# Secret Disclosure attack on Kazahaya, a Yoking-Proof For Low-Cost RFID Tags

Nasour Bagheri[1], Masoumeh Safkhani[3]

[1] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Iran NBagheri@srttu.edu
[2] Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran
M_Safkhani@iust.ac.ir

**Abstract.** Peris-Lopez *et al.* recently provides some guidelines that should be followed to design a secure yoking-proof protocol [10]. In addition, conforming to those guidelines and EPC C1 G2, they presented a yoking-proof for low-cost RFID tags, named Kazahaya. However, in this letter, we scrutinize its security showing how an passive adversary can retrieve secret parameters of patient's tag in cost of $O(2^{16})$ off-line $PRNG$ evaluations. Given the tag's secret parameters, any security claims are ruined. Nevertheless, to show other weaknesses of the protocol and rule out any possible improvement by increasing the length of the used $PRNG$, we presented a forgery attack that shows that a proof generated at time $t_n$ can be used to forge a valid proof for any desired time $t_j$. The success probability of this attack is '1' and the complexity is negligible.

**keywords:** RFID, Authentication, Yoking-Proof, Cryptanalysis.

## 1 Introduction

Radio Frequency IDentification (RFID), which is a technology that enables identification from distance [14], is already used for a large number of different applications, from cards used for building access or payments with mobile devices [9] to applications in sanitary environments [1].

A typical RFID system consists of three main components: tags, readers and database. Generally, tags are small devices with high constraint on memory, computation and storage resources, which employ a challenge for supporting security capabilities. Readers are devices with less computation and memory constraints (compared with tags) and communicate with the tags and a database and the database is used to store information to authenticate tags in the system (extra information linked with each tag can be stored too).

Over the last decade several RFID authentication protocols, which authenticate a reader to a tag or vise versa, have been proposed in the literature [13]. However, Juels [7] introduced the novel problem of evidencing that two tags have been simultaneously read, which has many potential applications, e.g. in health-care sector. He called this kind of evidence a yoking-proof, which is supposed to be verifiable off-line. Later, other researchers generalized the proof for a larger number of tags [12], this type of protocols also known as grouping-proof protocols.

Recently in [10] Peris-Lopez *et al.* have analyzed several recent proposals in the context [2–5,7,8,12] and shown their vulnerabilities. In addition, they have provided a list of tips that to be followed by a protocol's designer to preclude past errors and design a secure protocol. Moreover, following those guidelines, they have designed a novel yoking-proof, named Kazahaya. This is an EPC C1 G2 [6] compliant solution and designed for the low-cost RFID tags. However, in this letter we scrutinize its security showing how an passive adversary can retrieve secret parameters of tag in cost of $O(2^{16})$ off-line $PRNG$ evaluations. Hence, this new proposal also does not provide the expected security.

In the rest of this letter, we review Kazahaya protocol in Section 2 and present our secret disclosure attack against it in Section 3. In Section 4 we present a practical forgery attack against the protocol. Finally, we conclude the paper in Section 5.

| Notation | Description |
|---|---|
| $T_x$ | A tag indexed by $x$ |
| $r$ | Random number |
| $PRNG$ | 16 bit pseudo random number generator |
| $\oplus$ | Exclusive or operation |
| $ID_{group}$ | Unique Identifier of a group of tags |
| $ID_{T_x}$ | Unique Identifier of $T_x$ |
| $K_{group}$ | Private group key of a group of tags |
| $K_{T_x}$ | Private key of $T_x$ |
| $t_n$ | Timestamp |

**Table 1.** Notation

## 2 Review of Kazahaya Protocol

Kazahaya Protocol recently has been proposed by Peris *et al.* [10], to prevent the participation of unrelated tags in a proof. In this protocol, where to explain it we use the notations indicated in Table 1, tags are divided in groups and each group is identified by a group identifier $ID_{group}$ and a group key $K_{group}$. Moreover, each tag has a unique identifier $ID_{T_i}$ and a private key $K_{t_n}$. For each tag, the backend data base stores the tuple $\{ID_{T_i}, ID_{group}, K_{T_i}, K_{group}\}$. Kazahaya yoking-proof, as depicted in Fig. 1, runs as follows:

1. The reader queries $T_a$ by sending timestamp $t_n$.
2. Upon receiving the message, $T_a$ does as follows:
   - generates two random numbers $r_{T_a}$ and $r'_{T_a}$,
   - computes $M^1_{group}$ and $M_{T_a}$ as below:
     $M^1_{group} = PRNG(ID_{group} \oplus r_{T_a} \oplus PRNG(K_{group}) \oplus$
     $PRNG(t_n))$
     $M_{T_a} = PRNG(ID_{T_a} \oplus r'_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_n + 1))$
   - and sends $\{r_{T_a}, r'_{T_a}, M^1_{group}, M_{T_a}\}$ to the reader.
3. The reader stores $r'_{T_a}$ and sends $\{t_n, r_{T_a}, M^1_{group}, M_{T_a}\}$ to $T_b$.

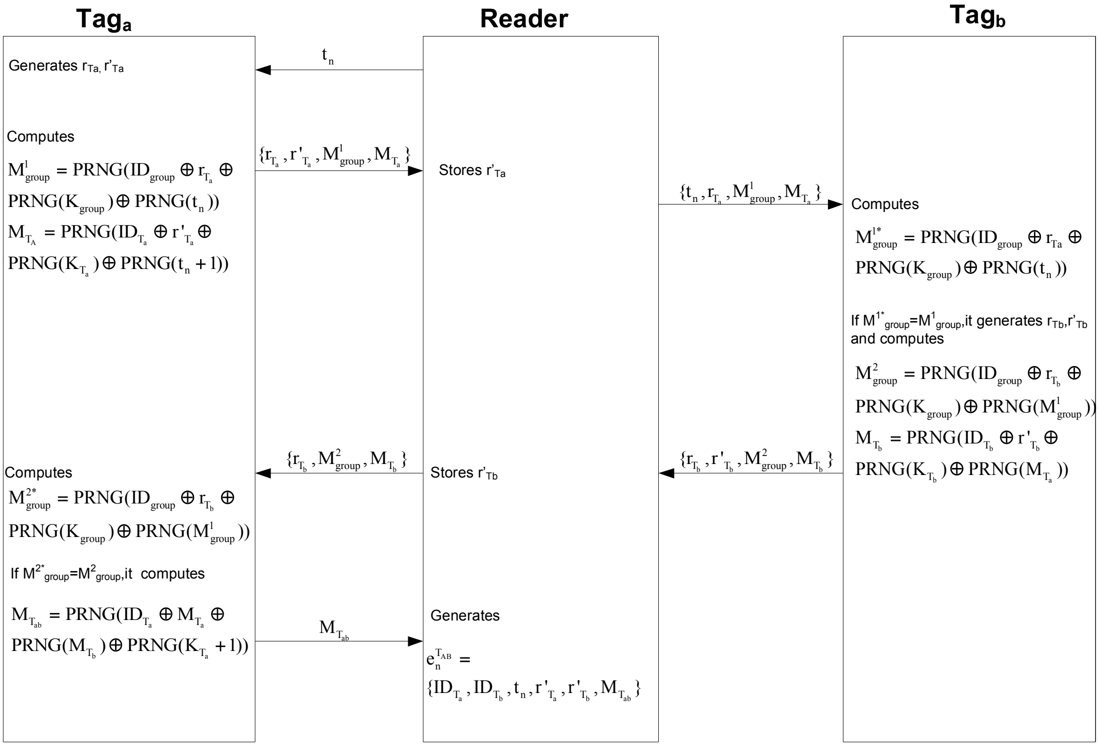
$\mathrm{Tag_a}$
Reader
$\mathrm{Tag_b}$
$t_n$
Generates $r_{Ta}, r'_{Ta}$
Computes
$M^1_{group} = PRNG(ID_{group} \oplus r_{T_a} \oplus PRNG(K_{group}) \oplus PRNG(t_n))$
$M_{T_A} = PRNG(ID_{T_a} \oplus r'_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_n + 1))$
$\{r_{T_a}, r'_{T_a}, M^1_{group}, M_{T_a}\}$
Stores $r'_{Ta}$
$\{t_n, r_{T_a}, M^1_{group}, M_{T_a}\}$
Computes
$M^{1*}_{group} = PRNG(ID_{group} \oplus r_{Ta} \oplus PRNG(K_{group}) \oplus PRNG(t_n))$
If $M^{1*}_{group} = M^1_{group}$,it generates $r_{Tb}, r'_{Tb}$ and computes
$M^2_{group} = PRNG(ID_{group} \oplus r_{T_b} \oplus PRNG(K_{group}) \oplus PRNG(M^1_{group}))$
$M_{T_b} = PRNG(ID_{T_b} \oplus r'_{T_b} \oplus PRNG(K_{T_b}) \oplus PRNG(M_{T_a}))$
$\{r_{T_b}, r'_{T_b}, M^2_{group}, M_{T_b}\}$
Stores $r'_{Tb}$
$\{r_{T_b}, M^2_{group}, M_{T_b}\}$
Computes
$M^{2*}_{group} = PRNG(ID_{group} \oplus r_{T_b} \oplus PRNG(K_{group}) \oplus PRNG(M^1_{group}))$
If $M^{2*}_{group} = M^2_{group}$,it computes
$M_{T_{ab}} = PRNG(ID_{T_a} \oplus M_{T_a} \oplus PRNG(M_{T_b}) \oplus PRNG(K_{T_a} + 1))$
$M_{T_{ab}}$
Generates
$e^{T_{AB}}_n = \{ID_{T_a}, ID_{T_b}, t_n, r'_{T_a}, r'_{T_b}, M_{T_{ab}}\}$

**Fig. 1.** Kazahaya Grouping Proof Protocol.

4. $T_b$ receives the message and verifies whether $M^1_{group} \overset{?}{=} PRNG(ID_{group} \oplus r_{T_a} \oplus PRNG(K_{group}) \oplus PRNG(t_n))$. If yes, it does as follows:
   - generates two random numbers $r_{T_b}$ and $r'_{T_b}$,
   - computes $M^2_{group}$ and $M_{T_b}$ as below:
     $M^2_{group} = PRNG(ID_{group} \oplus r_{T_b} \oplus PRNG(K_{group}) \oplus PRNG(M^1_{group}))$
     $M_{T_b} = PRNG(ID_{T_b} \oplus r'_{T_b} \oplus PRNG(K_{T_b}) \oplus PRNG(M_{T_a}))$
   - and sends $\{r_{T_b}, r'_{T_b}, M^2_{group}, M_{T_b}\}$ to the reader.
5. The reader stores $r'_{T_b}$ and sends $\{r_{T_b}, M^2_{group}, M_{T_b}\}$ to $T_a$.
6. $T_a$ receives the message and verifies whether $M^2_{group} \overset{?}{=} PRNG(ID_{group} \oplus r_{T_b} \oplus PRNG(K_{group}) \oplus PRNG(M^1_{group}))$. If yes, it computes $M_{T_{ab}}$ as below and sends it to the reader:
   $M_{T_{ab}} = PRNG(ID_{T_a} \oplus M_{T_a} \oplus PRNG(M_{T_b}) \oplus PRNG(K_{T_a} + 1))$.
7. On reception the message, the reader generates the evidence $e^{T_{ab}}_n = \{ID_{T_a}, ID_{T_b}, t_n, r'_{T_a}, r'_{T_b}, M_{T_{ab}}\}$.

The designers of Kazahaya claim that their protocol provides optimal security against all attacks in the context. However, in this letter, we show that it suffers from an efficient secret disclosure attack which makes feasible other known attacks such as traceability attack and impersonation attack.

## 3 Secret Disclosure Attack on Kazahaya

In this section we present a passive and efficient attack against Kazahaya. Our attack is based on the observation that, given $PRNG(x)$ and the fact that $x \in \{0,1\}^{16}$, an attacker can determine $x$ by $2^{16}$ evaluations of $PRNG$-function in off-line mode. Following the above observation, to determine the secret parameters of the tag in Kazahaya, the adversary can do as below:

1. Eavesdrop one successful run of protocol and store protocol's messages include $t_n, r_{T_a}, r'_{T_a}, M^1_{group}, M_{T_a}, r_{T_b}, r'_{T_b}$,
   $M^2_{group}, M_{T_b}$ and $M_{T_{ab}}$.
2. for $i = 0$ to $2^{16} - 1$:
   - $N_i = PRNG(i)$,
   - If $N_i = M_{T_a}$ then $i = ID_{T_a} \oplus r'_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_n + 1)$. Recall that $r'_{T_a}$ and $t_n$ and so $PRNG(t_n + 1)$ are known to any one even the adversary, it is possible to determine $ID_{T_a} \oplus PRNG(K_{T_a})$ as $x = ID_{T_a} \oplus PRNG(K_{T_a}) = i \oplus r'_{T_a} \oplus PRNG(t_n + 1)$.
3. for $j = 0$ to $2^{16} - 1$:
   - $N_j = PRNG(j)$,
   - If $N_j = M_{T_{ab}}$ then $j = PRNG(ID_{T_a} \oplus M_{T_a} \oplus PRNG(M_{T_b}) \oplus PRNG(K_{T_a} + 1))$. Recall that $M_{T_a}$, $M_{T_b}$ and hence $PRNG(M_{T_b})$ are public, it is possible to determine $ID_{T_a} \oplus PRNG(K_{T_a} + 1)$ as $y = ID_{T_a} \oplus PRNG(K_{T_a} + 1) = j \oplus M_{T_a} \oplus PRNG(M_{T_b})$.
4. Compute $w = x \oplus y$=$PRNG(K_{T_a}) \oplus PRNG(K_{T_a} + 1)$.

5. For $z = 0$ to $2^{16} - 1$:
   - Verify whether $w \overset{?}{=} PRNG(z) \oplus PRNG(z+1)$, if yes return $z$ as $K_{T_a}$ and $x \oplus PRNG(z)$ as $ID_{T_a}$.
6. for $i = 0$ to $2^{16} - 1$:
   - $N_i = PRNG(i)$,
   - If $N_i = M^1_{group}$ then $i = ID_{group} \oplus r_{T_a} \oplus PRNG(K_{group}) \oplus PRNG(t_n)$. Recall that $r_{T_a}$ and $t_n$ and so $PRNG(t_n)$ are known to any one even the adversary, it is possible to determine $ID_{group} \oplus PRNG(K_{group})$ as $ID_{group} \oplus PRNG(K_{group}) = i \oplus r_{T_a} \oplus PRNG(t_n)$.
   - Return $ID_{group} \oplus PRNG(K_{group})$.

Therefore following the above attack, a passive adversary can retrieve $K_{T_a}$, $ID_{T_a}$ and $ID_{group} \oplus PRNG(K_{group})$, given the transmitted messages on one session of protocol. Those parameters would be enough to impersonate the tag, forge a yoking-proof, trace the tag and etc. The total complexity of the attack is $O(2^{16})$ evaluations of $PRNG$-function in off-line mode (The complexity of attack can be reduced by storing all possible values of $PRNG$-function in a table and search over it for any input). It must be noted that following the same approach it is possible to retrieve $ID_{T_b} \oplus PRNG(K_{T_b})$ which is enough to impersonate $T_b$ in a yoking-proof with another tag $T'_a$ or trace $T_b$.

Given all secret values of the tag, it would be easy to apply the following attacks on the protocol with a success probability of 1, and the cost of one execution of the protocol:

1. Traceability attack,
2. Tag impersonation attack,
3. Reader impersonation attack,

Although the above attack ruins all the security properties objectives of Kazahaya, we continue presenting other attacks based on different strategies.

## 4 Forging a Proof in Kazahaya

Grouping-proof schemes should allow multiple RFID tags to be scanned at once such that their co-existence is guaranteed. On the other hand, given a proof for $T_a$ and $T_b$ at the time $t_n$, it should be infeasible to use that proof to generate a new proof includes other tags or at the other time. In the other word, the integrity of the proof should be guaranteed; otherwise, the tag holder (patient) or the reader (the nurse) can deny the proof. However, we show that in Kazahaya, given a proof at the time $t_n$, it is possible to generate a proof for $t_j \neq t_n$. This proof is known as *forged proof*. Given the proof $e_n^{T_{ab}} = \{ID_{T_a}, ID_{T_b}, t_n, r'_{T_a}, r'_{T_b}, M_{T_{ab}}\}$, which is connected to the time $t_n$, where
$M_{T_{ab}} = PRNG(ID_{T_a} \oplus M_{T_a} \oplus PRNG(M_{T_b}) \oplus PRNG(K_{T_a} + 1))$
$M_{T_a} = PRNG(ID_{T_a} \oplus r'_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_n + 1))$
$M_{T_b} = PRNG(ID_{T_b} \oplus r'_{T_b} \oplus PRNG(K_{T_b}) \oplus PRNG(M_{T_a}))$

To forge an evidence for the desired time $t_j$ the adversary does as follows:

- sets $r''_{T_a} = r'_{T_a} \oplus PRNG(t_n + 1) \oplus PRNG(t_j + 1)$
- outputs $e_j^{T_{ab}} = \{ID_{T_a}, ID_{T_b}, t_j, r''_{T_a}, r'_{T_b}, M_{T_{ab}}\}$

Now we verify that the forged proof passes the verification process, as follows:

- $M'_{T_a} = PRNG(ID_{T_a} \oplus r''_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_j + 1)) = PRNG(ID_{T_a} \oplus r'_{T_a} \oplus PRNG(t_n+1) \oplus PRNG(t_j+1) \oplus PRNG(K_{T_a}) \oplus PRNG(t_j+1)) = PRNG(ID_{T_a} \oplus r'_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_n + 1)) = M_{T_a}$
- $M'_{T_b} = PRNG(ID_{T_b} \oplus r'_{T_b} \oplus PRNG(K_{T_b}) \oplus PRNG(M'_{T_a})) = PRNG(ID_{T_b} \oplus r'_{T_b} \oplus PRNG(K_{T_b}) \oplus PRNG(M_{T_a}) = M_{T_b})$

- $M'_{T_{ab}} = PRNG(ID_{T_a} \oplus M'_{T_a} \oplus PRNG(M'_{T_b}) \oplus PRNG(K_{T_a}+1)) = PRNG(ID_{T_a} \oplus M_{T_a} \oplus PRNG(M_{T_b}) \oplus PRNG(K_{T_a} + 1))$

Hence, the forged proof passes the verification and is accepted as a valid proof. The success probability of the given attack is 1 while the complexity is just eavesdropping one session of protocol. Based on this attack, the dishonest nurse can adopt a proof which she gave at time $t_n$ to any desired time $t_j$; versa a dishonest patient can deny the receiving the appropriate medication service at right time, while she has received.

## 5 Conclusions

In this letter, we considered the security of a yoking-proof protocol which has been recently proposed by Peris-Lopez *et al.* We scrutinize its security showing how a passive adversary can retrieve secret parameters of patient's tag in cost of $O(2^{16})$ off-line $PRNG$ evaluations. Given the tag's secret parameters, any security claims are ruined.

The proposed attacking technique is in light of two vulnerabilities of the protocol: (1) the short length of the used $PRNG$, which is urged by the target technology, EPC C1 G2 [6]; (2) the message-generating mechanism utilizing $PRNG$ was not carefully scrutinized. Nevertheless, to show other weaknesses of the protocol which work for any $PRNG$ length, we presented a forgery attack for which a proof generated at time $t_n$ can be used to forge a valid proof for any desired time $t_j$. The success probability of this attack is '1' and the complexity is negligible.

## References

1. G. Benelli and A. Pozzebon. "NFcare - Possible Applications of NFC Technology in Sanitary". Proceedings of the Second International Conference on Health, pages: 58-65, 2009.
2. M. Burmester, B. de Medeiros and R. Motta. Provably secure grouping-proofs for RFID Tags. In *Proceeding of the 8th smartcard research and advanced applications - CARDIS 2008*,pages: 176-190, 2008.
3. H-Y. Chien and S-B. Liu. Tree-based RFID yoking proof. In *NSWCTC'09*, pages: 550-553, 2009.
4. H.-Y. Chien, C.-C. Yang, T.-C. Wu, and C.-F. Lee. Two RFID-based solutions to enhance inpatient medication safety. In *J. Med. Syst., VOL. 35, NO. 3, pages:369-375, 2011. doi:10.1007/s10916-009-9373-7.*
5. H.-H. Huang and C.-Y. Ku. A RFID grouping proof protocol for medication safety of inpatient. In *Journal of Medical Systems*, volume 33, pages: 467-474, 2009.

6. E. Inc. Class-1 Generation-2 UHF RFID protocol for communications at 860 MHz'C960 MHz (version 1.1.0). In *Retrieved from http://www.epcglobalinc.org/standards/ uhfc1g2/*$uhfc1g2\_1\_1\_0-standard-20071017.pdf$..
7. A. Juels. Yoking-proofs. In *International Workshop on Pervasive Computing and Communication Security-PerSec*, page 138-143, 2005.
8. C-C. Lin, Y-C. Lai, JD. Tygar, C-K. Yang and C-L. Chiang. Coexistence proof using chain of timestamps for multiple RFID tags. In *DBMAN2007*, LNCS, vol. 4537 ,pages:634-43.,2007.
9. M. Pasquet, J. Reynaud and C. Rosenberger. "Secure payment with NFC mobile phone in the SmartTouch project", Collaborative Technologies and Systems, 2008. CTS 2008. , pages:121-126, 2008.
10. P. Peris-Lopez, A. Orfila, J. C. Hernandez-Castro, and J. C. Lubbe. Flaws on RFID grouping-proofs. Guidelines for future sound protocols. In *Journal of Network and Computer Applications*, volume 34, pages: 833-845, 2011.
11. S. Piramuthu. On existence proofs for multiple RFID tags. In *Proceedings of the ACS/IEEE International Conference on Pervasive Services*, pages: 317-320, 2006.
12. J. Saito and K. Sakurai. Grouping proof for RFID tags. In *In Proc. of the 19th International Conference on Advanced Information Networking and Applications*, pages: 621-624, 2005.
13. T. van Deursen and S. Radomirovic. Attacks on RFID protocols. In *Cryptology ePrint Archive*, Report2008/310, 2008.
14. R. Want. "An introduction to RFID technology," Pervasive Computing, IEEE , vol. 5, no. 1, pages: 25-33, 2006.