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

- **Radio Frequency Identification**
  - 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 (~30m)

- **Passive**
  - No battery, gets the power from the reader, smaller communication range (~10m)

- **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**
  - **RCIA**
  - **SASI+**
  - **SLAP**
  - **FCS-Based**
Protocol structure

- They have essentially the same general structure

<table>
<thead>
<tr>
<th>Reader</th>
<th>Message</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hello</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IDS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>
Core tools

- Transforms
  - Pseudo-Kasami code
  - Recursive hash
  - Conversion
  - FCS-Based

- We analyze them
  - show several weaknesses
TRANSFORMS
pseudo-Kasami code
pseudo-Kasami code

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

\[
\begin{align*}
x &= 10010111010111011000101110 \\
y &= 01011101001011101011011100 \\
pKc(x,s) &= 1100110100000011000111101010
\end{align*}
\]
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 \( p\text{Kc}(x, s) \) 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 \( p\text{Kc}(x, s) = p\text{Kc}(x', s) \).

Proof:
Since \( y = \text{CRshift}(x, s) \) flippings cancel themselves out in the xor.

\[
\begin{align*}
x &= 11010111010111110000011110 \\
y &= 00011101110101110001011111100 \\
p\text{Kc}(x, s) &= 1100111010000001110011111010
\end{align*}
\]
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\[
\Pr[\text{hw}(pKc(x, s)) = \text{hw}(pKc(x', s))] = \frac{n+1}{2n}.
\]

Proof:

Case $s=n$. Happens with $P_X=1/n$.

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

\[ x = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \]

\[ y = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \]

\[ pKc(x,s) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]

Other values of \( s \). Happens with \( P \times (n-1)/n \).

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

But \( P \times [hw(pKc(x,s)) = hw(pKc(x',s))] = 1/2 \).

Thus \( P \times [hw(pKc(x,s)) = hw(pKc(x',s))] = \frac{1}{n} + \frac{1}{2} \cdot \frac{(n-1)}{n} = \frac{n+1}{2n} \).
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) $pKc(x, s)$ and $pKc(x', s)$ are equal with probability $\frac{1}{n-1}$

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

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

d) $\Pr[\text{hw}(pKc(x, s)) = \text{hw}(pKc(x', s))] = \frac{3n^2 + 3n - 2}{8n(n-1)}$. 
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, \( pKc(x, s) \) is the concatenation of two equal substrings of \( n/2 \) bits.

Proof: Let \( x = x_Lx_R \).

\[
s = n/2 \Rightarrow y = x_Rx_L
\]

\[
pKc(x) = x \oplus y = x_Lx_R \oplus x_Rx_L = (x_L \oplus x_R)(x_R \oplus x_L)
\]

\[
x = \begin{array}{cccccccccccccccccc}
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0
\end{array}
\]

\[
y = \begin{array}{cccccccccccccccccc}
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1
\end{array}
\]

\[
pKc(x,s) = \begin{array}{cccccccccccccccccc}
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}
\]
pseudo-Kasami code: weaknesses

- Summarizing
  - If $s \mid n$ and we flip $n/s$ bits at distance $s$ from each other,
  - If we flip one bit, we get the same weight with prob. $\sim 1/2$
  - If we flip two bits, we get, among other facts, the same weight with prob. $\sim 3/8$
  - 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 $\ell$ bits each)
- $s$= seed, $1 \leq s \leq z$

chunk₁: $\ell$ bits

chunk₂: $\ell$ bits

$\begin{array}{c}
\text{Chunk}_1: \ell \text{ bits} \\
\text{Chunk}_2: \ell \text{ bits} \\
\end{array}$

$s=3, h\omega=4$

$\begin{array}{c}
\text{Chunk}_s \\
\end{array}$

$\begin{array}{c}
\text{CrS}(4) \\
\end{array}$

$\begin{array}{c}
\text{CrS}(4) \\
\end{array}$
Recursive hash

**Lemma 2.5** Let \( \ell, n \) and \( z \) integers such that \( \ell \) 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 \( s \in \{1, \ldots, z\} \), \( \text{Rh}(x, s) \) and \( \text{Rh}(x', s) \) 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 $\ell$
  - Plot for $n=3000$

- Min for $\ell=n/2$
  - $1/4$
Recursive hash

Lemma 2.6 Let \( \ell, n \) and \( z \) integers such that \( \ell \) 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 \( s \in \{1, \ldots, z\} \), \( \Pr[\text{hw}(\text{Rh}(x, s)) = \text{hw}(\text{Rh}(x', s))] \) is at least

\[
\frac{1}{2} \cdot \frac{(n-\ell)(n-\ell-1)}{n(n-1)} + \left(\frac{1}{2}\right)^z \cdot \frac{\ell(\ell-1)}{n(n-1)} + \frac{1}{2} \cdot \frac{n-\ell}{n} \cdot \frac{1}{n-1} \cdot \frac{\left(\frac{z-1}{2}\right)}{2^{(z-1)}} + \frac{1}{2} \cdot \frac{n-\ell}{n} \cdot \frac{\ell-1}{n-1} \cdot \frac{\left(\frac{z+1}{2}\right)}{2^{(z+1)}}.
\]

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

- Increasing function of $\ell$
  - Plot for $n=3000$

- Min for $\ell=2$
  - $1/2$
Conversion

- $\text{Cnv}(A, B, t)$
  - $A, B$ $n$-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

\[
\begin{array}{cccccccc}
A = & 10 & 1101 & 0111 & 111 & 101 & 11 & 01 & 1101 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
B = & 101 & 10101 & 01011 & 101 & 01 & 0 & 11111 \\
\end{array}
\]
Conversion – Phase 2: bit re-arrangement

- Invert bit grouping

\[
\begin{align*}
A &= \begin{array}{cccccccc}
10 & 1101 & 0111 & 111 & 101 & 11 & 01 & 1101
\end{array} \\
B &= \begin{array}{cccccccc}
101 & 10101 & 01011 & 101 & 01 & 0 & 11111
\end{array}
\end{align*}
\]

- A gets B’s bit grouping and vice versa

\[
\begin{align*}
A &= \begin{array}{cccccccc}
101 & 10101 & 11111 & 101 & 11 & 0 & 11101
\end{array} \\
B &= \begin{array}{cccccccc}
10 & 1101 & 0101 & 011 & 101 & 01 & 01 & 1111
\end{array}
\end{align*}
\]

- Then, each group: left circular shift of its \(hw\)

\[
\begin{align*}
A' &= \begin{array}{cccccccc}
110 & 10101 & 11111 & 110 & 11 & 0 & 11110
\end{array} \\
B' &= \begin{array}{cccccccc}
01 & 1110 & 0101 & 101 & 110 & 10 & 10 & 1111
\end{array}
\end{align*}
\]
Conversion – Phase 3: xor

<table>
<thead>
<tr>
<th>$A'$ =</th>
<th>1101 0101 1111 1110 1101 1110</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B'$ =</td>
<td>0111 1001 0110 1110 1010 1111</td>
</tr>
<tr>
<td>$\text{Conv}(A, B, 6) =$</td>
<td>1010 1100 1001 0000 0111 0001</td>
</tr>
</tbody>
</table>
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

\[
\Pr[(\text{Cnv}(A, B, t), \text{Cnv}(A', B, t)) \text{ differ in two bits}] \approx \frac{1}{2} \cdot \frac{1}{6} \cdot \frac{t^3 - 2t^2 - t + 2}{t(t - 1)^2}
\]
Confusion and diffusion in recent ultralightweight RFID authentication protocols

FCS-Based
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., $K_i = K_{i1}K_{i2}$, the round function $f$ is defined by

$$f(K_i, Z) = (K_{i1} \bullet Z) \ll K_{i2},$$

where

- the $\bullet$ operator denotes a permutation of the bits of $Z$, according to the bits of $K_{i1}$
- the $\ll$ operator denotes a circular left shift of $K_{i2}$ positions of the bits of $(K_{i1} \bullet Z)$.

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,

$$C = A \bullet B = a_{k_m+1} \ldots a_{k_1} a_n a_{k_m} a_{k_{m-1}} \ldots a_1.$$
FCS-Based transform

\[ x = \begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{array} \]

\[ \begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & 0 \\ b_1 & b_4 & b_5 & b_6 & b_3 & b_2 \end{array} \]

\[ x_1 = \begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \]

\[ K_L \]

\[ K_R \]

\[ \ll w(K_R) \]

Round 1
FCS-Based transform

\[ x_1 = \begin{array}{cccccc}
1 & 1 & 0 & 1 & 1 & 0 \\
\end{array} \]

\[ x_2 = \begin{array}{cccccc}
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 \\
\end{array} \]

\[ b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \]

\[ \ll w(K_R) \]

Round 2

\[ K_R = \begin{array}{cccccccc}
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
\end{array} \]

\[ K_L = \begin{array}{cccccccc}
1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
\end{array} \]
FCS-Based transform

\[ x_2 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \end{bmatrix} \]

\[ x_3 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \]

\[ K_R \oplus K_L \]

Round 3
Feistel-like transform

- Flipping one bit of $x$
  - causes changes in at most 3 bits
FCS-Based transform

Round 1

Confusion and diffusion in recent ultralightweight RFID authentication protocols
FCS-Based transform

\[ x_1 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \]

\[ x_2 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \]

\[ 1 \]

\[ \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \]

\[ \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{bmatrix} \]

\[ \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \]

\[ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \]

\[ \begin{bmatrix} 10010111001110 \end{bmatrix} \]

\[ \begin{bmatrix} 10010111001110 \end{bmatrix} \]

\[ \begin{bmatrix} 10010111001110 \end{bmatrix} \]

\[ \begin{bmatrix} 10010111001110 \end{bmatrix} \]

\[ K_R \]

\[ K_L \]

\[ \ll w(K_R) \]

Round 2
FCS-Based transform

\[ x_2 = \begin{array}{cccccc}
0 & 1 & 1 & 1 & 1 & 1 \\
\end{array} \]

\[ x_3 = \begin{array}{cccccc}
0 & 1 & 1 & 1 & 1 & 0 \\
\end{array} \]

\[ K_R \oplus K_L \]

\[ \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{array} \]

\[ K_L \]

\[ \begin{array}{cccccccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{array} \]

\[ \ll w(K_R) \]

Round 3
Impersonation attacks
**Protocol**

Reader chooses randomly $n_1$ and $n_2$

$P = n_1 \oplus n_2$

$s = \text{hw}(P) \mod b$

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

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

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

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

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

$D = \text{Rot}(\text{Rh}(IDS, s, K_1^*)) \land (\text{Rot}(\text{Rh}(K_2^*, s), \text{Rh}(n_2, s)) \oplus IDS)$

**Updates**

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

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

$K_{2,new} = \text{Rh}(K_2^*, s)$
Impersonation attack to RCIA

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

Proof:

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