

ARX-based Cryptography

Nicky Mouha

ESAT/COSIC, K.U.Leuven, Belgium
IBBT, Belgium

ECRYPT II Summer School, Albena
Friday, June 3, 2011

Outline

- 1 Introduction
- 2 Addition and XOR
- 3 Multiplication, Counting
- 4 ARX
- 5 Conclusion

ARX

- Addition (mod 2^n): $+$, \boxplus
- Rotation: $\lll r$
- XOR: \oplus

- Term 'AXR': Ralf-Philipp Weinmann (Dagstuhl 2009)
 - Later: renamed to ARX
- Concept of ARX is much older
 - E.g. FEAL (Eurocrypt 1987)

Advantages of ARX

- Fast performance on PCs
- Compact implementation
- Easy algorithm
- No timing attacks
- Functionally complete (assuming constant included)

Disadvantages of ARX

- Not best trade-off in hardware
- Security against linear and differential cryptanalysis?
- Security margin?
- Side-channel attacks?

ARX Designs

- Block ciphers
 - FEAL, Threefish
- Stream ciphers
 - Salsa20, ChaCha, HC-128
- Hash functions:
 - SHA-3 Finalists: BLAKE, Skein
 - SHA-3 Second Round: Blue Midnight Wish, Cubehash
 - SHA-3 First Round: EDON- \mathcal{R}

Designs Similar to ARX

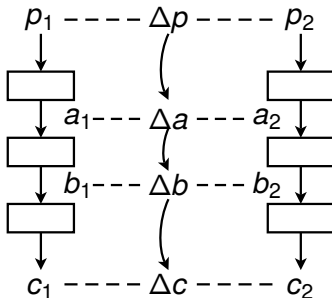
- Including left shift, right shift:
 - Block ciphers: TEA, XTEA, XXTEA
 - SHA-3 candidate: EnRUPT
- Including bitwise Boolean functions:
 - Hash functions: MD4, MD5, SHA-1
 - SHA-3 candidates: SIMD, Shabal

This presentation

- Introduce S-function concept
 - Can handle left/right shifts, bitwise Boolean functions, multiplication by constants
- Focus on differential cryptanalysis
- Analyze addition, XOR, and ARX components
- Provide observations on larger components

Differential Cryptanalysis

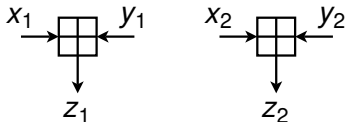
Differential characteristic: describes desired propagation of differences through cryptographic primitive



S-box vs ARX

- S-box
 - Typical size up to 8×8 bit
 - Difference distribution table: up to $2^{16} = 65536$ elements
 - Easy to calculate: differential probability, number of output differences, output difference with highest probability,...
- ARX operations
 - Typically, $n = 32$ or $n = 64$
 - Difference distribution table: 2^{64} or 2^{128} elements, too large!
 - Fast algorithms ($\mathcal{O}(n)$) required to calculate properties

xdp⁺: The XOR Differential Probability of Addition



$\Delta x, \Delta y, \Delta z$ are fixed xor differences such that

$$x_2 = x_1 \oplus \Delta x, \quad y_2 = y_1 \oplus \Delta y, \quad z_2 = z_1 \oplus \Delta z,$$

xdp⁺ expresses the fraction of pairs (x_1, y_1) for which the following holds:

$$((x_1 \oplus \Delta x) + (y_1 \oplus \Delta y)) \oplus (x_1 + y_1) = \Delta z.$$

xdp⁺: Motivating Example

From “On the Additive Differential Probability of Exclusive-Or”,
 Lipmaa, Wallén, Dumas, FSE 2004:

$$\begin{aligned} \text{xdp}^+(1\mathbf{1}100, 0\mathbf{0}110 \rightarrow 1\mathbf{0}110) \\ = LA_{101}A_{\mathbf{1}00}A_{111}A_{011}A_{000}C = \frac{1}{4} \end{aligned}$$

where

$$A_{000} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_{001} = A_{010} = A_{100} = \frac{1}{2} \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},$$

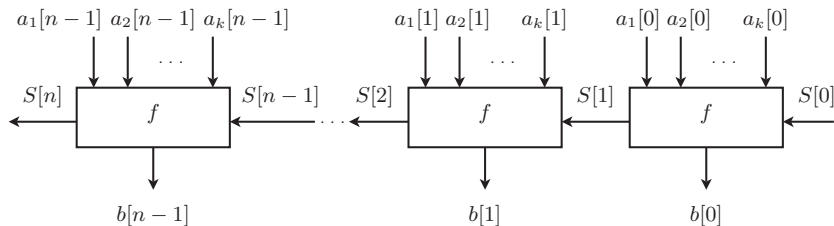
$$A_{011} = A_{101} = A_{110} = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad A_{111} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$L = [1 \quad 1], \quad C = [1 \quad 0]^T.$$

S-function

An S-function accepts n -bit words a_1, a_2, \dots, a_k and an n -digit input state S , and produces an n -bit output word b :

$$(b[i], S[i + 1]) = f(a_1[i], a_2[i], \dots, a_k[i], S[i]), \quad 0 \leq i < n .$$



xdp⁺: From Words to Bits: Constructing f

$$\left\{ \begin{array}{l} x_2 \leftarrow x_1 \oplus \Delta x \\ y_2 \leftarrow y_1 \oplus \Delta y \\ z_1 \leftarrow x_1 + y_1 \\ z_2 \leftarrow x_2 + y_2 \\ \Delta z \leftarrow z_2 \oplus z_1 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} x_2[i] \leftarrow x_1[i] \oplus \Delta x[i] \\ y_2[i] \leftarrow y_1[i] \oplus \Delta y[i] \\ z_1[i] \leftarrow x_1[i] \oplus y_1[i] \oplus c_1[i] \\ c_1[i+1] \leftarrow (x_1[i] + y_1[i] + c_1[i]) \ggg 1 \\ z_2[i] \leftarrow x_2[i] \oplus y_2[i] \oplus c_2[i] \\ c_2[i+1] \leftarrow (x_2[i] + y_2[i] + c_2[i]) \ggg 1 \\ \Delta z[i] \leftarrow z_2[i] \oplus z_1[i] \end{array} \right.$$

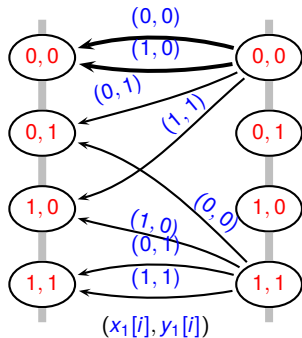
xdp^+ : From Words to Bits: S-function

The S-function for xdp^+ is:

$$\begin{aligned}(\Delta z[i], \mathbf{S}[i+1]) &= f(x_1[i], y_1[i], \Delta x[i], \Delta y[i], \mathbf{S}[i]), & 0 \leq i < n, \\ \mathbf{S}[i] &\leftarrow (c_1[i], c_2[i]), \\ \mathbf{S}[i+1] &\leftarrow (c_1[i+1], c_2[i+1]).\end{aligned}$$

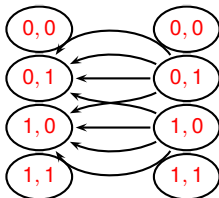
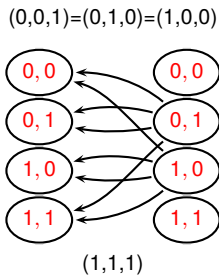
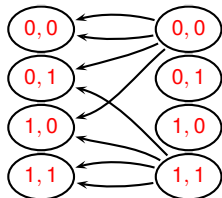
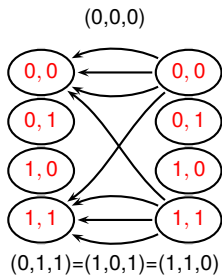
xdp^+ : Subgraph

$$(\Delta x[i], \Delta y[i], \Delta z[i]) = (1, 0, 1)$$



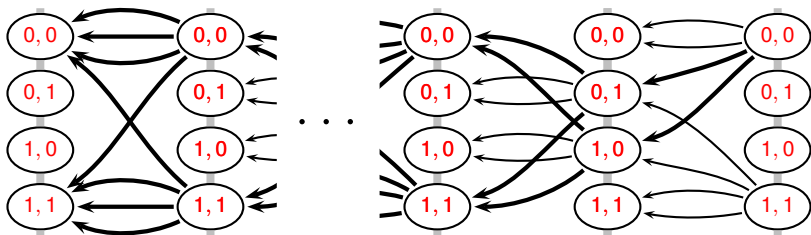
$$\left\{ \begin{array}{l} x_2[i] \leftarrow x_1[i] \oplus \Delta x[i] \\ y_2[i] \leftarrow y_1[i] \oplus \Delta y[i] \\ z_1[i] \leftarrow x_1[i] \oplus y_1[i] \oplus c_1[i] \\ c_1[i+1] \leftarrow (x_1[i] + y_1[i] + c_1[i]) \ggg 1 \\ z_2[i] \leftarrow x_2[i] \oplus y_2[i] \oplus c_2[i] \\ c_2[i+1] \leftarrow (x_2[i] + y_2[i] + c_2[i]) \ggg 1 \\ \Delta z[i] \leftarrow z_2[i] \oplus z_1[i] \end{array} \right.$$

xdp^+ : All Subgraphs



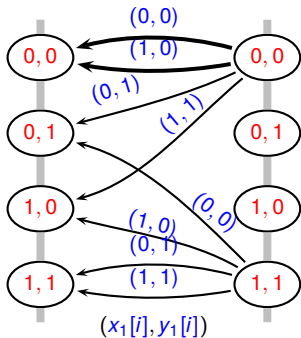
xdp^+ : From Graphs to Probability

Computing probability xdp^+ is equivalent to counting number of paths that satisfy $\Delta x, \Delta y, \Delta z$. Each valid pair (x_1, y_1) corresponds to path in graph (shown in bold).



xdp^+ : From Subgraph to Matrix

$$(\Delta x[i], \Delta y[i], \Delta z[i]) = (1, 0, 1)$$



$$S[i+1] \begin{matrix} (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{matrix}$$

$$S[i] \begin{matrix} (0,0), (0,1), (1,0), (1,1) \end{matrix}$$

$$\frac{1}{4} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix} = A_{101}$$

xdp^+ : All Matrices

There are four distinct matrices for xdp^+ :

$$A_{000}, A_{001} = A_{010} = A_{100}, A_{011} = A_{101} = A_{110}, A_{111}.$$

$$A_{000} = \frac{1}{4} \begin{bmatrix} 3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}, \quad A_{001} = \frac{1}{4} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$

$$A_{011} = \frac{1}{4} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad A_{111} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

xdp^+ : From Matrices to Probability

Computing the probability xdp^+ can be done using matrix multiplications

$$\text{xdp}^+(\Delta x, \Delta y \rightarrow \Delta z) = L A_{w[n-1]} \cdots A_{w[1]} A_{w[0]} C .$$

where

$$w[i] = \Delta x[i] \parallel \Delta y[i] \parallel \Delta z[i], \quad 0 \leq i < n,$$

$$L = [1 \quad 1 \quad \cdots \quad 1],$$

$$C = [1 \quad 0 \quad \cdots \quad 0]^T .$$

xdp⁺: Minimized Matrices

Reduce size of matrices by combining equivalent states
(FSM reduction algorithm):

$$A'_{000} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, A'_{001} = \frac{1}{2} \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},$$
$$A'_{011} = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, A'_{111} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Linearization

- How to find good differential characteristics for ARX?
- Very powerful technique: linearization!
- In case of ARX: replace addition by XOR, then find low-weight codewords
- Easy to prove: $\text{xdp}^+(\alpha, \beta \rightarrow \alpha \oplus \beta) > 0$

EDON- \mathcal{R}

- Hash function by Gligoroski et al., submission to SHA-3
- Here: analysis together with Bjørstad, unpublished

$$\begin{array}{l}
 T_0 \leftarrow (0x55555555 + Y_0 + Y_1 + Y_2 + Y_5 + Y_7) \ggg 0 \\
 T_1 \leftarrow (Y_0 + Y_1 + Y_3 + Y_4 + Y_6) \ggg 5 \\
 T_2 \leftarrow (Y_0 + Y_1 + Y_2 + Y_3 + Y_5) \ggg 9 \\
 T_3 \leftarrow (Y_2 + Y_3 + Y_4 + Y_6 + Y_7) \ggg 11 \\
 T_4 \leftarrow (Y_0 + Y_1 + Y_3 + Y_4 + Y_5) \ggg 15 \\
 T_5 \leftarrow (Y_2 + Y_4 + Y_5 + Y_6 + Y_7) \ggg 20 \\
 T_6 \leftarrow (Y_1 + Y_2 + Y_5 + Y_6 + Y_7) \ggg 25 \\
 T_7 \leftarrow (Y_0 + Y_3 + Y_4 + Y_6 + Y_7) \ggg 27
 \end{array}$$

EDON- \mathcal{R}

- Introduce XOR difference in bit i (i is not MSB)

$$\begin{array}{rcl}
 T_0 & \leftarrow & (0x55555555 + Y_0 + Y_1 + Y_2 + Y_5 + Y_7) \ggg 0 \\
 T_1 & \leftarrow & (Y_0 + Y_1 + Y_3 + Y_4 + Y_6) \ggg 5 \\
 T_2 & \leftarrow & (Y_0 + Y_1 + Y_2 + Y_3 + Y_5) \ggg 9 \\
 T_3 & \leftarrow & (Y_2 + Y_3 + Y_4 + Y_6 + Y_7) \ggg 11 \\
 T_4 & \leftarrow & (Y_0 + Y_1 + Y_3 + Y_4 + Y_5) \ggg 15 \\
 T_5 & \leftarrow & (Y_2 + Y_4 + Y_5 + Y_6 + Y_7) \ggg 20 \\
 T_6 & \leftarrow & (Y_1 + Y_2 + Y_5 + Y_6 + Y_7) \ggg 25 \\
 T_7 & \leftarrow & (Y_0 + Y_3 + Y_4 + Y_6 + Y_7) \ggg 27
 \end{array}$$

EDON- \mathcal{R}

For a pair (a_1, a_2) :

$$\Delta^{\pm}u[k] : \begin{cases} a_1[i] = 1, a_2[i] = 0, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

$$\Delta^{\pm}n[k] : \begin{cases} a_1[i] = 0, a_2[i] = 1, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

EDON- \mathcal{R} example:

$$\begin{aligned} T_0 &= (Y_1 + Y_7 + \dots) \ggg 0 \\ T_1 &= (Y_1 + Y_4 + \dots) \ggg 5 \\ T_3 &= (Y_4 + Y_7 + \dots) \ggg 11 \end{aligned}$$

EDON- \mathcal{R}

For a pair (a_1, a_2) :

$$\Delta^{\pm}u[k] : \begin{cases} a_1[i] = 1, a_2[i] = 0, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

$$\Delta^{\pm}n[k] : \begin{cases} a_1[i] = 0, a_2[i] = 1, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

EDON- \mathcal{R} example:

$$\begin{aligned} 0 &= (u[k] + Y_7 + \dots) \ggg 0 \\ 0 &= (u[k] + Y_4 + \dots) \ggg 5 \\ 0 &= (Y_4 + Y_7 + \dots) \ggg 11 \end{aligned}$$

EDON- \mathcal{R}

For a pair (a_1, a_2) :

$$\Delta^{\pm}u[k] : \begin{cases} a_1[i] = 1, a_2[i] = 0, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

$$\Delta^{\pm}n[k] : \begin{cases} a_1[i] = 0, a_2[i] = 1, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

EDON- \mathcal{R} example:

$$\begin{aligned} 0 &= (u[k] + n[k] + \dots) \ggg 0 \\ 0 &= (u[k] + Y_4 + \dots) \ggg 5 \\ 0 &= (Y_4 + n[k] + \dots) \ggg 11 \end{aligned}$$

EDON- \mathcal{R}

For a pair (a_1, a_2) :

$$\Delta^{\pm}u[k] : \begin{cases} a_1[i] = 1, a_2[i] = 0, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

$$\Delta^{\pm}n[k] : \begin{cases} a_1[i] = 0, a_2[i] = 1, & \text{if } i = k \text{ ,} \\ a_1[i] = a_2[i], & \text{for } 0 \leq i < n, i \neq k \text{ .} \end{cases}$$

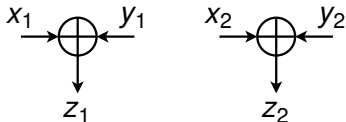
EDON- \mathcal{R} example:

$$\begin{aligned} 0 &= (u[k] + n[k] + \dots) \ggg 0 \\ 0 &= (u[k] + n[k] + \dots) \ggg 5 \\ 0 &\neq (n[k] + n[k] + \dots) \ggg 11 \end{aligned}$$

Linearization

- “Finding SHA-1 Characteristics: General Results and Applications”, De Cannière, Christian Rechberger, ASIACRYPY 2006
 - 64-step characteristic for SHA-1, no solution

adp^\oplus : The Additive Differential Probability of XOR



$\Delta x, \Delta y, \Delta z$ are fixed additive differences such that

$$x_2 = x_1 + \Delta x, \quad y_2 = y_1 + \Delta y, \quad z_2 = z_1 + \Delta z,$$

adp^\oplus expresses the fraction of pairs (x_1, y_1) for which the following holds:

$$(x_1 + \Delta x) \oplus ((y_1 + \Delta y) - (x_1 \oplus y_1)) = \Delta z.$$

adp[⊕]: Matrices and Probability

In a way similar to xdp⁺, we obtain 8 matrices for adp[⊕].

$$A_{101} = \frac{1}{4} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} .$$

The probability adp[⊕] is computed again as:

$$\text{adp}^{\oplus}(\Delta x, \Delta y \rightarrow \Delta z) = L A_{w[n-1]} \cdots A_{w[1]} A_{w[0]} C .$$

$\text{xdp}^{\times 3}$: Multiplication by 3

- Multiplication by constant: $\text{xdp}^{\times C}$
 - Hash functions Shabal ($\times 3$, $\times 5$), EnRUPT ($\times 9$)
- Let $\alpha = 0\text{x}12492489$ and $\gamma = 0\text{x}3\text{AEBAEAB}$
- Approximation using xdp^+ :

$$\text{xdp}^+(\alpha, \alpha \ll 1 \rightarrow \gamma) = 2^{-25}$$

- Correct probability:

$$\text{xdp}^{\times 3}(\alpha \rightarrow \gamma) = 2^{-15}$$

$\text{xdp}^{\times 3}$: All Matrices

After minimization algorithm: 16×16 matrices reduced to 4×4 :

$$A_{00} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad A_{01} = \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
$$A_{10} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad A_{11} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix}.$$

xdc^+ : # of Possible XOR Differentials of Addition

- xdc^+ counts number of *possible* output differences, when input differences are given
- Start with minimized matrices for xdp^+
- Apply subset construction (automata theory)

$$\text{xdc}^+(\Delta x, \Delta y) = LB_{w[n-1]} \cdots B_{w[1]} B_{w[0]} C ,$$

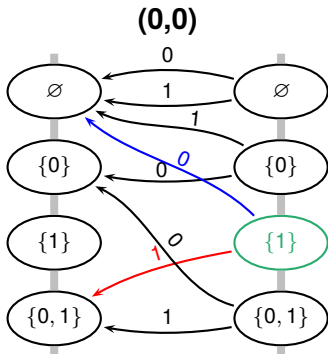
where

$$w[i] = \Delta x[i] \parallel \Delta y[i], \quad 0 \leq i < n ,$$

$$L = [1 \quad 1 \quad \cdots \quad 1] ,$$

$$C = [1 \quad 0 \quad \cdots \quad 0]^T .$$

x_{dc}^+ : All Possible XOR Output Differences

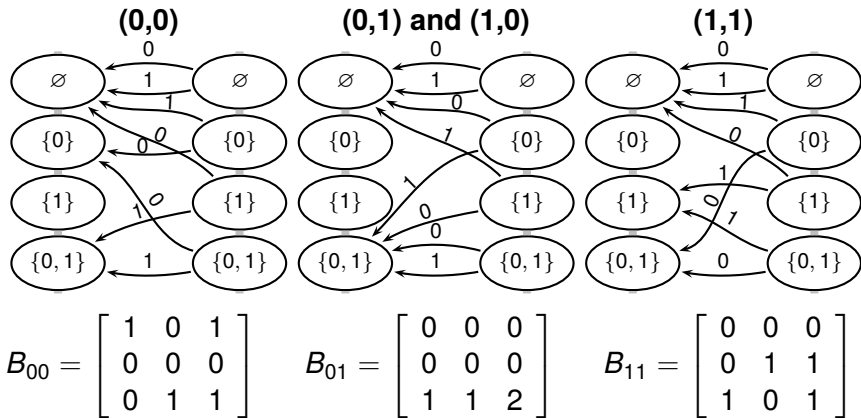


$$B_{00} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

$$A'_{000} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$A'_{001} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

xdc^+ : Graphs



Cryptanalysis of Hash Function Skein

- Aumasson et al. (ASIACRYPT 2009)
 - $\mathcal{O}(2^n)$ time algorithm for xdc^+
- Mouha et al. (SAC 2010)
 - $\mathcal{O}(n)$ time algorithm for xdc^+

$$\begin{aligned} & \text{xdc}^+(0 \times 1000010402000000, 0 \times 000000000000000000) \\ &= L \cdot B_{00}^3 \cdot B_{10} \cdot B_{00}^{19} \cdot B_{10} \cdot B_{00}^5 \cdot B_{10} \cdot B_{00}^8 \cdot B_{10} \cdot B_{00}^{25} \cdot C \\ &= 5880 \end{aligned}$$

Toolkit Available

- No need to re-implement!
- Toolkit can perform all calculations in this presentation
- Can also efficiently find maximum probability output differences (paper currently being written)

<http://www.ecrypt.eu.org/tools>

Ongoing Work

- Analyzing ARX as a single component – sufficient to analyze a cipher?
- Ongoing works shows not...
- Often many characteristics for same differential
- Then: Probability of differential \neq Probability of characteristic

Conclusion

- ARX: Addition, Rotation, XOR
- Fast in software, increasingly used in designs
- But: security analysis seems difficult
- We need:
 - More analysis
 - Toolkits: avoid reinventing the wheel
 - Strategy for secure design