Confusion and Diffusion in Recent Ultralightweight RFID Authentication Protocols

P. D'Arco and R. De Prisco

Dipartimento di Informatica

Università di Salerno

ITALY

Talk Outline

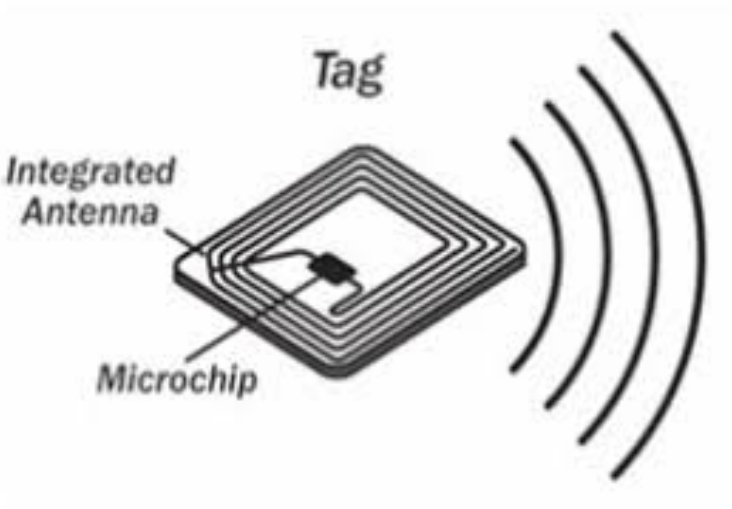
- **Introduction**
  - Rfid basics
  - Ultralightweight authentication protocols: structure
- **Transforms**
  - Pseudo-Kasami, Recursive Hash, Conversion, FCS-based
  - Design Weaknesses
- **Impersonation attacks**
  - KMAP
  - RCIA
  - SASI$^+$
  - SLAP
  - FCS-Based Protocol
- **Conclusions and open problems**

INTRODUCTION

RFID basics

- **R**adio **F**requency **Id**entification
  - ✓ Uses radio-frequency waves for communication between a reader and a tag
- Tags
  - ✓ Microchip labels attached to (movable) objects
    - ✧ Upon receiving energy (waves) emit a signal to communicate with the reader
- Reader
  - ✓ A device that interacts with tags

RFID Tags



- Active
  - ✓ Contains a battery, stronger signal, bigger communication range (~30*m*)
- **Passive**
  - ✓ No battery, gets the power from the reader, smaller communication range (~10*m*)

- **Computationally very limited**

RFID Readers

- Device that can “talk” with the Tags
  - ✓ Various types, mobile, or in fixed positions
  - ✓ More “powerful”

RFID authentication protocols

- Four categories of protocols
  - ✓ Full-fledged
    - ✧ Standard cryptographic operations
  - ✓ Simple
    - ✧ Can use hash function
  - ✓ Lightweight
    - ✧ Can generate random numbers
  - ✓ **Ultralightweight**
    - ✧ Only simple bitwise operations (e.g. AND, OR, XOR, Shift)

Ultralightweight protocols

- Challenging task
  - ✓ Often protocols provided without a robust security analysis
  - ✓ Broken soon after their presentations

- Example
  - ✓ SASI cryptoanalysis (e.g., [DD10]) and beyond
  - ✓ A full guide to common pitfalls available [ACH15]

Ultralightweight 2.0

- New feature which almost all of them exhibit
  - ✓ more ***involved transforms*** on the data stored in the tag memory
- Still informal security analyses
  - ✓ since the transforms are complex, **only the legal parts** who share the secret keys can produce the correct messages required by the authentication protocol
  - ✓ **no adversarial** entity, without the secret keys, can be successful with non negligible probability

A number of recent proposals

- We concentrated our attention on 5 protocols
  - ✓ KMAP
    - U. Mujahid, M. Najam-ul-Islam, S. Sarwar. A New Ultrlightweight RFID Authentication Protocol for Passive Low Cost Tags: KMAP. Wireless Personal Communication, Springer, 2016.
  - ✓ RCIA
    - U. Mujahid, M. Najam-ul-Islam, M. Ali Shami. RCIA: A New Ultralightweight RFID Authentication Protocol Using Recursive Hash. International Journal of Distributed Sensor Networks, Hindawi, 2015.
  - ✓ SASI+
    - U. Mujahid, M. Najam-ul-Islam, A. Raza Jafri, Qurat-ulAin, M. Ali Shami. A New Ultralightweight RFID Mutual Authentication Protocol: SASI Using Recursive Hash. International Journal of Distributed Sensor Networks, Hindawi, 2016.
  - ✓ SLAP
    - H. Luo, G. Wen, J. Su, Z. Huang. SLAP: Succint and Lightweight Authentication Protocol for Low-Cost RFID System. Wireless Netw, DOI 10.1007/s11276-016-1323-y, Springer, 2016.
  - ✓ FCS-Based
    - B. Mustapha, M. Djeddou and K. Drouiche. An ultralightweight RFID authentication protocol based on Feistel cipher structure. Security and Communication Networks, John Wiley & son, 2017

Protocol structure

- They have essentially the same general structure

Reader	Message	Tag
	Hello →	
	← IDS	
	A || B || C →	
	← D	

Core tools

- Transforms
  - ✓ Pseudo-Kasami code
  - ✓ Recursive hash
  - ✓ Conversion
  - ✓ FCS-Based
- We analyze them
  - ✓ show several weaknesses

TRANSFORMS

pseudo-Kasami code

pseudo-Kasami code

- $x=x_1,x_2,\ldots,x_n$ a string of bit
- seed s: an integer, $1 \leq s \leq n$
- pseudo-Kasami code of x is:
  - ✓ y=CRshift(x,s)
  - ✓ pKc(x,s)=$x \oplus y$
  - ✓ Example n=24, s=6

x =	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0
y =	0	1	0	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1	1	0	0
$pKc(x,s)$=	1	1	0	0	1	1	0	1	0	0	0	0	0	1	1	0	0	1	1	1	1	0	1	0

pseudo-Kasami code: weaknesses

Lemma 2.1 *Let $x = x_1, \ldots, x_n$ be a string of n bits, and let s be a seed for the pseudo-Kasami code $\mathtt{pKc}(x, s)$ such that n is a multiple of s. Let x' be a new string obtained from x by flipping n/s bits, all at distance s from each other. Then $\mathtt{pKc}(x, s) = \mathtt{pKc}(x', s)$.*

Proof:

Since y=CRshift(x,s) flippings cancel themselves out in the xor.

	← s →						← s →						← s →											
x =	1	1	0	1	0	1	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1	1	0
y =	0	0	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	1	1	1	1	1	0	0
pKc(x,s)=	1	1	0	0	1	1	0	1	0	0	0	0	0	1	1	0	0	1	1	1	1	0	1	0

pseudo-Kasami code: weaknesses

Lemma 2.2 *Let $x = x_1, \ldots, x_n$ be a string of n bits chosen uniformly at random, and let s be an integer chosen uniformly at random such that $1 \leq s \leq n$. Moreover, let x' be a new string obtained from x by flipping one bit. Then* $\Pr[\mathtt{hw}(\mathtt{pKc}(x,s)) = \mathtt{hw}(\mathtt{pKc}(x',s))] = \frac{n+1}{2n}$.

Proof:

Case s=n. Happens with $\mathbf{P}^{\times}$=1/n.

Then, x=y $\Rightarrow$ pKc(x,n)=0 and x'=y' $\Rightarrow$ pKc(x',n)=0.

Thus, pKc(x,n)=pKc(x',n).

pseudo-Kasami code: weaknesses

															←		s				→			
$x =$	0	0	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0	0	0	1	0	1	1	0
$y =$	0	1	0	1	1	0	0	0	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0	0
$pKc(x,s)=$	0	1	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1	0

Other values of s. Happens with $\mathbf{P}^{\times}$ =(n-1)/n.

Two bits flipped $\Rightarrow$ pKc(x,s) and pKc(x',s) differ in two bits.

But $\mathbf{P}^{\times}$[hw(pKc(x,s))=hw(pKc(x',s))]= ½.

Thus $\mathbf{P}^{\times}$[hw(pKc(x,s))=hw(pKc(x',s))]= $\frac{1}{n} + \frac{1}{2} \cdot \frac{(n-1)}{n} = \frac{n+1}{2n}$.

pseudo-Kasami code: weaknesses

Lemma 2.3 *Let $x = x_1, \ldots, x_n$ be a string of n bits chosen uniformly at random. Let x' be a new string obtained from x by flipping two randomly selected bits. Then, for any seed s such that $1 \leq s \leq n$, it holds that:*

a) $\mathtt{pKc}(x, s)$ *and* $\mathtt{pKc}(x', s)$ *are equal with probability* $\frac{1}{(n-1)}$

b) $\mathtt{pKc}(x, s)$ *and* $\mathtt{pKc}(x', s)$ *differ in two bits with probability* $\frac{n \cdot (n-2)}{n^2(n-1)}$

c) $\mathtt{pKc}(x, s)$ *and* $\mathtt{pKc}(x', s)$ *differ in four bits with probability* $\frac{n^3-3n+2}{n^2(n-1)}$

d) $\Pr[\mathtt{hw}(\mathtt{pKc}(x, s)) = \mathtt{hw}(\mathtt{pKc}(x', s))] = \frac{3n^2+3n-2}{8n \cdot (n-1)}$.

pseudo-Kasami code: weaknesses

Lemma 2.4 *Let $x = x_1, \ldots, x_n$ be a string of n bits chosen uniformly at random, and let the seed s be equal to $n/2$. Then, $\mathtt{pKc}(x, s)$ is the concatenation of two equal substrings of $n/2$ bits.*

Proof: Let $x = x_L x_R$.

$$s = n/2 \Rightarrow y = x_R x_L$$

$$\mathtt{pKc}(x) = x \oplus y = x_L x_R \oplus x_R x_L = (x_L \oplus x_R)(x_R \oplus x_L)$$

	←					s						→	←					s						→
$x =$	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0
$y =$	1	0	1	1	0	0	0	1	0	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1
$pKc(x,s) =$	0	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	0	0	0	1	1

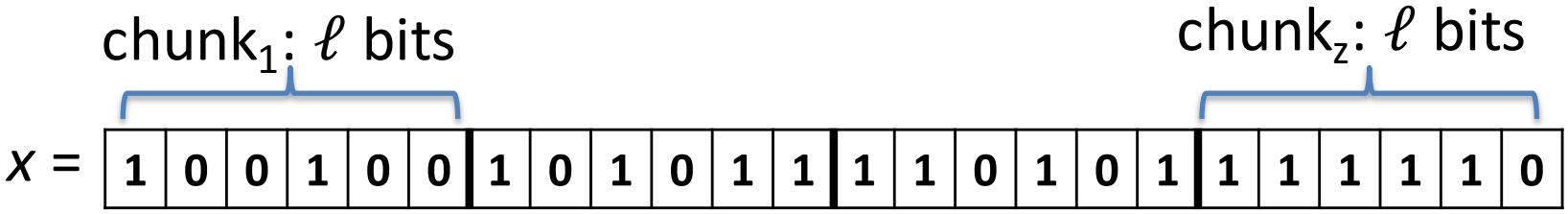
pseudo-Kasami code: weaknesses

- Summarizing
  - **same value** if $s|n$ and we flip n/s bits at distance s from each other
  - if we **flip one bit**, we get the **same weight** with prob. **~** 1/2
  - if we **flip two bits**, we get , among other facts, the **same weight** with prob. ~**3/8**
  - if $s = n/2$, the pseudo-Kasami code exhibits a **repetitive structure**

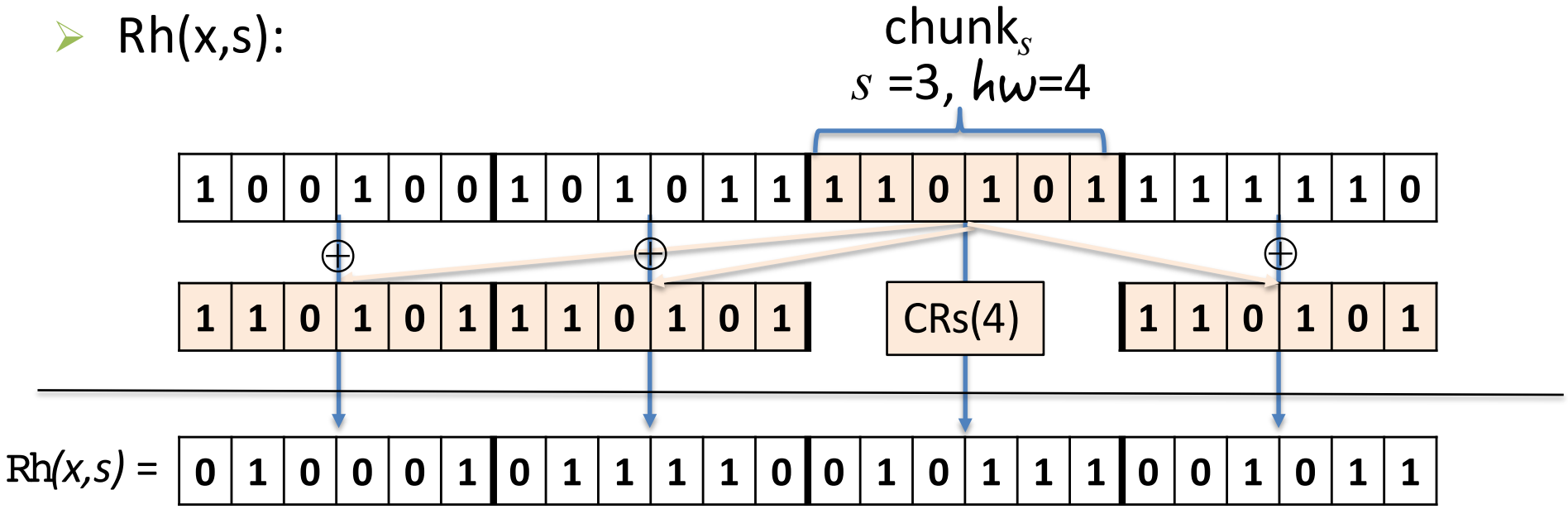
Recursive Hash

Recursive hash

- x=bitstring of $n = z \cdot \ell$ bit (z chunks of ℓ bits each)
- s= seed, $1 \leq s \leq z$



- Rh(x,s):



Recursive hash

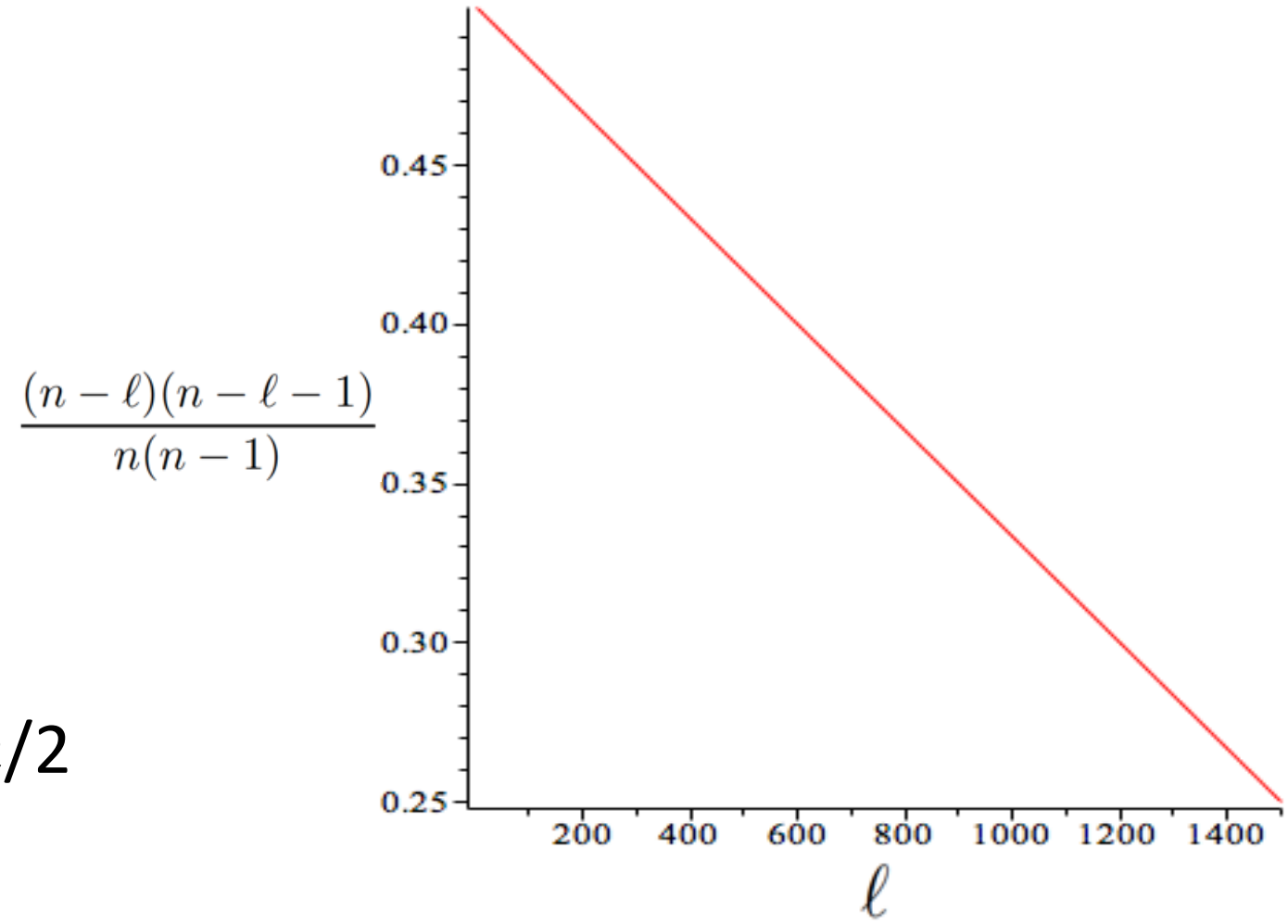
Lemma 2.5 *Let ℓ, n and z integers such that ℓ is a divisor of n and $z = n/\ell$. Moreover, let $x = x_1, \ldots, x_n$ be a string of n bits chosen uniformly at random, and x' a new string obtained from x by flipping two randomly selected bits. Then, for any seed $s \in \{1, \ldots, z\}$, $\mathtt{Rh}(x, s)$ and $\mathtt{Rh}(x', s)$ differ in two bits with probability equal to*

$$\frac{(n-\ell)(n-\ell-1)}{n(n-1)}.$$

Proof: details in the paper. ❐

Recursive hash

- Decreasing linear function of ℓ
  - ✓ Plot for n=3000



- Min for ℓ=n/2
  - ✓ 1/4

Recursive hash

Lemma 2.6 *Let ℓ, n and z integers such that ℓ is a divisor of n and $z = n/\ell$. Moreover, let $x = x_1, \ldots, x_n$ be a string of n bits chosen uniformly at random, and x' a new string obtained from x by flipping two randomly selected bits. Then, for any seed $s \in \{1, \ldots, z\}$, $\Pr[\mathtt{hw}(\mathtt{Rh}(x, s)) = \mathtt{hw}(\mathtt{Rh}(x', s))]$ is at least*

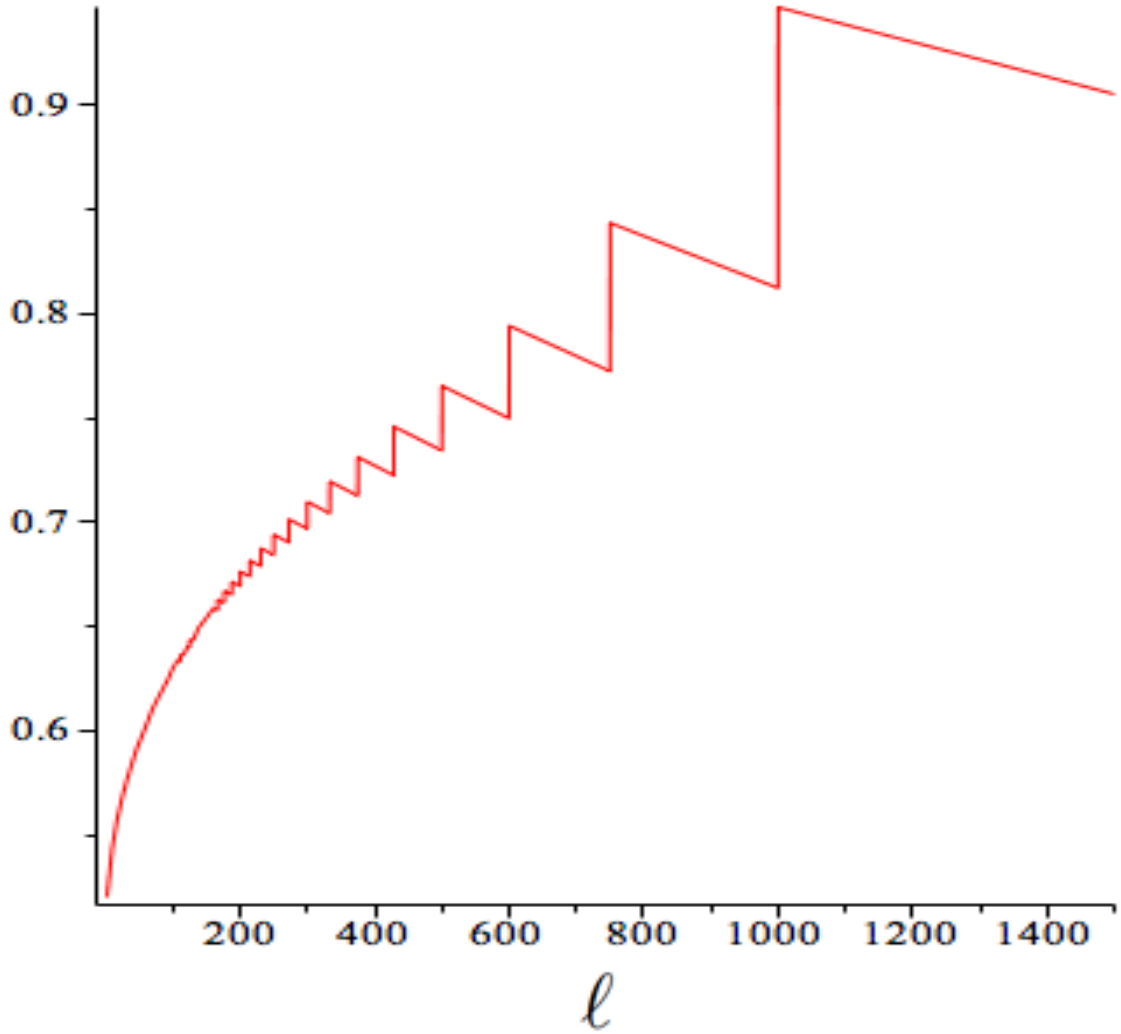
$$\frac{1}{2} \cdot \frac{(n-\ell)(n-\ell-1)}{n(n-1)} + \left(\frac{1}{2}\right)^z \cdot \frac{\ell(\ell-1)}{n(n-1)} + \frac{1}{2} \cdot \frac{n-\ell}{n} \cdot \frac{1}{n-1} \cdot \frac{\binom{z-1}{(z-1)/2}}{2^{(z-1)}} + \frac{1}{2} \cdot \frac{n-\ell}{n} \cdot \frac{\ell-1}{n-1} \cdot \frac{\binom{z+1}{(z+1)/2}}{2^{(z+1)}}.$$

Proof: *Similar to previous one but a little bit more involved* (details in the paper)

❐

Recursive hash

- Increasing function of ℓ
  - ✓ Plot for n=3000



- Min for ℓ=2
  - ✓ 1/2

Conversion

Conversion

- Cnv(A,B,t)
  - ✓ A,B n-bit strings
  - ✓ t integer parameter

- Phase 1: bit grouping of A and B
- Phase 2: bit re-arrangement
- Phase 3: xor

Conversion – Phase 1: bit grouping

- A= 1011 0101 1111 1101 1101 1101
- B= 1011 0101 0101 1101 0101 1111

$A =$	10	1101	0111	111	101	11	01	1101

$B =$	101	10101	01011	101	01	0	11111

Conversion – Phase 2: bit re-arrangement

- Invert bit grouping

$A =$	10	1101	0111	111	101	11	01	1101

$B =$	101	10101	01011	101	01	0	11111

  - ✓ A gets B's bit grouping and vice versa

$A =$	101	10101	11111	101	11	0	11101

$B =$	10	1101	0101	011	101	01	01	1111

  - ✓ Then, each group: left circular shift of its hw

$A' =$	110	10101	11111	110	11	0	11110

$B' =$	01	1110	0101	101	110	10	10	1111

Conversion – Phase 3: xor

$A' =$	1101 0101 1111 1110 1101 1110
$B' =$	0111 1001 0110 1110 1010 1111
$\texttt{Cnv}(A, B, 6) =$	1010 1100 1001 0000 0111 0001

Conversion

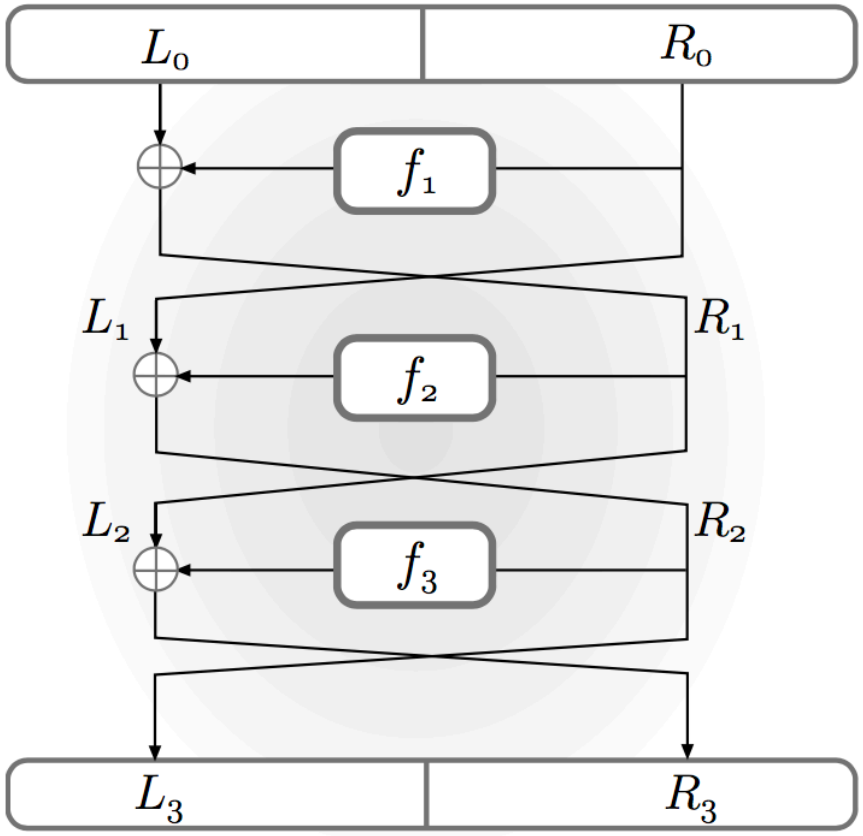
Lemma 2.7 *Given two binary strings A and B of length n, chosen uniformly at random, and a threshold t, with $1 < t \leq n$, if we flip two bits of A in the first t positions to obtain A', then*

$$\Pr[(\mathtt{Cnv}(A, B, t), \mathtt{Cnv}(A', B, t)) \textit{differ in two bits}]$$

$$\approx \frac{1}{2} \cdot \frac{1}{6} \cdot \frac{t^3 - 2t^2 - t + 2}{t(t-1)^2}$$

FCS-Based

FCS-Based transform



The transform F is a **three-round Feistel Cipher**, with a properly defined round function

FCS-Based transform

Splitting K_i in two halves, i.e., $K_i = K_{i_1} K_{i_2}$, the round function f is defined by

$$f(K_i, Z) = (K_{i_1} \bullet Z) << K_{i_2},$$

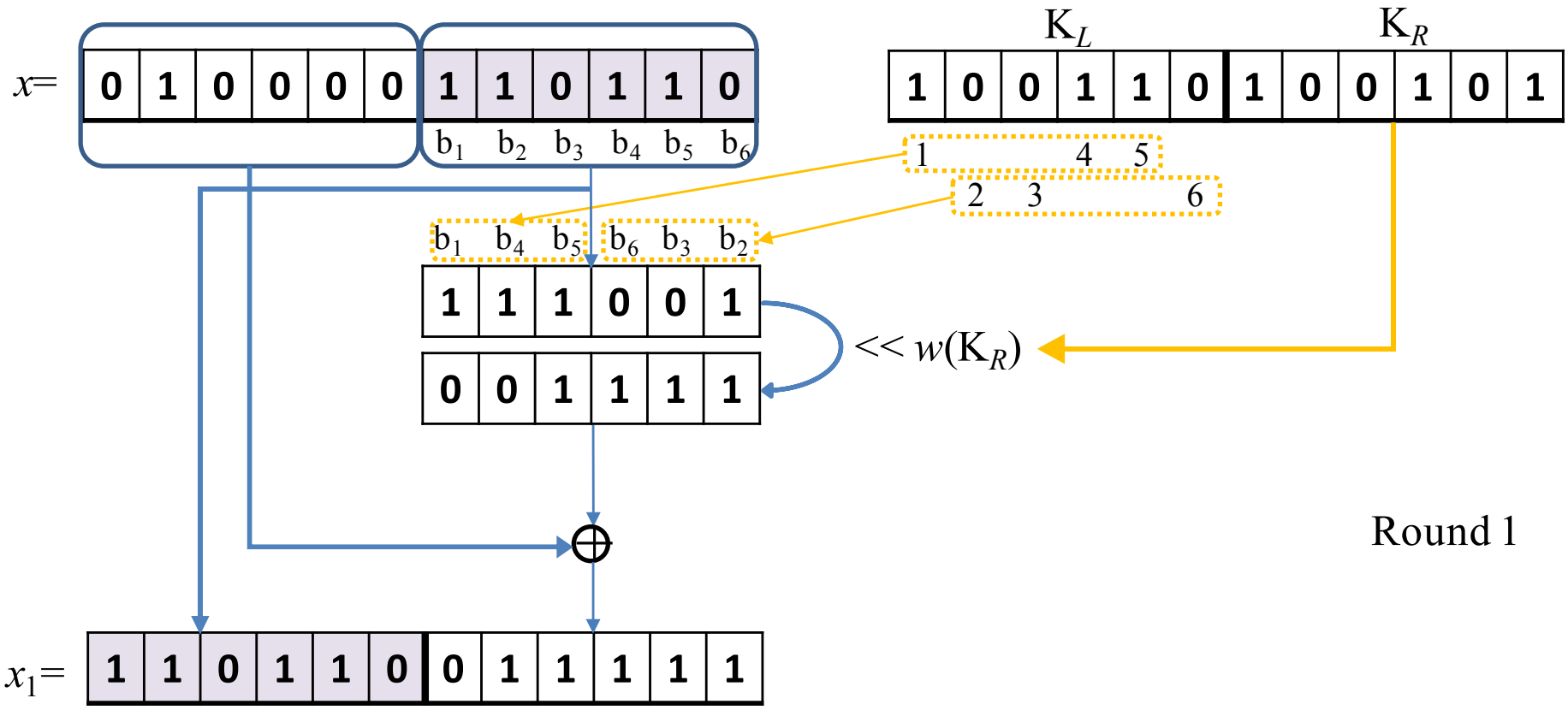
where

- the $\bullet$ operator denotes a permutation of the bits of Z, according to the bits of K_{i_1}
- the $<<$ operator denotes a circular left shift of K_{i_2} positions of the bits of $(K_{i_1} \bullet Z)$.

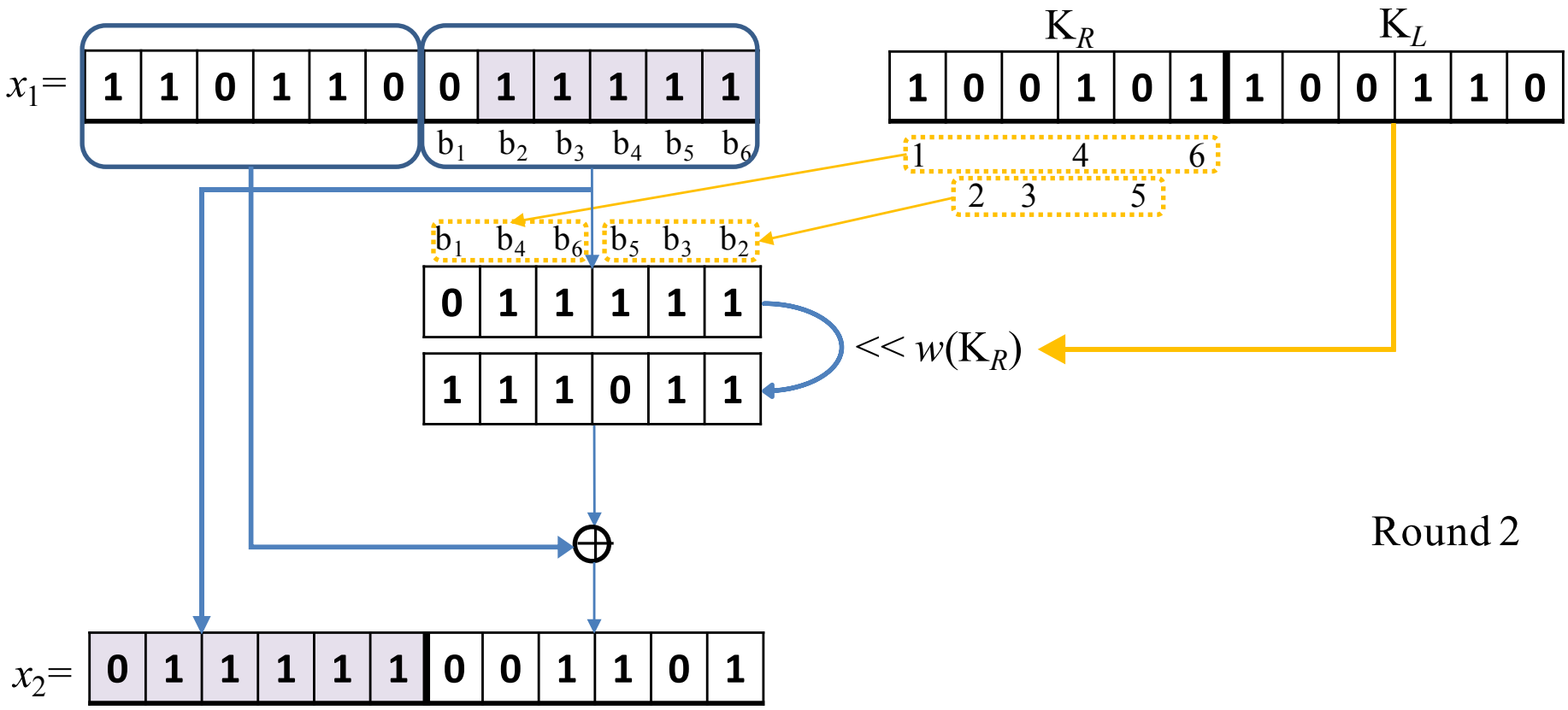
The permutation $C = A \bullet B$ consists in splitting the bits of A in a right and a left parts, according to the bits of B. Specifically, let $A = a_1 \ldots a_n$ and $B = b_1 \ldots b_n$. Moreover, let $s_1 = \{k_1, \ldots, k_m | k_1 < \ldots < k_m\}$ be the set of the indices of the bits of B equal to 1, and let $s_0 = \{k_{m+1}, \ldots, k_n | k_{m+1} < \ldots < k_n\}$ be the set of the indices of the bits of B equal to 0. Then,

$$C = A \bullet B = a_{k_m+1} \ldots a_{k_n-1} a_{k_n} a_{k_m} a_{k_m-1} \ldots a_1.$$

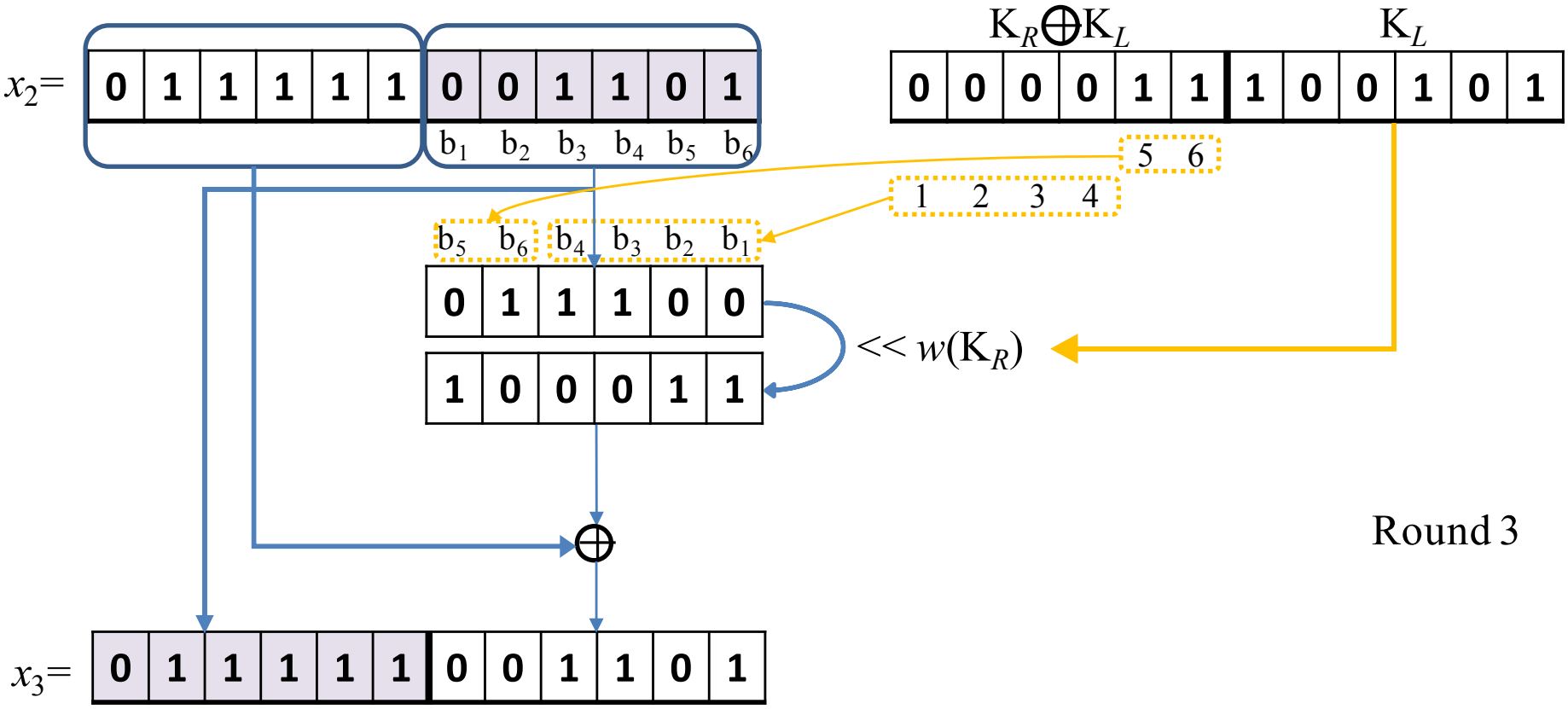
FCS-Based transform



FCS-Based transform



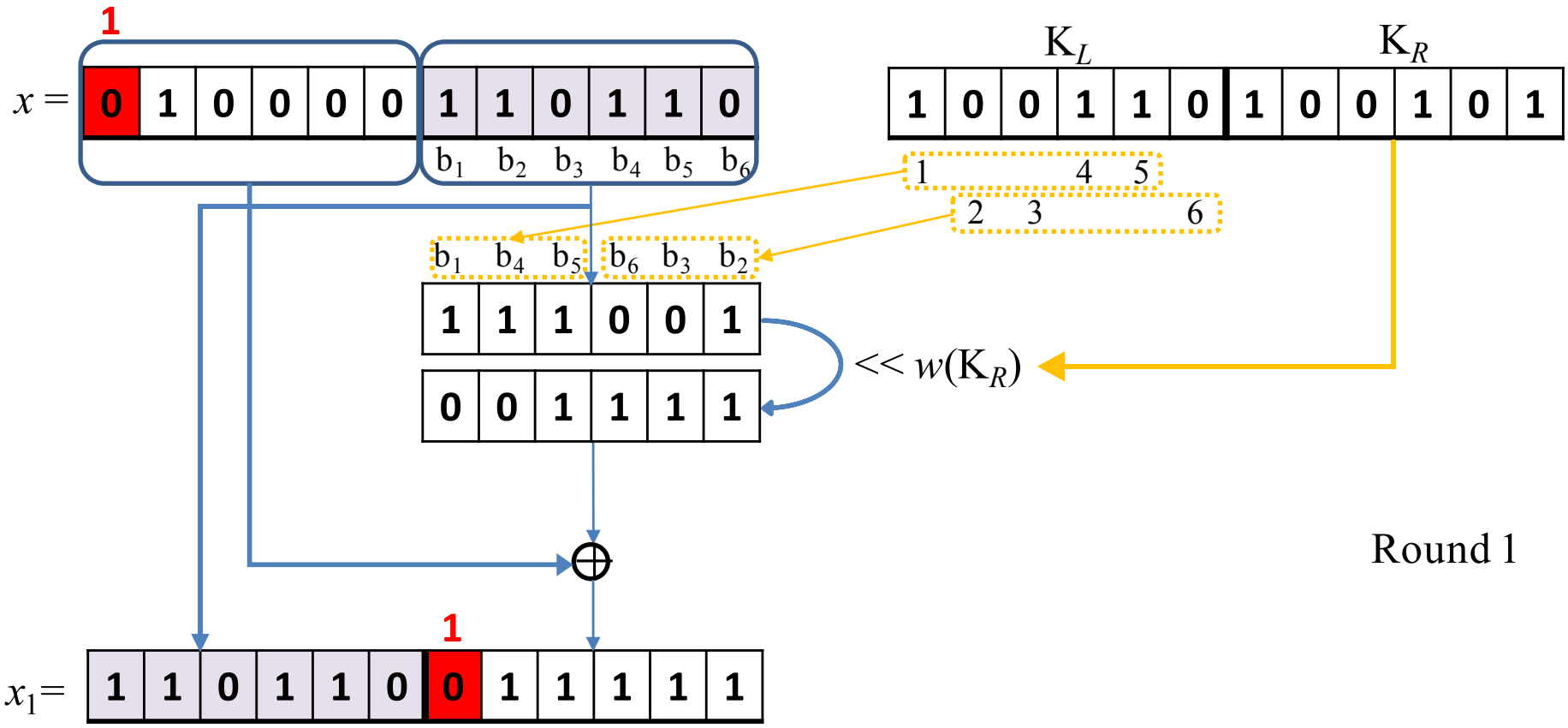
FCS-Based transform



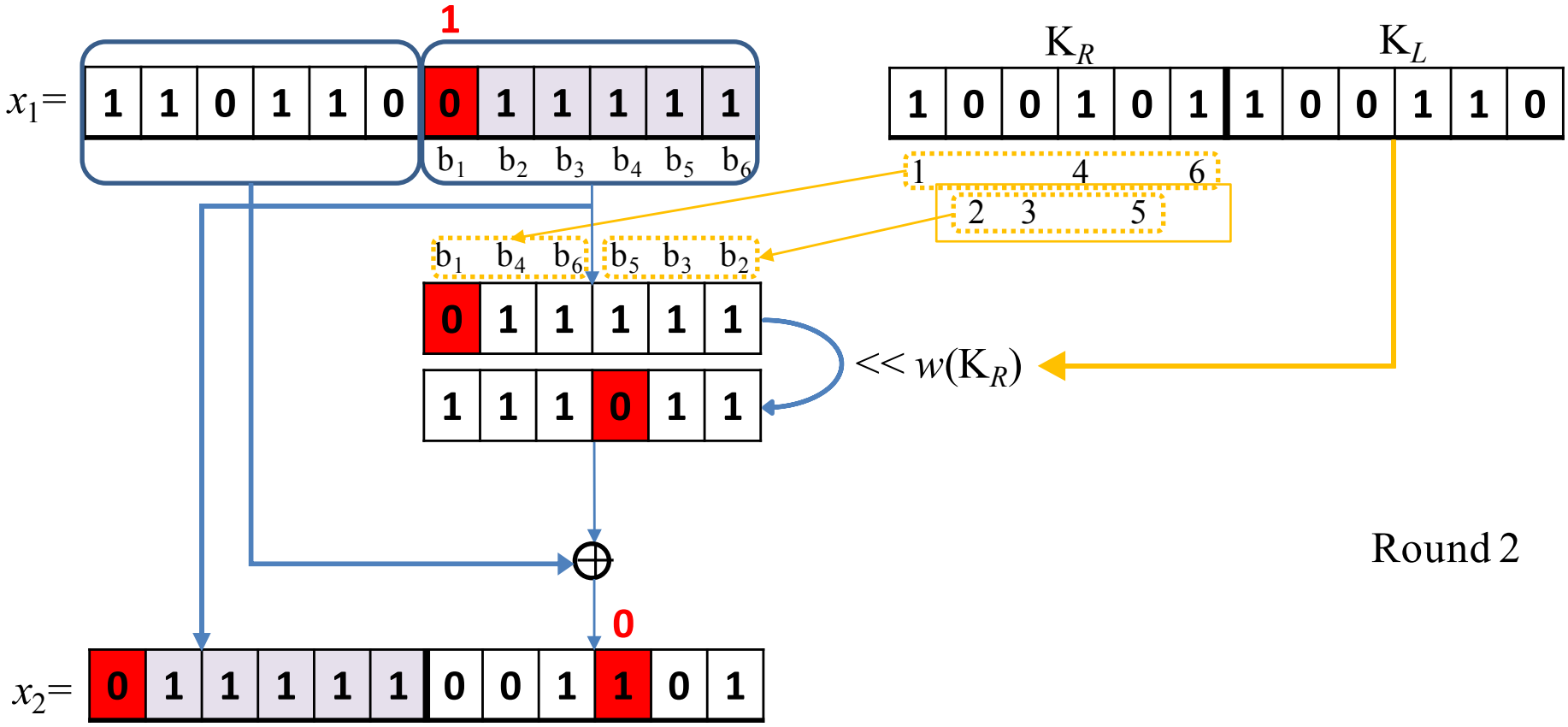
Feistel-like transform

- Flipping one bit of x
  - ✓ causes changes in at most 3 bits

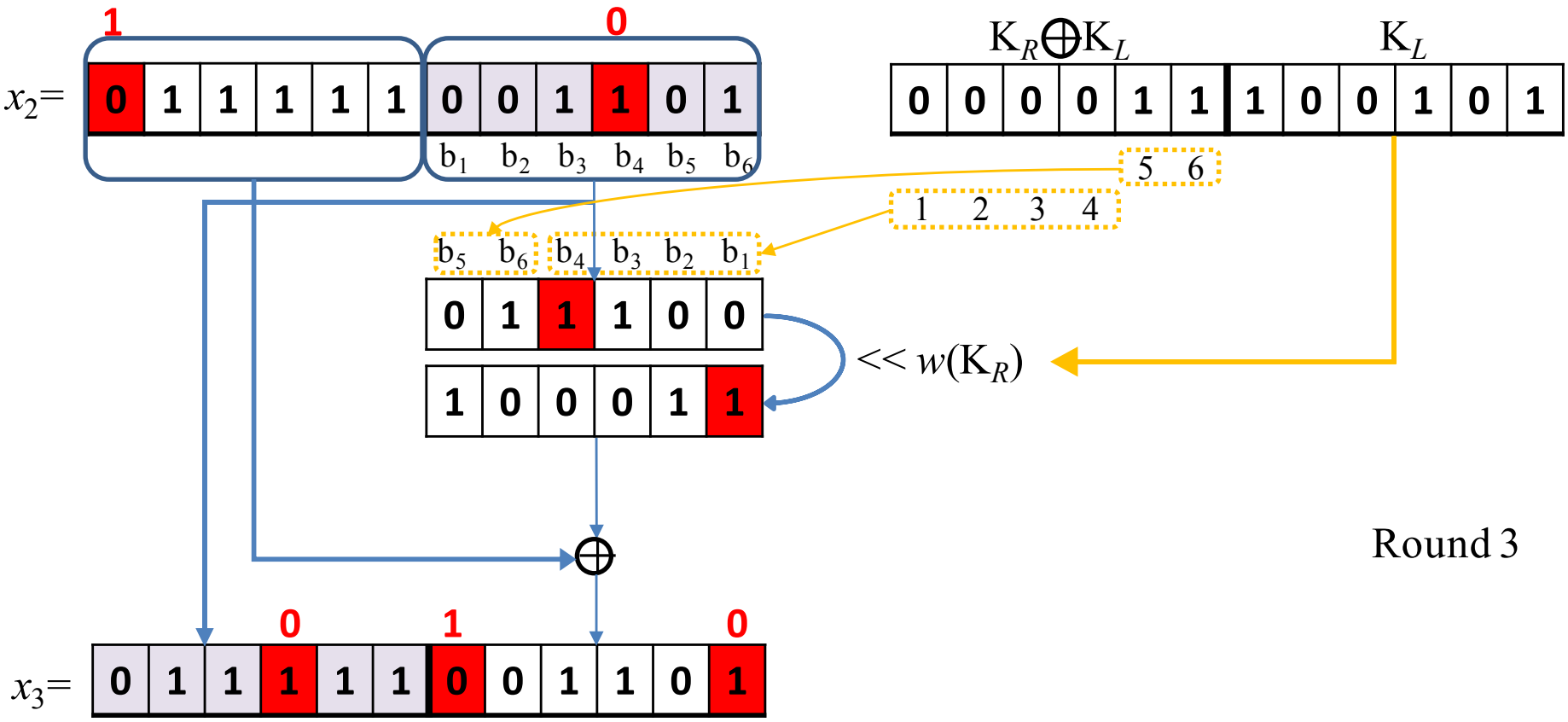
FCS-Based transform



FCS-Based transform



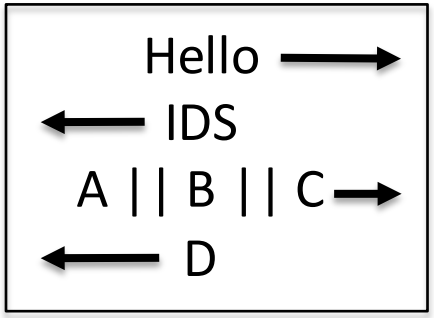
FCS-Based transform



Impersonation attacks

RCIA

Protocol



Shared parameters:
IDS,K_1,K_2

Protocol

Reader chooses randomly n_1 and n_2

$P = n_1 \oplus n_2$

$s = hw(P) \mod b$

$A = \texttt{Rot}(IDS, K_1) \oplus n_1$

$B = (\texttt{Rot}(IDS \wedge n_1, K_2) \wedge K_1) \oplus n_2$

$K_1^* = \texttt{Rot}(\texttt{Rh}(K_2, s), \texttt{Rh}(n_1, s)) \wedge K_1)$

$K_2^* = \texttt{Rot}(\texttt{Rh}(K_1, s), \texttt{Rh}(n_2, s)) \wedge K_2)$

$C = \texttt{Rot}(\texttt{Rh}(K_1^*, s), \texttt{Rh}(K_2^*, s)) \wedge \texttt{Rot}(\texttt{Rh}(n_1, s), \texttt{Rh}(n_2, s))$

$D = \texttt{Rot}(\texttt{Rh}(ID, s), K_1^*)) \wedge (\texttt{Rot}(\texttt{Rh}(K_2^*, s), \texttt{Rh}(n_2, s)) \oplus IDS)$

Updates

$IDS_{new} = \texttt{Rot}(\texttt{Rh}(IDS) \oplus n_2, n_1))$

$K_{1,new} = \texttt{Rh}(K_1^*, s)$

$K_{2,new} = \texttt{Rh}(K_2^*, s)$

Impersonation attack to RCIA

Lemma 4.1 *Assume an adversary eavesdrops an authentication session and stores* $\mathbf{A}||\mathbf{B}||\mathbf{C}$. *Let* $\mathbf{B}'$ *be equal to* $\mathbf{B}$ *up to two consecutive bits which are flipped. Then, forcing the tag to send the old IDS and replying with* $\mathbf{A}||\mathbf{B}'||\mathbf{C}$, *the adversary succeeds in impersonating the legal Reader with probability roughly equal to* $\frac{1}{4}$.

Proof:

- flip two bits of $\mathbf{B} \Rightarrow$ flip two bits of $n_2 \Rightarrow$
- $\mathbf{Pr}[hw(P')=hw(P)]=\mathbf{Pr}[hw(n_1 \oplus n'_2)=hw(n_1 \oplus n_2)]=1/2$
- By Lemma 2.6: $\mathbf{Pr}\,[hw(\mathrm{Rh}(n_2, s))=hw(\mathrm{Rh}(n'_2, s))] \simeq 1/2$
- In such a case K_2^*, $\mathrm{Rh}(K_2^*,s)$, C do not change
- $\mathbf{A}||\mathbf{B}'||\mathbf{C}$ is a valid response ❐

Conclusions and open problems

Conclusions

- We have shown
  - ✓ **weaknesses** in the transforms used in the design of recent ultralightweight authentication protocols
  - ✓ the common properties which are **missing** are **confusion and diffusion**
    - ✧ any change in the input should affect all bits in the output of the transform
  - ✓ efficient **impersonation attacks** against the protocols

Open problems

- The hardware imposes **very strong constraints** on the computing capabilities of the small elements.

- Two choices:
  - ✓ **give up,** because it is difficult (if not impossible) to achieve the security standard we get in other levels of our digital infrastructure

  - ✓ try to achieve a **reasonable** security level
    - ✧ what does "reasonable" mean?

Open problems

- The current state of knowledge is **quite poor**
  - ✓ we do not have any **impossibility result within a model** for such ultralightweight protocols
  - ✓ Neither do we have **positive results**
- A **more in-depth understanding** is needed