Cryptanalysis of the David-Prasad RFID Ultralightweight Authentication Protocol

Julio Cesar Hernandez-Castro[1], Pedro Peris-Lopez[2], Raphael C.-W. Phan[3], and Juan M. E. Tapiador[4]

[1] School of Computing, University of Portsmouth
[2] Security Lab, Faculty of EEMCS, Delft University of Technology
[3] Department of Electronic and Electrical Engineering, Loughborough University
[4] Department of Computer Science, University of York

Abstract. In September 2009, David and Prasad proposed at MobiSec'09 an interesting new ultralightweight mutual authentication protocol for low-cost RFID tags. In this paper, we present a quite powerful cryptanalytic attack against their proposal: we start with a traceability attack, then describe how it can be extended to leak long-term stored secrets, and finally present a full disclosure attack (named Tango attack) where all the secrets that the protocol is designed to conceal are shown to be retrievable, even by a passive attacker after eavesdropping only a small number of authentication sessions. These results imply that very realistic attack scenarios are completely possible. The Tango attack constitutes a new, simple, yet powerful technique of cryptanalysis which is based on the computation and full exploitation of multiple approximations to the secret values, using Hamming distances and the representation of variables in an n-dimensional space.

1 Introduction

Authentication protocols for Radio Frequency IDentification (RFID) systems allow an RFID reader and a tag to mutually authenticate each other. Numerous protocols have been recently proposed in the literature, and the field is challenging since RFID tags can only work in very confined environments with scarce resources, so protocols should ensure that the underlying computations are not resource intensive. Along this line, a class of ultralightweight authentication protocols have been proposed, notably [7–9]. These protocols use only triangular operations, e.g. exclusive OR (XOR), bitwise AND, bitwise NOT, which are very lightweight but, on the other hand, only offer very limited diffusion properties.

One of the critical requirements for RFID authentication protocols is that they should be untraceable, i.e. it should not be possible for a tag's movements to be traced; this is especially relevant when considered that tags are to be embedded within objects (e.g. clothing), and thus inherently ubiquitous. Aside from mounting traceability attacks, stronger attacks can be performed by passive adversaries, including the recovery of all the long-term secrets stored on tags,

which implies that the tag is not only traceable but also fully identifiable and clonable. Anonymity would be thus entirely broken.

This paper presents cryptanalytic results both in terms of traceability attacks and attacks that recover long-term stored secrets, including the keys and the static identifier. These only require the adversary to be passive (i.e. to eavesdrop), and thus are devastating attacks with huge security implications for the protocol under scrutiny.

In mounting these attacks, we demonstrate the full power of recent cryptanalytic developments, notably the traceability attack based on truth table differences with respect to an untraceability game [10], and the Tango cryptanalysis which is based on the computation of multiple approximations, and is a novel technique firstly introduced in this paper.

In the following we apply these cryptanalytic techniques to a recent RFID protocol proposed by David and Prasad at MobiSec '09 [2], and show and analyze the results in some depth.

2 The David-Prasad Protocol

In September 2009, David and Prasad proposed at MobiSec'09 a new ultralightweight authentication protocol inspired by previous approaches such as the UMAP family of protocols [7–9], and the SASI [1] and Gossamer [6] schemes. Their proposal aims to provide a strong authentication mechanism and, at the same time, to offer a significant reduction in the computational load of the tag, without compromising security.

The tag and the server (also called back-end database) share four values: The old and the potential new pseudonym $\{P_{ID}, P_{ID2}\}$, respectively, and two secret keys $\{K_1, K_2\}$. Furthermore, the tag stores a static identifier ID which facilitates its unequivocal identification. The authors assume that the ID and all the remaining variables have the same bit length (i.e. $\{P_{ID}, P_{ID2}, K_1, K_2, ID\} \in Z_2^{96}$). The common communication model is assumed, so communications between the reader and the server – both arguably powerful devices – are considered to be secure as these entities can afford to use classical security solutions (e.g., TLS or SSL). On the other hand, the forward (reader-to-tag) and backward (tag-to-reader) channels are considered to be insecure and open to all sorts of attacks.

We now describe the protocol, which is divided into six steps. The operands $\{\oplus, \wedge\}$ symbolize the bitwise exclusive OR (XOR) and the bitwise AND, respectively, while $\overline{x}$ denotes the bitwise NOT of x.

Step 1: The reader sends a request message $C_{request}$ to the server. If it proves to be an authorized reader, the server sends a one-day authorization access certificate C. If the reader has already a valid certificate, it jumps directly to Step 2.

Step 2: The reader sends a request message $ID_{request}$ to the tag, which replies with its pseudonym P_{ID2}.

Step 3: The reader sends the tuple $\{P_{ID2}, C\}$ to the server in order to acquire the private information linked to the tag. If the certificate is valid and P_{ID2} matches one of the entries in the database, the server sends $\{K_1, K_2\}$ back to the reader. Otherwise, the server informs the reader that P_{ID2} does not correspond to any entry in its database. In that case, the reader repeats Step 2 in order to get access to the old pseudonym P_{ID} of the tag. Then, Step 3 is executed with the tuple $\{P_{ID}, C\}$.

Step 4: The reader generates two random numbers n_1 and n_2. Then, it computes messages $\{A, B, D\}$ as follows and sends them to the tag:

$$A = (P_{ID2} \wedge K_1 \wedge K_2) \oplus n_1 \tag{1}$$

$$B = (\overline{P_{ID2}} \wedge K_2 \wedge K_1) \oplus n_2 \tag{2}$$

$$D = (K_1 \wedge n_2) \oplus (K_2 \wedge n_1) \tag{3}$$

Step 5: From messages $\{A, B\}$, the tag can easily infer the value of the nonces $\{n_1, n_2\}$ associated to the current session. Using these values, it computes its local version of message D (let's call it D') and checks if it is identical to the received value. If they coincide, then the reader is authenticated. Otherwise, the protocol is aborted. After a successful reader authentication, the tag computes messages $\{E, F\}$ as follows and sends them back to the reader:

$$E = (K_1 \oplus n_1 \oplus ID) \oplus (K_2 \wedge n_2) \tag{4}$$

$$F = (K_1 \wedge n_1) \oplus (K_2 \wedge n_2) \tag{5}$$

Finally, the tag updates its pseudonyms values using the session nonces:

$$P_{ID} = P_{ID2} \tag{6}$$

$$P_{ID2} = P_{ID2} \oplus n_1 \oplus n_2 \tag{7}$$

Step 6: Upon receiving messages E and F, the reader computes a local version, F', and checks if it is identical to the received value. If both coincide, the tag is authenticated and the reader can obtain the static identifier ID of the tag by using message E and the now known values $\{K_1, K_2, n_1, n_2\}$ (i.e., $ID = E \oplus (K_2 \wedge n2) \oplus K_1 \oplus n_1$). It then updates the pseudonyms linked to the tag in the same way:

$$P_{ID} = P_{ID2} \tag{8}$$

$$P_{ID2} = P_{ID2} \oplus n_1 \oplus n_2 \tag{9}$$

Finally, the reader sends an updated version of the pair $\{P_{ID}, P_{ID2}\}$ and its certificate C to the server. If the certificate is valid, the server updates the information (pseudonyms) associated to the tag.

3 Traceability Attack

Traceability is one of the most important security threats in RFID environments. Nevertheless, numerous RFID protocols put it at risk by designing schemes where

tags answer readers' queries with static values, thus making traceability attacks not only possible but trivial. For these and other reasons (notably the privacy implications due to tags' mobility), the traceability problem has recently attracted a lot of interesting research. In [4], Juels and Weis gave a formal definition of traceability, that was later reformulated, in a style more similar to that used for security protocols, in [10]. We use the latter approach to analyze the David-Prasad protocol. For completeness and readability, we will first present the model, and later we will detail our proposed attack.

In RFID schemes, tags ($\mathcal{T}$) and readers ($\mathcal{R}$) interact in protocol sessions. In general terms, the adversary ($\mathcal{A}$) controls the communications between all the participants and interacts passively or actively with them. Specifically, $\mathcal{A}$ can run the following queries:

- Execute($\mathcal{R}$, $\mathcal{T}$, i) query. This models a passive attacker. $\mathcal{A}$ eavesdrops on the channel, and gets read access to the exchanged messages between $\mathcal{R}$ and $\mathcal{T}$ in session i of a genuine protocol execution.
- Test(i, $\mathcal{T}_0$, $\mathcal{T}_1$) query. This does not model any ability of $\mathcal{A}$, but it is necessary to define the untraceability test. When this query is invoked for session i, a random bit is generated $b \in \{0,1\}$. Then, a pseudonym $P_{ID2}^{\mathcal{T}_b}(i)$ and a new set of exchanged messages $\{A^{\mathcal{T}_b}, B^{\mathcal{T}_b}, D^{\mathcal{T}_b}, E^{\mathcal{T}_b}, F^{\mathcal{T}_b}\}$ from the set $\{P_{ID2}^{\mathcal{T}_0}(i), P_{ID2}^{\mathcal{T}_1}(i)\}$ and $\{\{A^{\mathcal{T}_0}, B^{\mathcal{T}_0}, D^{\mathcal{T}_0}, E^{\mathcal{T}_0}, F^{\mathcal{T}_0}\}, \{A^{\mathcal{T}_1}, B^{\mathcal{T}_1}, D^{\mathcal{T}_1}, E^{\mathcal{T}_1}, F^{\mathcal{T}_1}\}\}$, respectively, and corresponding to tags $\{\mathcal{T}_0, \mathcal{T}_1\}$ is given to $\mathcal{A}$.

Upon definition of the adversary's abilities, the untraceability problem can be defined as a game $\mathcal{G}$ divided into three phases:

Phase 1 (Learning): $\mathcal{A}$ can make any number of Execute queries, which facilitate the eavesdropping of exchanged messages – modeling a passive attack – over the insecure radio channel.

Phase 2 (Challenge): $\mathcal{A}$ chooses two current tags whose associated identifiers are $ID^{\mathcal{T}_0}$ and $ID^{\mathcal{T}_1}$. He then sends a Test(i, $\mathcal{T}_0$, $\mathcal{T}_1$) query. As a result, $\mathcal{A}$ is given a pseudonym $P_{ID2}^{\mathcal{T}_b}(i)$ and a new set of exchanged messages $\{A^{\mathcal{T}_b}, B^{\mathcal{T}_b}, D^{\mathcal{T}_b}, E^{\mathcal{T}_b}, F^{\mathcal{T}_b}\}$ from the set $\{P_{ID2}^{\mathcal{T}_0}(i), P_{ID2}^{\mathcal{T}_1}(i)\}$ and $\{\{A^{\mathcal{T}_0}, B^{\mathcal{T}_0}, D^{\mathcal{T}_0}, E^{\mathcal{T}_0}, F^{\mathcal{T}_0}\}, \{A^{\mathcal{T}_1}, B^{\mathcal{T}_1}, D^{\mathcal{T}_1}, E^{\mathcal{T}_1}, F^{\mathcal{T}_1}\}\}$, respectively, which depend on a chosen random bit $b \in \{0,1\}$.

Phase 3 (Guessing) $\mathcal{A}$ ends the game and outputs a bit d ($d \in \{0,1\}$) as its conjecture of the value of b.

$\mathcal{A}$'s success in winning $\mathcal{G}$ is equivalent to the success of breaking the untraceability property offered by the protocol. So the advantage of $\mathcal{A}$ in distinguishing whether the messages correspond to $\mathcal{T}_0$ or $\mathcal{T}_1$ is defined below, where t is a security parameter (e.g. the bit length of the key shared by the tag and the reader) and r is the number of times $\mathcal{A}$ runs an Execute query.

$$Adv_{\mathcal{A}}^{\mathsf{UNT}}(t,r) = |Pr[d=b] - \frac{1}{2}|.$$

So, an RFID protocol offers resistance against traceability, i.e. it is said to be untraceable (UNT), if $Adv_{\mathcal{A}}^{\mathsf{UNT}}(t,r) < \varepsilon(t,r)$, where $\varepsilon(\cdot,\cdot)$ symbolizes some negligible function.

In essence, this untraceability (UNT) notion is analogous to the conventional notion of ciphertext indistinguishability (IND) for encryption or key indistinguishability for key establishement protocols. In similar vein, the UNT notion captures the fact that no adversary can distinguish between two tags even if s/he can choose what they are to be. Indeed, if the adversary cannot do this, then clearly s/he cannot track a tag's movements.

We will show in the following how the David-Prasad scheme does not satisfy the above mentioned condition, thus putting at risk the privacy location of tags holders. More precisely, an adversary $\mathcal{A}$ conducts the procedure described below:

Phase 1 (Learning): $\mathcal{A}$ makes the query Execute($\mathcal{R}$, $\mathcal{T}_0$, i), and thus obtains the pseudonym $X^i = P_{ID2}^{\mathcal{T}_0}(i)$ and messages $\{A, B, D, E, F\}$.
By computing the XOR between E and F, we get

$$\begin{aligned} E \oplus F &= (K_1 \oplus n_1 \oplus ID) \oplus (K_2 \wedge n_2) \oplus (K_1 \wedge n_1) \oplus (K_2 \wedge n_2) \\ &= (K_1 \oplus n_1 \oplus ID) \oplus (K_1 \wedge n_1) \\ &= (K_1 \oplus n_1) \oplus (K_1 \wedge n_1) \oplus ID. \end{aligned}$$

If we analyze bit by bit the truth tables provided below

a	b	$a \oplus b$	$a \wedge b$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

it is easy to see that XOR and AND are complements of each other with probability $\frac{3}{4}$. Therefore, for any bit position, the bit value of $(K_1 \oplus n_1)$ is the opposite of that of $(K_1 \wedge n_1)$ with probability $\frac{3}{4}$, so their XOR is 1. Thus we have that $E \oplus F = \overline{ID}$ for each bit with probability $\frac{3}{4}$.

Phase 2 (Challenge): $\mathcal{A}$ chooses two new tags whose associated identifiers are $ID^{\mathcal{T}_0}$ and $ID^{\mathcal{T}_1}$. He then sends a Test(i', $\mathcal{T}_0$, $\mathcal{T}_1$) query. As a result, $\mathcal{A}$ is given a new pseudonym $P_{ID2}^{\mathcal{T}_b}(i)$ and a new set of exchanged messages $\{A^{\mathcal{T}_b}, B^{\mathcal{T}_b}, D^{\mathcal{T}_b}, E^{\mathcal{T}_b}, F^{\mathcal{T}_b}\}$ from the set $\{P_{ID2}^{\mathcal{T}_0}(i), P_{ID2}^{\mathcal{T}_1}(i)\}$ and $\{\{A^{\mathcal{T}_0}, B^{\mathcal{T}_0}, D^{\mathcal{T}_0}, E^{\mathcal{T}_0}, F^{\mathcal{T}_0}\}, \{A^{\mathcal{T}_1}, B^{\mathcal{T}_1}, D^{\mathcal{T}_1}, E^{\mathcal{T}_1}, F^{\mathcal{T}_1}\}\}$, respectively, which depend on a chosen random bit $b \in \{0,1\}$.

Phase 3 (Guessing) $\mathcal{A}$ ends $\mathcal{G}$ and outputs a bit $d = \overline{\mathsf{lsb}(E \oplus F)} \oplus \overline{\mathsf{lsb}(E^{\mathcal{T}_b} \oplus F^{\mathcal{T}_b})}$ as its conjecture of the value b, where $\mathsf{lsb}(\cdot)$ denotes the least significant bit.

The adversary outputs as d the bitwise NOT of the lsb of $E \oplus F$. From our analysis in the description of Phase 1 above, we recall that $\mathsf{lsb}(E \oplus F)$ equals the corresponding bit of $\overline{ID}$ with probability $\frac{3}{4}$. Thus we have

$$Adv_{\mathcal{A}}^{\mathsf{UNT}}(t,1) = |Pr[d = b] - \frac{1}{2}| = \frac{3}{4} - \frac{1}{2} = \frac{1}{4} > \varepsilon.$$

Thus, the David-Prasad protocol in an RFID system (S= $\{R_i, \mathcal{T}_0, \mathcal{T}_1, \dots\}$) in which a passive adversary $\mathcal{A}$ only eavesdrops a single run of the protocol (modeled by one Execute query in the game $\mathcal{G}$), is vulnerable to the most simple and effective traceability attack conceivable.

4 Leakage of Stored Secrets

Aside from traceability problems, the David-Prasad protocol also leaks out its long-term stored secrets, notably the static identifier ID and secret keys K_1, K_2. Generalizing our above analysis, specifically the Phase 1 of the traceability attack, if we denote by k the bitlength[5] of ID, then the full static identifier ID can be recovered with probability $\left(\frac{3}{4}\right)^{-k}$. This leaks out too many bits of ID, and seriously threatens the anonymity of the tag.

An attack to leak out information on the stored secret keys works as follows. The adversary can make the queries Execute($\mathcal{R}$, $\mathcal{T}_0$, $i-1$), Execute($\mathcal{R}$, $\mathcal{T}_0$, i) for two consecutive sessions, to obtain the pseudonyms $X^{i-1} = P^{\mathcal{T}_0}_{ID2}(i-1)$, $X^i = P^{\mathcal{T}_0}_{ID2}(i)$ and messages $\{A_{i-1}, B_{i-1}, D_{i-1}, E_{i-1}, F_{i-1}\}$, $\{A_i, B_i, D_i, E_i, F_i\}$, respectively. From equation (7), we see that X^{i-1}, X^i allows us to compute the XOR between the two nonces $\{n_1, n_2\}$ of the ith session:

$$\begin{aligned} Y &= X^{i-1} \oplus X^i \\ &= n_1 \oplus n_2. \end{aligned}$$

Furthermore, the adversary can compute the XOR of A_i and B_i:

$$\begin{aligned} Z &= A_i \oplus B_i \\ &= ((P^{\mathcal{T}_0}_{ID2}(i) \wedge K_1 \wedge K_2) \oplus n_1) \oplus ((\overline{P^{\mathcal{T}_0}_{ID2}(i)} \wedge K_2 \wedge K_1) \oplus n_2) \\ &= (K_1 \wedge K_2) \oplus n_1 \oplus n_2. \end{aligned}$$

Thus, the adversary obtains

$$Y \oplus Z = K_1 \wedge K_2$$

Note that for those bits where $K_1 \wedge K_2$ is 1, this implies that both key bits are 1. Consequently, on average $(\frac{k}{4})$ bits of both keys will be retrieved after two sessions. These observations have great security implications, and can be further explored and refined to disclose even more information, but this is no longer necessary in view of the following full disclosure attack.

5 A Passive Tango Cryptanalysis

In this section we present a novel passive (i.e. completely realistic in the underlying security model) and extremely efficient attack to fully recover both the secret

[5] David and Prasard assume that the bitlength of all variables is set to 96.

key values $\{K_1, K_2\}$ and the static identifier of the tag ID, which are indeed all the secret information the protocol is designed to conceal. The attack is divided into two main phases: 1) Selection of good approximations; and 2) Combination of the thus obtained good approximations for disclosing K_i or ID. We describe each of these phases below.

Phase 1: The attack exploits the leakage of secret information over the insecure radio channel due to fact that exchanged messages are derived from secret values by using triangular functions [5] only. Triangular operations and their composition (which is also triangular) are well known to have very poor diffusion properties. This is why the attacker can check and succeed in using multiple simple combinations of the exchanged public messages $\{A, B, D, E, F\}$ as Good Approximations (GA) for the secrets K_i or ID. Public exchanged messages do not hide well enough these secret values. From all the set of approximations, the adversary is interested on those that are systematically closer (on average) to the target secret value $X \in \{K_1, K_2, ID\}$. That is, those for which the Hamming distance between an approximation Z and the value X deviates from the expected value $\frac{96}{2}$, so either $hw(Z, X) < 48$ or $hw(\overline{Z}, X) < 48$.[6] In Appendix A, we list the average Hamming distance $dist(X, \cdot)$ of all possible combinations of the exchanged messages to the secrets. We present in the following table the best approximations for each of the three secret values we want to retrieve, which are the ones we employ in our attack:

Target	**Good Approximations**
K_1	$GA\text{-}K_1 = \{D, F, (A \oplus D), (\overline{A \oplus F}), (\overline{B \oplus D}), (B \oplus F),$ $(A \oplus B \oplus D), (A \oplus B \oplus F)\}$
K_2	$GA\text{-}K_2 = \{D, F, (\overline{A \oplus D}), (A \oplus F), (B \oplus D), (\overline{B \oplus F}),$ $(A \oplus B \oplus D), (A \oplus B \oplus F)\}$
ID	$GA\text{-}ID = \{(\overline{E \oplus F}), (A \oplus B \oplus E), (A \oplus D \oplus E),$ $(A \oplus E \oplus F), (B \oplus D \oplus E), (D \oplus E \oplus F), (\overline{A \oplus B \oplus D \oplus E}),$ $(A \oplus D \oplus E \oplus F), (\overline{B \oplus D \oplus E \oplus F})\}$

Phase 2: The basic idea in this phase of the attack is to combine multiple approximations (i.e. $Z \in \{GA\text{-}K_1, GA\text{-}K_2, GA\text{-}ID\}$)) obtained in different sessions, to construct a global one which is highly correlated with the secret values (i.e. keys $\{K_1, K_2\}$, and static identifier ID). This can be done in a number of different ways and forms, but in the case of the David-Prasad protocol a very simplistic approach works quite nicely. The way we proceed is the following: For each authentication session eavesdropped, we compute a number of good approximations to the secret values, and then store them as rows of three different matrices (one for each of K_1, K_2 and ID). After eavesdropping a given number of sessions, we compute the global values just by repeatedly adding each of the columns of the matrices, and returning a **0**

[6] We assume a bitlength of 96 for each of K_i, ID [3].

if the total number of ones in the said column is below a given threshold γ, or a **1** in any other case. In Figure 1, we provide a simple numerical example to further describe the attack, where the bitlength of the involved variables has been set to only 8 bits. The procedure to retrieve $\{K_1, K_2\}$ is very similar. The adversary has to provide a conjecture of the static identifier ID or the key K_i after the eavesdropping of some sessions. In each of them, multiple approximations of the pursued value are obtained – each of these approximations represent a row in the corresponding matrix. The simplest way to obtain a final value is to select the majority value in each column of this matrix. We can quickly sum all the rows to obtain a final vector. Then, if the value in a column of this vector is greater than half of the number of approximations N_A times the number of eavesdropped sessions N_S, we conjecture a **1** in that column. Otherwise, we conjecture a **0**. We can define that in a more formal way: Let be X and Y two vectors and x_i and y_i the value in each column of these vectors respectively. If the vector X is the input of the threshold function $th(X)$, the resulting vector is defined by:

$$th(X) = \begin{cases} \text{if } (X_i \geq \gamma) & Y_i = 1 \\ \text{if } (X_i < \gamma) & Y_i = 0 \end{cases} \quad \text{where } \gamma = 0.5 * N_A * N_S$$

This extremely easy and efficient way of combining approximations works surprisingly well for producing very accurate global approximations to all three secret values after eavesdropping a relatively small number of authentication sessions. The results are presented in the following figures.
We have simulated our attack to evaluate its feasibility and effectiveness. First, we randomly initialize the secret values (i.e. $\{P_{ID}, P_{ID2}, K_1, K_2, ID\}$). Then, we simulate N_S legitimate sessions of the protocol – the attacker eavesdrops N_S sessions – and we run the adversary's strategy (Phase 2) to obtain a conjecture of the keys $\{K_1, K_2\}$ and the static identifier ID. Finally, we compare the global conjecture value $X_{conjecture} \in \{K_{1_{conjecture}}, K_{2_{conjecture}}, ID_{conjecture}\}$ with the real value $X \in \{K_1, K_2, ID\}$ to measure the adversary's success. The mean and standard deviation of the number of bits successfully recovered, for various values of eavesdropped sessions (N_S), are summarized in Figures 2, 3 and 4. In our simulations, the bitlength of variables is set to 96 and for each value of N_S we repeated the experiment 10.000 times. For $\{K_{1_{conjecture}}, K_{2_{conjecture}}, ID_{conjecture}\}$, the threshold is set to $\{0.5 * 8 * N_S, 0.5 * 8 * N_S, 0.5 * 9 * N_S\}$ respectively, which means that in all cases we are guessing the majority value between those observed.

As we are using the same number of approximations (8 for every eavesdropped session) for K_1 and K_2, and they are similarly powerful, the results obtained are quite close. In both cases, the number of required eavesdropped sessions by an attacker to disclose the full secret key K_i is less than or equal to 65. The effectiveness of this attack in disclosing the static identifier ID is slightly superior in comparison, partly due to the fact that in this case 9 approximations – instead of 8 – are used. For the ID, the adversary needs only around 50 sessions to completely disclose the full 96 bits of the static identifier. Even though these

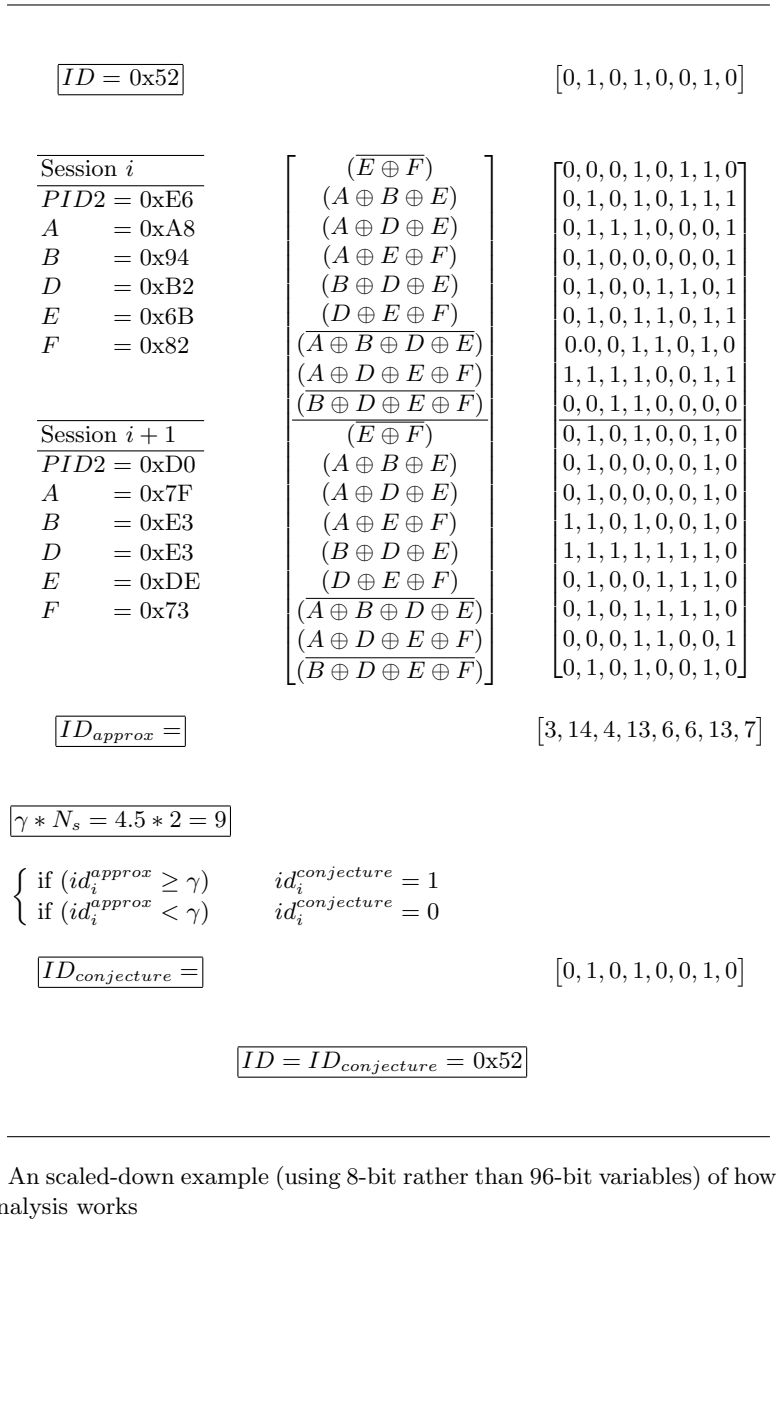
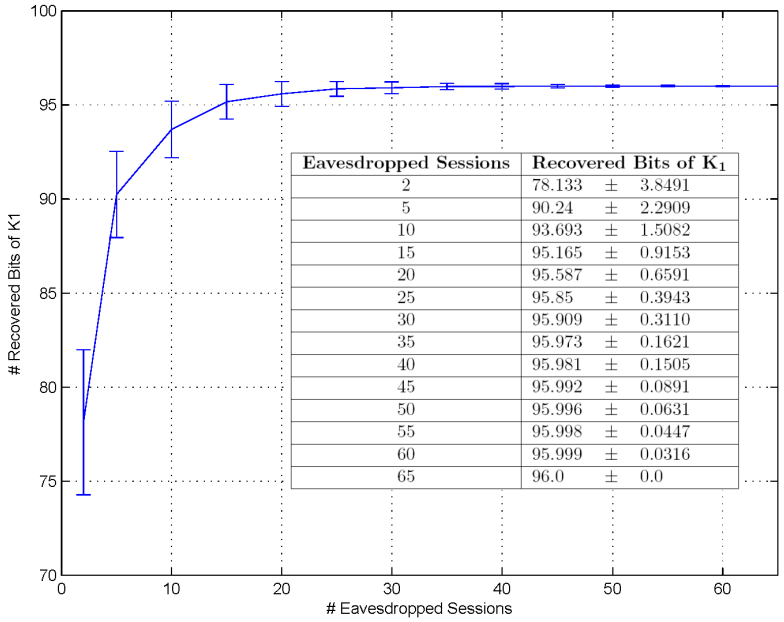
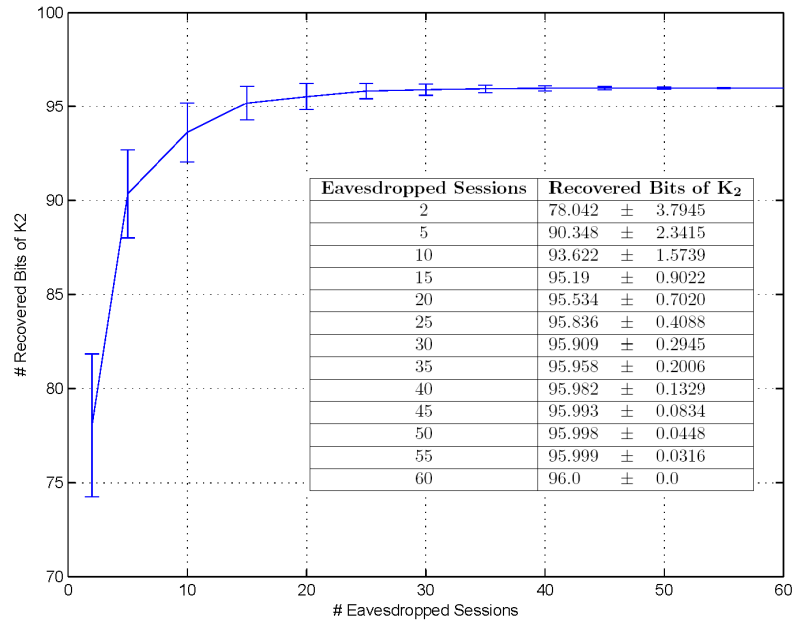


Fig. 1. An scaled-down example (using 8-bit rather than 96-bit variables) of how Tango cryptanalysis works



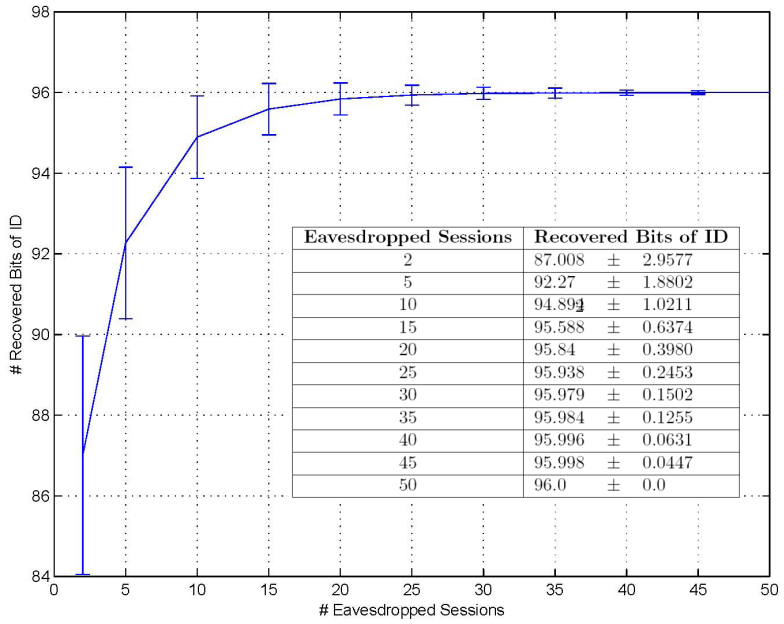
Eavesdropped Sessions	Recovered Bits of K_1
2	78.133 ± 3.8491
5	90.24 ± 2.2909
10	93.693 ± 1.5082
15	95.165 ± 0.9153
20	95.587 ± 0.6591
25	95.85 ± 0.3943
30	95.909 ± 0.3110
35	95.973 ± 0.1621
40	95.981 ± 0.1505
45	95.992 ± 0.0891
50	95.996 ± 0.0631
55	95.998 ± 0.0447
60	95.999 ± 0.0316
65	96.0 ± 0.0

Fig. 2. K_1 bit recovery, against the # of eavesdropped sessions



Eavesdropped Sessions	Recovered Bits of K_2
2	78.042 ± 3.7945
5	90.348 ± 2.3415
10	93.622 ± 1.5739
15	95.19 ± 0.9022
20	95.534 ± 0.7020
25	95.836 ± 0.4088
30	95.909 ± 0.2945
35	95.958 ± 0.2006
40	95.982 ± 0.1329
45	95.993 ± 0.0834
50	95.998 ± 0.0448
55	95.999 ± 0.0316
60	96.0 ± 0.0

Fig. 3. K_2 bit recovery, against the # of eavesdropped sessions



Eavesdropped Sessions	Recovered Bits of ID
2	87.008 ± 2.9577
5	92.27 ± 1.8802
10	94.894 ± 1.0211
15	95.588 ± 0.6374
20	95.84 ± 0.3980
25	95.938 ± 0.2453
30	95.979 ± 0.1502
35	95.984 ± 0.1255
40	95.996 ± 0.0631
45	95.998 ± 0.0447
50	96.0 ± 0.0

Fig. 4. ID bit recovery, against the # of eavesdropped sessions

figures are more than enough to consider the protocol completely broken, we also note that a more constrained attacker is not forced to evasdrop such a number of sessions to fully recover the 96 bits: After only 5 or 10 sessions, more than 90 bits are correctly guessed, and the remaining can be easily identified by an offline brute force search.

The attacks just presented have serious consequences for the overall security of the protocol. In fact, they utterly ruin all the security properties claimed by their authors. After conducting the attack, the adversary is able to retrieve all the secret information shared between the tag and the server, so he can trivially bypass any authentication mechanisms (i.e. tag and reader authentication) and impersonate the tag in the future, or just clone it. Confidential information is put at risk and tag's answers can be tracked even though two random numbers are used in each session in a failed attempt to stop this from happening. A desynchronization attack against the tag (or the sever) is also quite straightforward, since the adversary can generate any desired valid synchronization messages.

6 Conclusions

The design of ultralightweight security protocols for low cost RFID tags is a stimulating challenge due to the severe computational restrictions of these devices. Although interesting proposals have recently been published in this research

area, the design of secure schemes is still an open question. If fact, the vast majority of the published schemes are already broken.

Triangular functions are very attractive because they can be efficiently implemented in hardware, but a cryptanalyst can take advantage of their use due to their very poor diffusion properties. So they can and probably should be used, but not alone, as the composition of triangular functions is still triangular. They should be combined with non-triangular functions – as proposed in SASI [1] – to hinder the task of breaking the scheme. Rotation operations are a quite interesting possibility as they are not triangular, allow to amplify diffusion, and are also very efficient to implement in hardware. If we had to single out the main reason for the weaknesses found in the David-Prasad protocol, apart from the design of some messages, this would definitely be the non inclusion of any kind of rotations (Hamming based or modular) in the set of operations used. The inclusion of nonces is very likely a necessary condition to guarantee anonymity, but by itself does not ensure this desirable property, or any protection against traceability attacks.

We do not claim that the attacks and techniques presented here are optimal in any way, and can conceivably design more subtle and maybe slightly more powerful attacks, but we believe that in the light of the results offered here there is no need for that. However, possibly a mixture of the approximation to the ID obtained in Section 4, combined with the approximations used in the Tango attack might lead to a slightly more efficient approach.

The cryptanalytic technique introduced in this paper, named Tango attack, could also be seen as a new tool to analyze lightweight protocols, and thus helpful in the design of more secure future proposals. We believe it will prove successful against other lightweight protocols and algorithms because, almost by definition, they do not have in many cases the computational resources needed to allow for an adequate (i.e. highly nonlinear) mixture of the internal secret values as to avoid leaking some bits in every session.

References

1. H.-Y. Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Trans. Dependable Secur. Comput.*, 4(4):337–340, 2007.
2. M. David and N. R. Prasad. Providing Strong Security and High Privacy in Low-Cost RFID Networks. In *Proc. of Security and Privacy in Mobile Information and Communication Systems, MobiSec'09*, pages 172–179. Springer Berlin Heidelberg, September 2009.
3. EPCglobal. Class-1 Generation 2 UHF Air Interface Protocol Standard Version 1.2.0: Gen 2. http://www.epcglobalinc.org/standards/, 2008.
4. A. Juels and S. Weis. Defining strong privacy for RFID. In *Proc. of PerCom 2007*, pages 342–347. IEEE Computer Society Press, 2007.
5. A. Klimov and A. Shamir. New Applications of T-Functions in Block Ciphers and Hash Functions. In *Proc. of FSE'05*, volume 3557 of *LNCS*, pages 18–31. Springer-Verlag, 2005.

6. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In *Proc. of WISA'08*, volume 5379 of *LNCS*, pages 56–68. Springer-Verlag, 2008.
7. P. Peris-Lopez, J.C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags. In Hand. of RFIDSec'06, 2006.
8. P. Peris-Lopez, J.C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags. In *Proc. of UIC'06*, volume 4159 of *LNCS*, pages 912–923. Springer-Verlag, 2006.
9. P. Peris-Lopez, J.C. Hernandez-Castro, Juan M. Estevez-Tapiador, and A. Ribagorda. EMAP: An Efficient Mutual Authentication Protocol for Low-cost RFID Tags. In *Proc. of IS'06*, volume 4277 of *LNCS*, pages 352–361. Springer-Verlag, 2006.
10. R. Phan. Cryptanalysis of a new ultralightweight RFID authentication protocol - SASI. *Dependable and Secure Computing, IEEE Transactions on*, DOI: 10.1109/TDSC.2008.33, 2008.

APPENDIX

A Approximations to K_1, K_2, and ID (10.000 tests)

$\mathbf{X}$	$dist(\mathbf{X}, \mathbf{K_1})$	$dist(\mathbf{X}, \mathbf{K_2})$	$dist(\mathbf{X}, \mathbf{ID})$
A	49.4 ± 1.8547	48.3 ± 4.3829	49.3 ± 5.1196
B	49.4 ± 5.0990	48.3 ± 6.2578	49.3 ± 3.9560
D	$\mathbf{34.0 \pm 1.9493}$	$\mathbf{35.1 \pm 3.8587}$	52.4 ± 3.8000
E	47.8 ± 4.284	46.2 ± 4.6861	49.3 ± 4.1485
F	$\mathbf{36.1 \pm 3.3600}$	$\mathbf{35.6 \pm 3.1686}$	50.8 ± 5.0160
$A \oplus B$	48.6 ± 4.055	47.9 ± 5.1662	49.0 ± 3.7148
$A \oplus D$	$\mathbf{37.2 \pm 3.4293}$	$\mathbf{61.6 \pm 2.2000}$	48.7 ± 2.9343
$A \oplus E$	42.8 ± 3.628	48.3 ± 2.052	50.6 ± 4.3174
$A \oplus F$	$\mathbf{61.3 \pm 3.769}$	$\mathbf{37.7 \pm 4.6054}$	48.9 ± 3.0806
$B \oplus D$	$\mathbf{61.8 \pm 4.3543}$	$\mathbf{36.9 \pm 4.2532}$	47.1 ± 3.4771
$B \oplus E$	47.6 ± 3.8262	47.8 ± 3.1874	47.6 ± 7.1722
$B \oplus F$	$\mathbf{37.7 \pm 2.6851}$	$\mathbf{60.8 \pm 4.5343}$	46.9 ± 2.3000
$D \oplus E$	42.6 ± 2.9732	45.7 ± 3.5228	52.3 ± 5.3675
$D \oplus F$	47.1 ± 1.9723	46.7 ± 4.0509	51.6 ± 2.8355
$E \oplus F$	41.9 ± 4.5705	56.2 ± 4.1665	$\mathbf{67.7 \pm 5.4598}$
$A \oplus B \oplus D$	$\mathbf{37.6 \pm 5.8173}$	$\mathbf{36.8 \pm 2.4000}$	48.2 ± 5.8617
$A \oplus B \oplus E$	56.0 ± 2.1448	44.5 ± 3.4132	$\mathbf{24.5 \pm 3.6946}$
$A \oplus B \oplus F$	$\mathbf{35.5 \pm 3.2939}$	$\mathbf{36.3 \pm 3.0348}$	49.8 ± 3.6824
$A \oplus D \oplus E$	47.2 ± 3.1875	38.4 ± 3.9294	$\mathbf{35.8 \pm 4.9759}$
$A \oplus D \oplus F$	47.5 ± 3.5284	47.0 ± 5.0398	50.3 ± 6.4195
$A \oplus E \oplus F$	48.5 ± 3.3838	48.1 ± 2.6627	$\mathbf{22.2 \pm 1.7205}$
$B \oplus D \oplus E$	51.2 ± 4.7286	45.7 ± 3.1953	$\mathbf{34.0 \pm 3.7947}$
$B \oplus D \oplus F$	49.9 ± 4.5706	47.5 ± 4.7802	47.5 ± 3.4424
$B \oplus E \oplus F$	49.9 ± 5.1662	45.6 ± 4.200	47.6 ± 6.9022
$D \oplus E \oplus F$	50.3 ± 3.9762	45.3 ± 4.5177	$\mathbf{31.1 \pm 3.5903}$
$A \oplus B \oplus D \oplus E$	47.6 ± 4.5211	55.4 ± 4.8208	$\mathbf{61.1 \pm 4.3920}$
$A \oplus B \oplus D \oplus F$	44.5 ± 3.9812	49.2 ± 3.3106	49.4 ± 3.555
$A \oplus B \oplus E \oplus F$	48.3 ± 5.2354	44.9 ± 5.6648	45.7 ± 5.0408
$A \oplus D \oplus E \oplus F$	44.9 ± 3.8066	40.6 ± 2.7276	$\mathbf{35.8 \pm 6.1449}$
$B \oplus D \oplus E \oplus F$	45.5 ± 1.8028	55.5 ± 4.7592	$\mathbf{62.4 \pm 2.7276}$
$A \oplus B \oplus D \oplus E \oplus F$	53.5 ± 5.0843	45.4 ± 5.5534	42.7 ± 3.06757