# Non-standard Attacks against Cryptographic Protocols, with an Example over a Simplified Mutual Authentication Protocol

Julio C. Hernandez-Castro, Juan M.E. Tapiador, and Arturo Ribagorda

Department of Computer Science, Carlos III University of Madrid
{jcesar,jestevez,arturo}@inf.uc3m.es

**Abstract.** In this work, we present a simple model for the automated cryptanalysis of cryptographic protocols based on meta-heuristic search. We illustrate our approach with a straightforward application in the form of an attack against a slightly simplified version of an ultra-lightweight authentication protocol for RFID environments called SASI. We show how an attack based on Simulated Annealing can efficiently recover the tag's secret ID, which is the value the protocol is designed to conceal.

## 1 Introduction

In recent years there has been a proliferation of cryptographic protocols aimed at providing security services in very constrained environments. The most common examples are mutual authentication schemes for Radio Frequency IDentification (RFID) systems, where the shortage of computational resources in tags makes impossible to apply classical constructions based on cryptographic primitives such as block/stream ciphers or hash functions.

Typical proposals are often forced to consist of a number of steps in which very simple operations are performed over public and private values. In this work, we present a general model for the automated cryptanalysis of such schemes based on meta-heuristic search. We will illustrate our idea with an application against a simplified version of one of the most prominent protocols proposed so far: an ultra-lightweight authentication protocol for RFID environments called SASI. We will be able to show how an attack based on a Simulated Annealing technique can efficiently recover the tag's secret ID, which is the value the protocol is designed to conceal.

The idea of attacking cryptographic protocols by means of heuristic procedures is not new: Clark et al. [4] presented a seminal work in this area where they were able to break the PPP [8] identification protocol. It is, however, true that since that work no further progress has been made in the field. Additionally, the technique employed here is quite different from that used by Clark et al. The related area of evolving or automatically designing cryptographic protocols by means of different heuristic techniques has seen, on the other hand, considerable success [5].

H.A. Le Thi, P. Bouvry, and T. Pham Dinh (Eds.): MCO 2008, CCIS 14, pp. 589–596, 2008.

The rest of the paper is organized as follows. In the next section, we present our general model proposal for non-standard attack of cryptographic protocols. After this, in Section 3 we describe a novel authentication protocol for RFID environments called SASI, together with its simplified variant CR-SASI, which succumbs to the attack introduced and analyzed in Section 4. Finally, in Section 5 we extract some conclusions.

## 2 General Attack Model

The main idea behind our approach is to transform the cryptanalysis of a security protocol into a search problem, where search heuristics can be applied. In general, during this search what we will try to find are the secret state values (keys, nonces, etc.) of some subset of the involved parties. This, of course, could be done in various ways, but the most natural approach is to measure the fitness of the tentative secret values by the proximity of the messages produced by these tentative solutions to the real public messages generated and exchanged during the actual protocol execution.

Most cryptographic protocols should exchange one or more messages to accomplish their intended objective (authentication, key exchange, key agreement, etc.), and in the vast majority of cases these messages are sent via an insecure or public channel that can be easily snooped.

In our attack model, the cryptanalyst will try to infer the secret values that the two parties are trying to hide by exploiting the knowledge of the exchanged messages. In an robust, secure and well-designed cryptographic protocol, even states that are very close to the real state should not produce messages that are very close (for any useful distance definition) of the real public messages.

This should be done, typically, by means of a carefully design and message construction based on the use of some highly nonlinear cryptographic primitives such as block ciphers or hash functions. Weaknesses in a protocol design could lead, on the other hand, to the lack of this desirable property, a fact that can be exploited to mount a non-standard cryptanalytic attack based in some kind of heuristic search guided by a fitness measuring distances between exchanged and computed messages.

This shall exactly be the approach followed in the rest of the paper.

## 3 Description of the SASI Protocol

In 2007, Chien presented an interesting ultra-lightweight mutual authentication protocol providing Strong Authentication and Strong Integrity (SASI), intended for very low-cost RFID tags [1]. This was a much needed answer to the increasing need for schemes providing such properties in extremely constrained environments like RFID systems. As all the previous attempts to design ultralightweight protocols have failed (essentially all proposals have been broken), this new scheme was specially interesting.

The SASI protocol is briefly described in the following, where $R$ represents the reader, $T$ represents the tag, $IDS$ stands for an index pseudonym, $ID$ is tag's private ID, $K_i$ represent tag's secret keys and $n_1$ and $n_2$ are nonces. Each message takes the form $A \rightarrow B : m$, meaning that sender $A$ sends to receiver $B$ message $m$.

1. $R \rightarrow T : hello$
2. $T \rightarrow R : IDS$
3. The reader uses $IDS$ to find in the back-end database the tag's secret values $ID$, $K_1$, and $K_2$. Then $R$ generates nonces $n_1$ and $n_2$ to construct messages $A$, $B$ and $C$ as follows:
   $A = IDS \oplus K_1 \oplus n_1$
   $B = (IDS \vee K_2) + n_2$
   $C = (K_1 \oplus \bar{K_2}) + (K_2 \oplus \bar{K_1})$, where
   $\bar{K_1} = Rot(K_1 \oplus n_2, K_1)$
   $\bar{K_2} = Rot(K_2 \oplus n_1, K_2)$
   where $\oplus$ stands for the usual addition modulo 2, $+$ represents addition modulo $2^{96}$, $Rot(A, B) = A << wht(B)$ with $wht(B)$ the Hamming weight of $B$, and $\vee$ is the usual bitwise or operation. Finally, the reader sends to the tag the concatenation of $A$, $B$ and $C$:

   $R \rightarrow T : A \| B \| C$

4. From $A$ and $B$,respectively, the tag can obtain values $n_1$ and $n_2$. Then it locally computes $C$ and checks if the result of its local computation is equal to the sent value. If this is the case, it updates the values of $IDS$, $K_1$ and $K_2$ in the following manner:

   $IDS^{next} = (IDS + ID) \oplus (n_2 \oplus \bar{K_1})$
   $K_1^{next} = \bar{K_1}$
   $K_2^{next} = \bar{K_2}$

5. $T \rightarrow R : D$, where $D = (\bar{K_2} + ID) \oplus ((K_1 \oplus K_2) \vee \bar{K_1})$
6. Finally, $R$ verifies $D$ and, if it is equal to the result of its local computation, it updates $IDS, K_1$ and $K_2$.

SASI has received no serious attacks yet, except for a couple of minor weaknesses that could be employed to mount two desynchronization scenarios [2].

### 3.1 CR-SASI: The Simplified SASI Protocol

CR-SASI is the simplified version of the SASI protocol we will cryptanalyze in the paper. It is essentially identical to the published version, but for two minor differences:

1. CR-SASI is a scaled-down version of SASI, which operates over $\mathbb{Z}_2^{32}$ while SASI does it over elements in $\mathbb{Z}_2^{96}$
2. CR-SASI uses constant distance rotations, instead of the Hamming-dependant rotations proposed by Chien.

Any amount of non-zero rotation produces a similarly robust protocol, but for reasons explained below we have fixed this rotation amount to $\frac{32}{2} = 16$. We have experimentally found, anyway, that any other value produces a protocol that is breakable in essentially the same way. An important observation is that the amount of the rotation operation, as originally defined, is far from being uniform. In fact, if we assume this second argument $B$ to be random, then the probability that the rotation amount takes value $k$ is given by the formula:

$$Prob(wht(B) = k) = \frac{\binom{32}{k}}{2^{32}} \tag{1}$$

which attains a maximum for $k = \frac{32}{2} = 16$ with an associated probability of 0.139949, or around 14% of the times. This additionally justifies our chosen value (as it will be the most common) for the amount of left rotation in CR-SASI.

All in all, these two modifications should not greatly modify the security characteristics of the underlying protocol, so the study of this variant is relevant for understanding the security of the whole SASI protocol. It is important to notice that constant distance rotations, such as that used in CR-SASI, are usually part of cryptographic primitives and protocols, so they could have been part of the original protocol proposal as they were components of modern cryptographic primitives such as TEA[6], XTEA[9] and Salsa20[7], to name a few.

## 4 Cryptanalysis of the SASI Protocol

In the light of the equations that define the protocol (see Section 3) we can initially see that the internal secret state we will look for is formed of the values:

$$State = \{K_1, K_2, n_1, n_2, ID\}$$

Assuming that message $A$ is known (as it is the case), it can be seen than $K_1$ and $n_1$ are related ($n_1 = A \oplus IDS \oplus K_1$), so we can reduce the state size to $\{K_1, K_2, n_2, ID\}$.

Analogously, from the knowledge of message $B$ we can conclude that $K_2$ and $n_2$ are also related, following the equation $n_2 = B - (IDS \vee K_2)$. We can therefore still reduce the state size to $State = \{K_1, K_2, ID\}$.

A further reduction is still possible, since $ID$ also depends on $\{K_1, K_2\}$, although in a more complex way, once $\{IDS^{next}, IDS\}$ or $D$ are known because:

$$ID = IDS^{next} \oplus (n_2 \oplus \bar{K_1}) - IDS \tag{2}$$

$$ID = D \oplus ((K_1 \oplus K_2) \vee \bar{K_1}) - \bar{K_2} \tag{3}$$

So we finally are left with a minimal state of the form $State = \{K_1, K_2\}$, where no further reduction is possible. This implies that our set of possible solutions are of the above described form, and have a size of $2^{64}$.

Note that among the public messages $\{IDS, A, B, C, D, IDS^{next}\}$ that can be observed after one authentication session, we have used all except $C$ and $D$ to reduce the state space. These two last messages will be the base of our fitness function.

Starting from a candidate state $\{K_1', K_2'\}$, and using public values $\{IDS, A, B, IDS^{next}\}$, we will compute the corresponding values for messages $C'$ and $D'$, and measure their distance to the known actual values of $C$ and $D$. We will try to minimize this distance in order to find values as close as possible to the real $\{K_1, K_2\}$ values. Different definitions of distance have been tried (euclidean, edit, weighted, etc.) and could be useful, but we have got the best results with the usual Hamming distance. In the general case of having $N$ public messages $M_1, \ldots, M_N$, the resulting fitness function is given by:

$$f_S = -\sum_{i=0}^{N} d_H(M_i, M_i') \tag{4}$$

where $M_i$ stands for the real (snooped) message and $M_i'$ is its approximation as computed from the values of state $S$.

For our particular problem, equation (4) will have the form:

$$f_S = -\big(d_H(C, C') + d_H(D, D')\big) \tag{5}$$

For this optimization process, we will use Simulated Annealing as heuristic. After extensive experimentation, the set of parameters which consistently lead to good results are those shown in Table 1.

### 4.1 Experimental Results

We have implemented various versions of the same cryptanalytic method, which start the cryptanalytic process after eavesdropping two, three or four consecutive rounds of the protocol, respectively. As expected, the knowledge of more authentication rounds leads to better attacks.

We have performed simulations for measuring the effectiveness of this approach. In all the cases, we initialized all secret and public values of the protocol to the first hexadecimal values of $\pi$, as taken from `http://www.super-computing.org/`. In particular, the state values we are looking for are fixed to $State = \{0x243F6A88, 0x85A308D3\}$.

Results for five different runs, after capturing data from only two consecutive authentication sessions are shown in Table 2.

It can be clearly seen that, although no solution is perfect, all of them are quite close to the real secret state values. In fact, it is very easy to compute

**Table 1.** Simulated Annealing parameters for the cryptanalysis of CR-SASI

| | |
|---|---|
| Initial Temperature | 10 |
| Cooling Rate | 0.99 |
| Max. Failed Cycles | 100 |
| Moves | 1000 |
| IC Max. | 500 |

the secret from these accurate approximations. Just a bitwise majority function (weighted by the fitness as provided in Table 2) will lead to an almost correct solution. Then, a simple trial and error process could be started to find the correct values. In case of any doubt, more runs should improve the accuracy of the attack.

We have experimentally found that about four or five runs are generally enough to get results very close to the secret values, as in the case of the above example. At worst, only $16^2$ additional trials are needed for recovering the correct keys.

Each of these runs takes approximately 800 seconds in a very modest laptop. Furthermore, it is important to stress that they are completely parallelizable. After obtaining the correct key values, the secret static $ID$ will be easily recovered using either equation (2) or (3). This will allow a fraudulent tag (with, say, altered prices or false stock information) to impersonate the legitimate tag, possibly corrupting the back-end database with false data, after eavesdropping only two consecutive authentication sessions.

The efficiency of this attack can be slightly improved just by observing more sessions. We have performed tests after three and four consecutive sessions following exactly the same approach described above, and the results were consistently better.

However, with more authentication sessions there are approaches that do not work after eavesdropping only two sessions that now become entirely possible. In this case, the best attacking strategy becomes to try to infer the secret $ID$ from the best approximation found for $\{K_1, K_2\}$ with the help of equations (6)–derived from (2))– and (7):

$$ID = IDS^{next} \oplus (n_2 \oplus \bar{K_1}) - IDS \tag{6}$$

$$= IDS^{next} \oplus ((B - (IDS \vee K_2')) \oplus \bar{K_1'}) - IDS \tag{7}$$

After 10 runs following this scheme, we obtained the exact value of the secret $ID$ three times (see Table 3), and very good approximations to it (with a Hamming distance to the real $ID$ of 8 or less) in another six occasions. The best approximation was very easy to identify because it has the best fitness within the 10 runs.

**Table 2.** Attack results for five runs, after capturing two authenication sessions

| | | |
|---|---|---|
| K1=0x24FF6B8E | K2=0x84E308D5 | Fitness=-7.000000 |
| K1=0x74300A88 | K2=0x35ACA8C3 | Fitness=-7.000000 |
| K1=0x347FCA88 | K2=0x35E368D3 | Fitness=-6.000000 |
| K1=0x343FCAC8 | K2=0x35E36893 | Fitness=-4.000000 |
| K1=0x243F2A88 | K2=0x85A348D3 | Fitness=-1.000000 |
| K1=0x243F?A88 | K2=0x85A3?8D3 | Majority weighted function |
| K1=0x243F6A88 | K2=0x85A308D3 | Real Values |

**Table 3.** Attack results for 10 runs, after capturing 4 authentication sessions

| Fitness | $d_H$ to $ID$ |
|---|---|
| -21 | 0 |
| -27 | 0 |
| -26 | 0 |
| -23 | 5 |
| -33 | 9 |
| -31 | 7 |
| -31 | 3 |
| -24 | 5 |
| -33 | 8 |
| -27 | 3 |

## 5 Concluding Remarks

In this paper, we have presented a new and efficient attack against a simplified version of a novel and interesting ultra-lightweight authentication protocol.

This attack is performed by means of a non-standard technique (SA-based) that we have shown as a particular instance of a more general attack methodology against cryptographic protocols.

We believe that more and more of these non-standard attacks will be sucessfully employed against the new lightweight protocols designed for very constrained environments such as RFID systems and some kinds of sensor networs, because in most of the cases they can't allow the use of standard cryptographic primitives.

Attacking the full SASI protocol with similar but improved techniques is a future and interesting research direction.

## References

1. Chien, H.-Y.: SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. IEEE Transactions on Dependable and Secure Computing 4(4), 337–340 (2007)
2. Sun, H.-M., Ting, W.-C., Wang, K.-H.: On the Security of Chien's Ultralightweight RFID Authentication Protocol. Cryptology ePrint Archive, `http://eprint.iacr.org/2008/083`
3. Klimov, A., Shamir, A.: New Applications of T-functions in Block Ciphers and Hash Functions. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557. Springer, Heidelberg (2005)
4. Clark, J.A., Jacob, J.L.: Fault Injection and a Timing Channel on an Analysis Technique. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 181–196. Springer, Heidelberg (2002)
5. Clark, J.A., Jacob, J.L.: Protocols are Programs Too: the Meta-heuristic Search for Security Protocols. Metheuristics for Software Engineering. Information Software Technology 43(14), 891–904 (2001)

6. Wheeler, D.J., Needham, R.M.: TEA, a tiny encryption algorithm. In: Fast Software Encryption: Second International Workshop, Leuven, Belgium. LNCS, vol. 1008, pp. 363–366. Springer, Heidelberg (1994)
7. Bernstein, D.J.: The Salsa20 stream cipher, slides of talk at ECRYPT STVL Workshop on Symmetric Key Encryption (2005), `http://cr.yp.to/talks.html#2005.05.26`
8. Pointcheval, D.: A New Identification Scheme Based on the Perceptron Problems. In: Advances in Cryptology Eurocrypt 1995. LNCS, vol. 2199. Springer, Heidelberg (1995)
9. Needham, R.M., Wheeler, D.J.: Tea extensions. Technical report, Computer Laboratory. University of Cambridge, Cambridge (October 1997)