwise guess $K[k_1, k_2, \dots, k_c] = 1$ or $K[k_1, k_2, \dots, k_c] = 0$ respectively when $p > \frac{1}{2}$ or $p < \frac{1}{2}$.

Linear Cryptanalysis

Clearly, the success rate of this method increases when N or $|p - \frac{1}{2}|$ does. Thus the fundamental purposes in order to apply this attack are:

- find effective linear expressions;
- obtain an explicit description of the success rate by N and p;
- search best espressions and evaluate the best probabilities, i.e. those expressions having a probability p which gives the maximum value of $|p - \frac{1}{2}|$.

Linear Cryptanalysis

- An useful concept related to linear cryptanalysis is the idea of *correlation*.
- We define the inner product for two n–bit boolean vectors x and y as $x \cdot y = \bigoplus_{i=0}^{n-1} x_i y_i$,
- Considering a vectorial boolean function $f : \mathbb{F}_{2n} \to \mathbb{F}_{2m}$ if Γ_{in} and Γ_{out} are the masks for input x and output $f(x)$ respectively, we can define the correlation of the linear approximation as

$$Cor(\Gamma_{in}, \Gamma_{out}) = 2 \cdot \mathbb{P}(\Gamma_{in} \cdot x \oplus \Gamma_{out} \cdot f(x) = 0) - 1.$$

Linear Cryptanalysis

- The correlation gives a clear measure of "affinity" between the parities of plaintext and ciphertext and also can be evaluated for more rounds obtained by the composition of r round functions f_i.
- If we consider the iterated permutation $g = f_{r-1} \circ \cdots \circ f_1 \circ f_0$, we have that a linear approximations $(\gamma_i, \cdots, \gamma_{i+1})$ of a single round f_i can be concatenated into a linear trail $(\gamma_0, \gamma_1, \cdots, \gamma_r)$ of g
- The correlation of the linear trail can be calculated as

$$Cor(\gamma_0, \gamma_r) = \prod_{i=0}^{r-1} Cor(\gamma_i, \gamma_{i+1}).$$

- Correlation is also used to evaluate the expected value of the data complexity of a linear attack.

XOR-Differential cryptanalysis

- Introduced by Biham and Shamir with the basic idea of exploiting pairs of plaintext with certain differences yielding to other certain differences in the corresponding ciphertexts (or internal states of the cipher) which have a non–uniform distribution.
- Given a n–bit input and corresponding output strings $X = [X_1, X_2, \ldots, X_n]$ and $Y = [Y_1, Y_2, \ldots, Y_n]$ and denote with (X', X'') a pair of input with corresponding pair of output (Y', Y''), the related input and output differerences are

$$\Delta X = X' \oplus X'' = [X'_1 \oplus X''_1, X'_2 \oplus X''_2, \ldots, X'_n \oplus X''_n],$$

$$\Delta Y = Y' \oplus Y'' = [Y'_1 \oplus Y''_1, Y'_2 \oplus Y''_2, \ldots, Y'_n \oplus Y''_n].$$

XOR-Differential cryptanalysis

- Differential cryptanalysis studies when, given a particular ΔX, a particular ΔY occurs with a probability distribution much different from the uniform one.
- Evaluate $\mathbb{P}[\Delta Y | \Delta X]$ and compare this value with the expected one from an uniform distribution.
- If $\mathbb{P}[\Delta Y | \Delta X]$ is greater or less than the uniform probability we call the pair $(\Delta X, \Delta Y)$ a *differential*.

XOR-Differential cryptanalysis

- Finding one or more differential can help to distinguish a ciphertext from randomness and to recover the (partial) key used in the cipher.
- Encrypt many pairs of chosen plaintexts having difference ΔX and try to decrypt the corresponding ciphertexts using all the possible subkeys to get the outputs (or internal states) Y.
- Checking the frequency that ΔY occurs, we can select with high probability the correct subkey, observing that this frequency of ΔY must be close to the conjectured value $\mathbb{P}[\Delta Y | \Delta X]$.

XOR-Differential cryptanalysis

- It is not always necessary to predict the full n–bit values of ΔY: a differential which only predicts parts of a n–bit value is a *truncated differential*
- There are some generalizations of differentials, e. g. the *higher order differentials* considering differences of differentials and *impossible differentials* which are differentials which occurs with low probability.

Additive-Differential cryptanalysis

- Introduced to study the difference of two outputs of an *ARX* operation taking in account also the effects of modular addiction.
- A standard ARX operation is defined as

$$ARX(a, b, d, r) = ((a \boxplus b) \lll r) \oplus d$$

where a, b, d are n–bit vectors

- Fixing the additive differences $\Delta\alpha$, $\Delta\beta$, $\Delta\lambda$ and $\Delta\mu$, define the difference Δe between two outputs of *ARX* as

$$\Delta e = ARX(a \boxplus \Delta\alpha, b \boxplus \Delta\beta, d \boxplus \Delta\lambda, r) \boxminus ARX(a, b, d, r),$$

- Additive differences pass through modular addition with probability one, thus we have $\Delta\gamma = \Delta\alpha \boxplus \Delta\beta$

Additive-Differential cryptanalysis

- we may define adp^{ARX} the additive differential probability of ARX as

$$adp^{ARX} = (\Delta\gamma, \Delta\lambda \xrightarrow{r} \Delta\mu) = \frac{|\{(a \boxplus b, d) : \Delta e = \Delta\mu\}|}{|\{(a \boxplus b, d)\}|}.$$

- An estimation of adp^{ARX} can be obtained as the product of the additive differential probabilities of rotation and XOR, : $adp^{\lll r}$ and $adp^{\oplus}$.

Additive-Differential cryptanalysis

- The additive differential of rotation and XOR are respectively defined as

$$\Delta R = ((a \boxplus \Delta\alpha) \lll_r) \boxminus (a \lll_r)$$

$$\Delta X = ((a \boxplus \Delta\alpha) \oplus (d \boxplus \Delta\beta)) \boxminus (a \oplus d)$$

  where $\Delta\alpha$ and $\Delta\beta$ are fixed differentials.
- There are only four possibilities for ΔR because $\Delta R \in \{(\Delta R_{u,v} = \Delta\alpha \lll_r) - u2^r + v, \quad u, v \in \{0,1\}\}$ we have

$$adp^{\lll}(\Delta\alpha \xrightarrow{r} \Delta R) = P(u,v),$$

Additive-Differential cryptanalysis

$$
\begin{aligned}
P(0,0) &= \mathbb{P}(\Delta\alpha \to \Delta R_{0,0}) = 2^{-n}(2^r - \Delta\alpha_L)(2^{n-r} - \Delta\alpha_R),\\
P(0,1) &= \mathbb{P}(\Delta\alpha \to \Delta R_{0,1}) = 2^{-n}(2^r - \Delta\alpha_L - 1)\Delta\alpha_R,\\
P(1,0) &= \mathbb{P}(\Delta\alpha \to \Delta R_{1,0}) = 2^{-n}\Delta\alpha_L(2^{n-r} - \Delta\alpha_R),\\
P(1,1) &= \mathbb{P}(\Delta\alpha \to \Delta R_{1,1}) = 2^{-n}(\Delta\alpha_L + 1)\Delta\alpha_R,
\end{aligned}
$$

$\Delta\alpha_L, \Delta\alpha_R$ are respectively the words composed of the r most significant and the $n-r$ least significant bits of $\Delta\alpha$.

Additive-Differential cryptanalysis

- On the other hand we define

$$adp^{\oplus}(\Delta\alpha, \Delta\beta \to \Delta\gamma) = \frac{|\{(c,d) : \Delta X = \Delta\gamma\}|}{|\{(c,d)\}|}.$$

- Therefore

$$adp^{ARX} \cong \sum_{i=1}^{4} (adp^{\lll}(\Delta\alpha \xrightarrow{r} \Delta\rho_i) adp^{\oplus}(\Delta\rho_i, \Delta\lambda \to \Delta\mu)),$$

where $\Delta\rho_i$, $i = 1, 2, 3, 4$ are the four possible output differences after the rotation.

Additive-Differential cryptanalysis

- This evaluation of adp^{ARX} would be accurate if the inputs to the rotation and to the XOR operation were independent, but a counterexample due to Velichov et al. shows that they are not.
- We need an alternative way in order to correctly find adp^{ARX} and this probability can be evaluated, for example, using S–functions

Rotational-Differential cryptanalysis

- Introduced in order to study the propagation of rotations throughout the encryption steps of an ARX scheme.
- X, Y be messages of n bits $\overleftarrow{X} = X \lll_1$ $\overrightarrow{X} = X \ggg_1$ and $\mathcal{S}$ an ARX scheme with q additions. The rotational cryptanalysis is based on the following properties:
  - $\overrightarrow{X \oplus Y} = \overrightarrow{X} \oplus \overrightarrow{Y}$
  - $\overrightarrow{X} \lll_r = \overrightarrow{X \lll_r}$
  - $p_r := Pr(\overrightarrow{X \boxplus Y} = \overrightarrow{X} \boxplus \overrightarrow{Y}) = \frac{1}{4}(1 + 2^{r-n} + 2^{-r} + 2^{-n})$
  - this probability is maximized to $2^{-1.415}$ when n is large and $r = 1$
  - $\mathcal{S}(\overrightarrow{X}) = \overrightarrow{\mathcal{S}(X)}$ with probability $(p_r)^q$
  - given a random function $\mathcal{P} : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, $\mathcal{P}(\overrightarrow{X}) = \overrightarrow{\mathcal{P}(X)}$ with probability 2^{-n}

Rotational-Differential cryptanalysis

Thus, we can detect nonrandomness in the ARX scheme $\mathcal{S}$ if $(p_r)^q > 2^{-n}$. For example, when $r = 1$, an ARX scheme implemented with less than $t/1.415$ additions is vulnerable to rotational cryptanalysis.
In general the attack procedure is

- Generate a random plaintext P and evaluate $C = \mathcal{S}_K(P)$, where K is the first secret key
- Evaluate $P' = \overleftarrow{P} \oplus d$, where d is the correction due to the presence of constants
- Evaluate $C' = \mathcal{S}_{K'}(P')$, with $K' = \overleftarrow{K} \otimes e$ the second key
- Check if (C, C') is a rotational pair

Rotational-XOR cryptanalysis with constants

Rotational-XOR difference

Combine rotational difference with XOR difference

$$(x, (x \lll \gamma) \oplus a)$$

$((a_1, a_2), \gamma)$-Rotational-XOR difference (RX-difference)

$$(x \oplus a_1, (x \lll \gamma) \oplus a_2)$$

equivalent to

$$(\tilde{x}, (\tilde{x} \lll \gamma) \oplus (a_1 \lll \gamma) \oplus a_2)$$

Rotational-XOR cryptanalysis with constants

Rotational-XOR difference through ARX

Rotation

$$x \xrightarrow{\lll \gamma} x \lll \gamma$$

$$\overleftarrow{x} \oplus a \xrightarrow{\lll \gamma} \overleftarrow{x \lll \gamma} \oplus (a \lll \gamma)$$

$$\Rightarrow ((0,a),1) \xrightarrow{\lll \gamma} ((0, a \lll \gamma), 1)$$

XOR

$$x, y \xrightarrow{\oplus} x \oplus y$$

$$\overleftarrow{x} \oplus a, \overleftarrow{y} \oplus b \xrightarrow{\oplus} \overleftarrow{x \oplus y} \oplus (a \oplus b)$$

$$\Rightarrow ((0,a),1), ((0,b),1) \xrightarrow{\oplus} ((0, a \oplus b), 1)$$

Rotational-XOR cryptanalysis with constants

We introduce some notations

- $x = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ a n-bit boolean vector,
- $SHL(x)$ an arithmetic left shift of x by one bit, $(I \oplus SHL)(x) = x \oplus SHL(x)$
- $x|y$ the vector bitwise OR operation,
- $x||y$ the concatenation of x and y
- $|x|$ the Hamming weight of a boolean vector x
- $L(x)$ the γ most significant bits of x
- $L'(x)$ the $n-\gamma$ most significant bits of x
- $R(x)$ the $n-\gamma$ least significant bits of x
- $R'(x)$ the γ least significant bits of x
- $x \preceq y$ holds if and only if $x_i \leq y_i$ for all $i = 0, \cdots, n-1$

Rotational-XOR cryptanalysis with constants

Theorem (T. Ashur, Y. Liu)

Let $x, y \in \mathbb{F}_{2^n}$ be independent random variables. Let $a_1, b_1, a_2, b_2, \Delta_1, \Delta_2$ be constants in $\mathbb{F}_{2^n}$ then

$$\mathbb{P}[\overleftarrow{(x \oplus a_1) \boxplus (y \oplus b_1) \oplus \Delta_1} = (\overleftarrow{x} \oplus a_2) \boxplus (\overleftarrow{y} \oplus b_2) \oplus \Delta_2]$$

is equal to $2^{-|SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3))|-3}$ if the following condition holds

$$(I \oplus SHL)(\delta_1 \oplus \delta_2 \oplus \delta_3) \oplus 1 \preceq SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3)),$$

otherwise is equal to $2^{-|SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3))|-1.415}$ if the following condition holds

$$(I \oplus SHL)(\delta_1 \oplus \delta_2 \oplus \delta_3) \preceq SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3))$$

where $\delta_1 = R(a_1) \oplus L'(a_2)$, $\delta_2 = R(b_1) \oplus L'(b_2)$ and $\delta_3 = R(\Delta_1) \oplus L'(\Delta_2)$.

Rotational-XOR cryptanalysis with constants

Lemma (E. Shulte–Geers)

Let $\zeta_1, \zeta_2, \zeta_3 \in \mathbb{F}_{2^n}$ *be constants. Let* $x, y \in \mathbb{F}_{2^n}$ *be independent random variables. Then*

$$\mathbb{P}[x \boxplus y = (x \oplus \zeta_1) \boxplus (y \oplus \zeta_2) \oplus \zeta_3] = 2^{-|SHL((\zeta_1 \oplus \zeta_3)|(\zeta_2 \oplus \zeta_3))|}$$

if the following condition holds

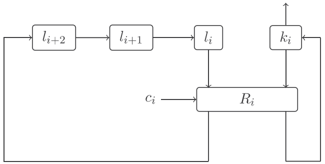
$$(I \oplus SHL)(\zeta_1 \oplus \zeta_2 \oplus \zeta_3) \preceq SHL((\zeta_1 \oplus \zeta_3)|(\zeta_2 \oplus \zeta_3))$$

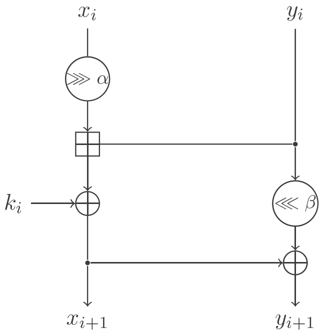
and $\mathbb{P}[x \boxplus y \boxplus 1 = (x \oplus \zeta_1) \boxplus (y \oplus \zeta_2) \oplus \zeta_3] = 2^{-|SHL((\zeta_1 \oplus \zeta_3)|(\zeta_2 \oplus \zeta_3))|}$ *if the following condition holds*

$$(I \oplus SHL)(\zeta_1 \oplus \zeta_2 \oplus \zeta_3) \oplus 1 \preceq SHL((\zeta_1 \oplus \zeta_3)|(\zeta_2 \oplus \zeta_3))$$

Rotational-XOR cryptanalysis with constants-SPECK32/64

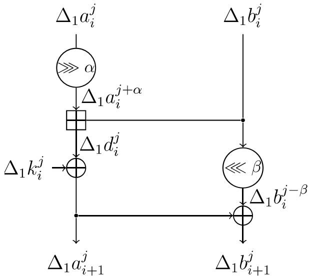
Application to SPECK32/64



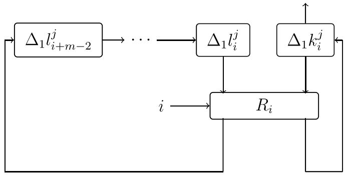


- Track RX-difference propagation in the key schedule
- Based on the good RX-trails found in the key schedule, track the propagation of RX-differences in the encryption

Rotational-XOR cryptanalysis with constants-SPECK32/64



(a) Notation of the bits of RX-differences in the round function of Speck.



(b) Notation of the bits of RX-differences in the key schedule of Speck.

Automated search for RX-Characteristics

- A possible path of RX-differences through different encryption rounds of a block cipher is called an *RX-characteristic*
- Using the notation

$$\Delta_1 x = x \oplus \overleftarrow{x}$$

the RX-differences in round i with $0 \leq i \leq r$, are

$$\Delta_1 a_i \ggg_\alpha, \Delta_1 b_i, \Delta_1 d_i$$

and for the key schedule

$$\Delta_1 l_i \ggg_\alpha, \Delta_1 k_i, \Delta_1 e_i, \Delta_1 c$$

where c depends on round number i.

Automated search for RX-Characteristics

- Search of valid r-round RX-characteristics for SPECK 32/64 with Automated Search (SAT-Solvers) finding when the boolean bits of RX-differences in round i with $0 \leq i \leq r$ satisfy one of the conditions of Theorem 1 plus the additional conditions for $0 \leq j < n$

$$\Delta_1 a_{i+1}^j = \Delta_1 d_i^j \oplus \Delta_1 k_i^j$$

$$\Delta_1 b_{i+1}^j = \Delta_1 b_i^{j-\beta} \oplus \Delta_1 a_{i+1}^j$$

- Similar situation concerning the key schedule in order to find a valid r-round RX-characteristic, with the additional conditions for $0 \leq j < n$

$$\Delta_1 l_{i+1}^j = \Delta_1 e_i^j \oplus \Delta_1 c$$

$$\Delta_1 k_{i+1}^j = \Delta_1 k_i^{j-\beta} \oplus \Delta_1 l_{i+1}^j$$

Automated search for RX-Characteristics

- Importance of software libraries as Arxpy to test theorical results.
- Some data from the paper of T. Ashur and Y.Liu

Application to SPECK32/64

An RX-characteristic in the keyschedule

Round	a_1	b_1	Δ_1	a_2	b_2	Δ_2	Predicted Prob.	Empirical Prob.	Accumulated Prob.
1	0	0	0	0	0	0	$2^{-1.415}$	$2^{-1.415}$	$2^{-1.415}$
2	0	0	0	0	0	0	$2^{-1.415}$	$2^{-1.415}$	$2^{-2.83}$
3	0	1	0	0	1	2	$2^{-2.415}$	$2^{-2.415}$	$2^{-5.245}$
4	0	2	6	0	0	8	$2^{-2.415}$	$2^{-2.415}$	$2^{-7.66}$
5	0	D	C4	0	B	78	$2^{-6.415}$	$2^{-6.415}$	$2^{-14.075}$
6	0	F4	0	1000	50	1088	$2^{-7.415}$	$2^{-7.415}$	$2^{-21.49}$
Total							$2^{-21.49}$		

Automated search for RX-Characteristics

Application to SPECK32/64

A corresponding RX-characteristic in the round function

Round	Input diff. (left,right)	Key diff.	Output diff. (left,right)	Predicted accumu. Prob.	Empirical accumu. Prob.
0	0, 0	0	0, 0	$2^{-1.415}$	$2^{-1.415}$
1	0, 0	0	0, 0	$2^{-2.83}$	$2^{-2.85}$
2	0, 0	3	3, 3	$2^{-4.245}$	$2^{-4.27}$
3	3, 3	4	607, 60B	$2^{-8.66}$	$2^{-8.68}$
4	607, 60B	11	40E, 1C22	$2^{-15.075}$	$2^{-15.01}$
5	40E, 1C22	1B8	3992, 491A	$2^{-21.49}$	$2^{-21.44}$
6	3992, 491A	1668	333F, 1756	$2^{-31.905}$	$2^{-31.6}$

All RX-differences are in hexadecimal notation.

Future works

- A deep comprehension of these results, in order to repeat the empirical data of the authors and consider the feasibility of a generalization to at least a toy version of Chacha, a lot of work in progress!

Essential bibliography

- Darkmatter report Survey on the Cryptanalysis of Arx Ciphers
- T.Ashur and Y. Liu *Rotational Cryptanalysis in the Presence of Constants* IACR Trans. Symmetric Cryptol. 2016 (1) 57–70, 2016
- E. Schulte–Geers *On CCZ–equivalence of addition* mod2^n Designs, Codes and Cryptography 66 (1–3), 111–127, 2013
- M. Matsui *Linear cryptanalysis method for DES ciphers* Advances in Cryptology-EUROCRYPT-1993, 386–397, Springer 1994.
- M. Daum *Cryptananlysis of Hash Functions of the MD4-Family* Ph.D. Thesis, Rhur Univesität Bochum,2005