Wireless Pers Commun (2018) 100:1355–1375
https://doi.org/10.1007/s11277-018-5643-3

Are RNGs Achilles' Heel of RFID Security and Privacy Protocols?

Atakan Arslan[1,2] · Süleyman Kardaş[3] · Sultan Aldırmaz Çolak[1] · Sarp Ertürk[1]

Published online: 13 April 2018

Abstract Security and privacy concerns have been growing with the increased utilisation of RFID technology in our daily lives. To mitigate these issues, numerous privacy-friendly authentication protocols have been published in the last decade. Random number generators (RNGs) are necessarily used in RFID tags to provide security and privacy. However, low-end RNGs can be the weakest point in a protocol scheme and using them might undesirably cause severe security and privacy problems. On the other hand, having a secure RNG with large entropy might be a trade-off between security and cost for low-cost RFID tags. Furthermore, RNGs used in low-cost RFID tags might not work properly in time. Therefore, we claim that the vulnerability of using an RNG deeply influences the security and privacy level of the RFID system. To the best of our knowledge, this concern has not been considered in the RFID literature. Motivated by this need, in this study, we first revisit Vaudenay's privacy model which combines the early models and presents a new mature privacy model with different adversary classes. Then, we extend the model by

✉ Atakan Arslan
atakan.arslan@kocaeli.edu.tr

Süleyman Kardaş
skardas@gmail.com

Sultan Aldırmaz Çolak
sultan.aldirmaz@kocaeli.edu.tr

Sarp Ertürk
sertur@kocaeli.edu.tr

[1] Department of Electronics and Communication Engineering, Kocaeli University, İzmit, Kocaeli, Turkey

[2] TÜBİTAK BİLGEM, 41470 Gebze, Kocaeli, Turkey

[3] Faculty of Engineering and Architecture, Batman University, Batman, Turkey

introducing RANDOMEYE privacy, which allows analyzing the security of RNGs in RFID protocols. We further apply our extended model to two existing RFID schemes.

Keywords RFID · Protocol · Privacy · Security · RNG

1 Introduction

Radio Frequency IDentification (RFID) has become one of the most emerging wireless technologies used in order to identify and authenticate objects, animals and people in recent years. RFID is also one of the most likely technologies to promote the Internet of Things (IoT) paradigm and is proliferated in many real-life applications such as access control, supply chain, hospital care system, automatic toll collection, payment systems, e-passport, vicinity/proximity cards, etc. It is considered that near-field communication (NFC) technology in smart phones is a new up-to-the-minute opportunity for RFID technology and we are on the doorstep of a new RFID era [1, 2].

A simple RFID system consists of a tag (*transponder*), a reader (*interrogator*) and a back-end server. A tag basically has a microchip which stores data and an antenna used to transmit and receive messages through electromagnetic waves. Generally, it is considered that a back-end server is separated from an RFID reader and it acts as a mediator between the tags and the server for the communication. A back-end server keeps all the information (secret keys, data, etc.) about tags. Furthermore, RFID tags can be categorized into active, passive and semi-passive tags. Passive tags do not have their own power source and energize their integrated circuit (IC) by using the waves transmitted by the reader. Moreover, tags can also divided into four groups with respect to their operating frequency that usually depends on the availability of frequency bands and regulations: Low frequency (LF, 125–134.2 kHz and 140–148.5 kHz), high frequency (HF, ISM band at 13.56 MHz), ultra high frequency (UHF, 860–960 MHz) and microwave ($>$ 2.45 GHz) [2]. Passive low-cost RFID tags of smaller sizes are highly preferred in many applications and this desire introduces some computation, energy and size restrictions on the tag. The production price of the tags is usually around \$0.05–\$0.10 and the cost pressure is quite dominant on hardware capabilities [3].

Security and privacy concerns arise since a tag communicates with a reader over an insecure wireless channel. Tag impersonating, tracking (forward and backward tracing), eavesdropping, replay, man-in-the-middle and denial of service (DoS) attacks can be performed by an attacker using the messages transmitted in the air [4]. Implementing heavy cryptographic algorithms to overcome these issues is a challenging task due to the limited capabilities of low-cost RFID tags [2, 3, 5–7]. For protocol designers, such constraints enforce a trade-off between security and practicality. Furthermore, over the past few years, numerous lightweight authentication protocols have been proposed so as to mitigate security and privacy concerns for RFID systems [8]. Most of new protocols claimed that they were impregnable against every type of attack, providing different RFID system features such as scalable identification, tag ownership transfer, mutual authentication, robustness against noisy environments, reader corruption resiliency, etc. Unfortunately, many of them failed to satisfy the claimed security and privacy properties [8–11].

On the other hand, privacy models have been presented to systematically analyze the security and privacy of proposed authentication protocols. Such an evaluation is theoretically accomplished based on the privacy models to examine the security, anonymity and

untraceability properties before using an RFID protocol in real-life systems. Recently, several models have been proposed to formalize security and privacy in the context of RFID systems [12–20]. A privacy model should be detailed, attentive and flexible not to overlook the realities of practical RFID systems. Although it has been considered that Vaudenay's model [14] is one of the most evolved and well-defined privacy models, some papers have been published to ameliorate his model [16, 19–21]. These results, to the best of our knowledge, have claimed that their improvements fulfill the missing parts of the model but the privacy model has still fractures. In our opinion, the design of a new, appropriate, complete, and flexible security and privacy model considering the various abilities of an adversary is an essential need. Most importantly, we have noticed that Vaudenay's model has not taken the misuse of random number generators into consideration and this is a new and different adversary ability especially for real-word scenarios introduced in this paper.

Designers generally build the security and privacy of their protocols on the utilization of a random number generator (RNG) which is one of the most common primitive cryptographic functions. Eventhough designers regard RNGs as secure, their improper deployment might cause serious weaknesses in a protocol scheme. More importantly, many proposed RNGs that are asserted secure today, might be broken or become weaker in the near future. In the literature, presented RNG attacks [22–25] show that protocol designers should put care into the deployment of RNGs in order not to encounter security and privacy issues in their protocols.

2 Overview

2.1 Related Work

Privacy models are proposed as a base for analyzing the security and privacy of authentication protocols in a methodological manner. For this purpose, the privacy models formally define some properties such as RFID schemes, security and privacy prerequisites of the schemes and abilities of an adversary. In this context, Avoine et al. [26] has published a framework to formalize privacy in RFID protocols in 2005. Avoine also extended the previous model in his thesis [12]. Then, Juels and Weis [13] modified Avoine's model by adding a side channel information attribute. Furthermore, different model definitions were provided in [27, 28]. Although, there were several other attempts to design useful, proper and complete privacy model to represent and analyze RFID systems, the models did not consider all, or miss some important adversary properties (corruption, using side channel information, etc.) and they did not appropriately model an RFID scheme in terms of authentication, identification, protocol execution, etc. However, Vaudenay [14] has proposed a well-designed and relatively complete privacy model that has been quite popular among many protocol designers. In time, several researchers have improved Vaudenay's model [16, 19–21] for which more detail is provided below.

Avoine et al. [16] introduced the notion of time and formalized it by modifying Vaudenay's model with a new privacy class called TIMEFUL privacy. They show that an adversary can trace an RFID tag by only following the time that a reader has taken to authenticate the tag. According to their model, an adversary can call *timer oracle* to learn the spent time for its overall computations during authentication and can distinguish the tag. They stated that if an RFID protocol is TIMEFUL-private, an adversary cannot obtain anything about the tag identity using time information.

Akgün and Çağlayan [19] defined the notion of forward untraceability by extending Vaudenay's model. In their model, they emphasized the relay of valuable information on each communication round of the protocol and they claim that Vaudenay's model does not represent real-world settings because an adversary can miss some communication rounds due to some reason such as low signal to noise ratio. They applied their revised model to analyze some existing RFID protocols and showed that the schemes are not resistant to forward untraceability and server impersonation as claimed.

Kardaş et al. [20] improved Vaudenay's model by claiming that an adversary has capability to corrupt a tag at most k times. Hence, they introduced k-strong privacy that is an extension of the privacy classes of Vaudenay's model and is positioned between strong privacy and destructive privacy.

Hermans et al. [21] modified Vaudenay's model by introducing insider privacy notion based on the insider attack that is first discussed for RFID schemes by van Deursen and Radomirović [29]. They analyzed some existing RFID protocols to show the applicability of their model. Moreover, they propose a new RFID authentication protocol that provides wide-forward-insider privacy.

2.2 Contributions of The Paper

In this paper, we show that RNGs could be the weakest point in RFID authentication protocols and misusing them can cause severe security and privacy issues. From this point of view, we first revisit and extend Vaudenay's privacy model [14] by introducing the notion of RNGs based on their improper usage. To do so, we formalize a new privacy level called RANDOMEYE privacy that is integrated into Vaudenay's model.

We also claim that Vaudenay's model is not sufficient for some real-world scenarios. For instance, consider the following case that is not covered by Vaudenay's model: An adversary obtains some random numbers in a scheme and predicts the outputs of the RNG or the RNG loses its randomness because of some reasons such as aging, environmental effects, etc. (see Sect. 3.2 for further some explanations and existing attacks about RNGs). Motivated by this need, we introduce a novel adversary class what we called *RANDOMEYE* and define a new random oracle $\mathcal{O}^{RNG}$.

We further apply our enhanced model to two existing RFID schemes and analyze their security with respect to RANDOMEYE adversary class. First, we address the scheme by Song and Mitchell [30], and then the scheme by Akgün and Çağlayan [31]. We show that these schemes are vulnerable to RNG attacks and are not RANDOMEYE private according to our extended model. Namely, the adversary can obtain the secrets of the RFID tags by benefiting from the improper usage of RNGs.

Finally, we point out that RNGs might be the bottleneck of many RFID schemes. We highlight that using RNGs to mitigate security and privacy concerns can be Achilles' heel of an RFID authentication protocol.

2.3 Structure of The Paper

In Sect. 3, we present prior information about RFID protocols, RNGs and computational capabilities. This section also gives a glance of available literature. In Sect. 4, our new extended model that is a modification of Vaudenay's model is presented. In Sect. 5, the security and privacy of some existing schemes are analyzed based on our model. Section 6 concludes the paper with a brief conclusion and highlighting future research directions.

3 Preliminaries

This section provides some background information on lightweight RFID protocols, random number generators and computational capabilities. This section also covers brief information on recent work related to the aforementioned topics.

3.1 Lightweight RFID Protocols

Unlike wireless protocols that require conventional cryptographic operations [29, 32–34] such as symmetric and public key algorithms, restricted systems (in terms of computational power, storage, bandwidth, etc.) require *lightweight* or *ultra-lightweight* authentication protocols. Low-cost RFID systems are one of the prominent real-life applications of these protocols due to the capabilities and the price range of RFID tags.

Lower cost and smaller size demands for RFID tags enforce them to be some resource limitations such as reduced number of logic gates, lower energy consumption and low computational complexity. Lightweight and ultra-lightweight protocols need to be designed by taking into account the constraints of low-cost RFID tags. Hence, low-cost tags introduce many challenges in terms of security and privacy; numerous researchers have proposed protocols in order to obviate the security and privacy concerns [8].

Extremely restricted RFID tags require ultra-lightweight protocols that only supports bitwise operations (such as XOR, AND, OR, rotation, permutation, etc.) and are compliant to EPC Class-1 Generation-2 specification. Some of the famous ultra-lightweight protocols are SASI [35], LMAP [36], M2AP [37], EMAP [38] and Gossomar [39]. On the other hand, lightweight protocols use the same bitwise operations, as well as RNGs and Cyclic Redundancy Check (CRC) but no cryptographic hash functions. Several well-known protocols are presented in [40–42]. However, the restrictions mentioned above greatly limit aptitudes of RFID tags and cause security and privacy vulnerabilities. Avoine et al. [43] have evaluated and compared well-known lightweight protocols and indicated the security and privacy weaknesses. Zeeshan has also quite recently addressed the security and privacy issues in low-cost RFID systems in his Ph.D. thesis in [2].

3.2 RNGs

There are two types of random number generator: pseudo-random number generator (PRNG) and truly random number generator (TRNG). TRNG is an algorithm that generates random numbers from a natural source of randomness. PRNG, also known as deterministic random number generator (DRNG), is an algorithm for generating random numbers with a provided initial value called a seed. The output of the PRNG is called a pseudo-random bit sequence. The output of a PRNG is much longer than the length of the seed. In addition to this, the output of a PRNG seems to be random because it has to be statistically indistinguishable from random values and it is assumed to be unpredictable when its seed is not known.

Two general conditions are required from the security perceptive for a pseudorandom random generator: (1) the output of a PRNG should be statistically indistinguishable from truly random sequences, (2) the next output of the sequence should be unpredictable to an adversary with limited computational resources. Theoretically, the next output can be predictable with a negligible probability such as 2^{-80}. In fact, the minimum security requirement is that the length of the random seed has to be sufficiently large (s-bit) to be

infeasible for the adversary to search over a 2^s sized space (s is called the security parameter). In other words, the complexity of that attack is 2^s.

It is impossible to prove that the output of an RNG is random but there are various statistical tests that measure the quality of an RNG. This is accomplished by taking sample output sequences and apply the tests. The tests are probabilistic so they determine whether the samples look like a truly random sequence or not. If the generator fails, the output is regarded to be non-random. On the other hand, if an RNG passes all the tests, it is not rejected as being non-random. The five basic tests are (1) frequency test (mono bit test), (2) serial test (two-bit test), (3) poker test, (4) runs test, (5) auto-correlation test [44]. Detailed information about tests, generators, algorithms and definitions are presented in [44]. Moreover, some institutes, research centers, government agencies or organizations have specified some criteria to control the randomness of RNGs. For instance, the German Federal Office for Information Security has established several procedures for quality assessment of RNGs [45].

The use of RNGs has become the key function in most private and secure light-weight RFID protocols for low-cost RFID tags. Low-cost RFID tags have approximately 5–10 K gates and only 0.4–4 K gates can be dedicated to security operations [46]. Furthermore, designers are also restricted with the time that is required by a tag while generating a random number because RFID readers should be able to read a bunch of tags in a certain amount of time. Many publications have been presented to design and use RNGs in low-cost RFID tags. Melia-Segui et al. [47] have presented a lightweight PRNG design for low-cost passive RFID tags, called J3Gen in 2013. J3Gen is based on a LFSR (Linear Feedback Shift Register) configured with multiple feedback polynomials that are changed during the generation of sequences from a physical source. They have demonstrated that their most efficient J3Gen design, that has a 32-bit LFSR output with 16-bit feedback polynomials, requires around 1.2 K logic gate equivalence (GE). Peinado et al. [22] analyzed J3Gen and they claimed that there are two possible cryptanalytic attacks on J3Gen. Garcia-Alfaro et al. [48] showed that Peinado et al.'s assumptions are incorrect and their attack against J3Gen is not valid. At this point, although Garcia-Alfaro et al. fend off the attack on J3Gen, the literature is still waiting for objections to J3Gen is PRNG.

Peris-Lopez et al. [46] proposed a PRNG, named LAMED, for low-cost RFID tags compliant with the EPC C1G2 standard in 2009. They claimed that LAMED successfully passes several randomness tests. LAMED requires roughly 1.6 K gates and 1.9 ms to generate a 32-bit random number.

Melia-Segui et al. [23] presented a practical attack on a weak PRNG proposed by Che et al. [49] designed for EPC Gen2 tags. Che et al. proposed a LFSR based PRNG with the combination of thermal noise signal modulation. Melia-Segui et al. obtained the feedback polynomial function of the LFSR that they could predict its generated sequences. They showed that an adversary can reach the PRNG configuration with a confidence of 42% by only eavesdropping 128 bits of PRNG data.

Garcia et al. [24] have shown that the PRNG used in the MIFARE Classic chip has vulnerabilities.

Armknecht et al. [3] have pointed out that ensuring a sufficient level of entropy for RNGs is still a difficult task. They said that different experts from industry who provided them information, all agree stated that generating more than 128 true random bits per authentication on an RFID tag in the price range of $0.05–$0.10 seems currently improbable.

The EPC C1G2 (Class-1 Gen-2) RFID standard was proposed and adopted by EPC-global in 2004. In 2006, it was published as an amendment to the ISO 18000-6 standard for low-cost lightweight UHF RFID tags. The new version of standards has been recently ratified in 2013 with some optional cryptographic properties [40, 50]. According to the

recent standard, a tag generates 16-bit pseudo-random numbers (RN16) using the RNG. The RNG shall meet three randomness criteria: probability of a single RN16, probability of simultaneously identical sequences and probability of predicting an RN16. Although these requirements may be more stringent, a brute-force attack can be applied to reveal the random numbers because lightweight low-cost RFID tags are able to use $32 - bit$ output of PRNG which is a weakness. If an adversary eavesdrops the messages between the reader and the RFID tag, then a brute-force attack or a time-memory trade-off attack can be used to reveal the secrets of a victim tag.

RNGs are implemented by electronic circuits and their randomness quality can be affected by various factors such as seed entropy, aging, environmental effects (such as temperature, humidity, pressure, vibration, electromagnetic field, chemicals, etc.). As a result, biased RNGs cause irretrievable weaknesses.

Bayon et al. [25] demonstrated a practical attack ring oscillator (RO) based TRNG by injecting an EM signal and they also mention previous work about another practical assault to RO based TRNGs by injecting a sine wave signal onto the power pad of the device. Both attacks showed that it is possible to dynamically control the bias of the TRNG output.

In [44], the authors claimed that randomness and size of key generation help to eliminate the advantages of adversaries. Then, they gave an example using Data Encryption Standart (DES) encryption algorithm has 2^{56} key space size. In this case, when a secret key is selected by using a TRNG, an adversary has to try on average 2^{55} possible searches to find the correct key. On the other hand, if the encryption key was selected by using a 16-bit random secret and expanding it into with a 56-bit key by using well-known functions the adversary would need to try on average only 2^{15} possible keys to find the correct one.

In [51], the authors presented a detailed survey paper about random number generators. They compared different types of PRNGs and TRNGs. They also criticized about real randomness, theoretic and practical RNG approaches. They stated that most researchers chose the minimum-action strategy: design a TRNG, obtain at least one random number sequence that passes a chosen set of randomness tests and publishes. However, this does not mean that the corresponding TRNGs have a really good randomness quality because small variations in hardware can weaken them. Hence, a theoretical design cannot proceed towards a product without a detailed investigation of hardware and without extensive randomness proof. Furthermore, Barak et al. [52] proposed an extractor functions to make RNGs robust against aging, temperature changes, etc. Moreover, they presented a couple of weak RNGs caused by hardware imperfections.

3.3 Computational Capabilities

Hashcat is the well-known fastest password recovery cracker [53] and different versions are available for Linux, OSX, and Windows. It also comes in two variants: CPU-based (Hashcat password recovery tool) or GPU-based (oclHashcat, accelerated tool). oclHashcat is a GPU-based multi-hash cracker using a brute-force attack (implemented as a mask attack), combinator attack, dictionary attack, hybrid attack, mask attack, and rule-based attack.

The performance of oclHashcat in different operating systems (PC1[1], PC2[2], PC3[3] and PC4[4]) for MD5, SHA1, SHA256, SHA512 is depicted in Table 1 [53]. It is seen that PC3

[1] PC1: Windows 7, 32 bit Catalyst 14.9 1× AMD hd7970 1000 MHz core clock oclHashcat v1.35.

[2] PC2: Windows 7, 64 bit ForceWare 347.52 1× NVidia gtx580 stock core clock oclHashcat v1.35.

[3] PC3: Ubuntu 14.04, 64 bit ForceWare 346.29 8× NVidia Titan Xstock core clockoclHashcat v1.36.

[4] PC4: Ubuntu 14.04, 64 bit Catalyst 14.9 8× AMD R9 290X stock core clock oclHashcat v1.35.

Table 1 Performance list of oclHashcat in different operating systems

Hash type	PC1 (Mh/s)	PC2 (Mh/s)	PC3 (Mh/s)	PC4 (Mh/s)
MD5	8581	2753	135,232	92,672
SHA1	3037	655	42,408	31,552
SHA256	1122	355	16,904	12,288
SHA512	414	104	5240	4552

can do 135,232 Mh/s against MD5, which approximately accounts to 0.135 billion tries per second. Hence, if the same computer is used for exhaustive search, less than 32 ms will be required to find the result matching to the output of 32-bit PRNG.

4 The Proposed Modified Vaudenay Privacy Model

In this section, the main notation used throughout the paper (see Table 2) are provided and the proposed modified version of the well-known Vaudenay's privacy model [14] is introduced before the analysis of privacy aspects of RFID schemes. Finally, in the context of our model, the adversary abilities which includes the proposed RANDOMEYE adversary class are presented. The main notation used in this paper is shown in Table 2).

An RFID system is basically composed of three entities: a tag $\boldsymbol{T}$, a reader $\boldsymbol{R}$ and a back-end system/database $\boldsymbol{DB}$. A tag $\boldsymbol{T}$ is interrogated by a reader $\boldsymbol{R}$ and the reader identifies/authenticates $\boldsymbol{T}$ by using a unique identifier of the tag ID (in this article it is sometimes denoted as ID_T to improve the readability). $\boldsymbol{DB}$ stores all identifiers and secret keys of valid tags. $\boldsymbol{R}$ communicates with both $\boldsymbol{T}$ and $\boldsymbol{DB}$ and provides a link between them. $\boldsymbol{DB}$ might be considered as a part of the reader. Moreover, $\boldsymbol{T}$ has a restricted memory and computational capacities and can communicate with $\boldsymbol{R}$ for a limited distance. We assume that $\boldsymbol{R}$ is much more talented than the tag which is the common case [16]. An adversary $\boldsymbol{Adv}$ can corrupt a tag and use its internal secrets against the system but she cannot corrupt $\boldsymbol{R}$. We also assume that the communications between $\boldsymbol{R}$ and $\boldsymbol{DB}$ is protected by a secure channel such as Secure Sockets Layer (SSL)/Transport Layer Security (TLS).

4.1 Definitions of RFID Scheme

An RFID system is defined by the following procedures.

- SetupReader $(1^{\alpha}) \longrightarrow (K_P, K_S)$ is a setup algorithm that generates a public-private key pair (K_P, K_S) for the reader $\boldsymbol{R}$ where α is the security parameter, and then initializes an empty database $\boldsymbol{DB}$ to store all identifiers and secret keys of all tags. Although K_s is kept secretly in the $\boldsymbol{DB}$ with the security parameter α; K_p is publicly released.
- SetupTag $\left(K_p, ID\right) \longrightarrow (K, S)$ is a probabilistic algorithm which returns a tag secret K and the initial state S of a tag $\boldsymbol{T}$ with the input identifier ID. When $\boldsymbol{T}$ is legitimate, the pair (ID, K) is to be stored in the database $\boldsymbol{DB}$.
- Ident $\longrightarrow$ *Output* is an interaction protocol between a tag $\boldsymbol{T}$ and the reader $\boldsymbol{R}$ to complete the protocol transcripts. At the end of the protocol, if $\boldsymbol{T}$ is legitimate, $\boldsymbol{R}$

Table 2 Main notation used throughout the paper

Adv	Adversary
$\boldsymbol{Adv^B}$	Blinded Adversary
b_i	Legitimacy of the ith tag (e.g. The ith tag is legitimate when $b_i = 1$)
B	Blinder
DB	Database/Back-end System
distr	Probability distribution
$ID_{\boldsymbol{T}_n}$	Unique identifier of nth tag
K	Key
(K_P, K_S)	Public-private key pair for a reader ***R***
$\mathcal{O}^A$	The oracle for the functionality of A
m	Message
RNG_i	The output of RNG for ith protocol instance
r_i	ith random bit string
R	Reader
S	The whole memory of $\psi_{\boldsymbol{T}}$
s_i	The corruption state of the *RNG* of a tag ***T*** for the ith protocol instance
T	Tag
$tbl(.)$	Table function (takes pseudonym of the ith tag as an input and outputs its unique identifier, $tbl(\psi_{\boldsymbol{T}i}) = ID_i$)
θ_π	Sufficient tuples which contains the pairs of protocol instances and *RNG* states (π_i, s_i)
α	Security parameter
$\psi_{\boldsymbol{T}_n}$	Pseudonym of the nth tag
π_i	ith protocol transcript

accepts the tag (***R*** identifies ***T***) and outputs its identifier *Output=ID*, otherwise (i.e. if it is not valid) ***R*** refuses ***T*** and outputs $\perp$.

4.2 Definitions of the Oracles

An adversary ***Adv*** against an RFID scheme acts as an honest reader and/or an honest tag to attack the system. We assume that there is only one legitimate reader ***R*** in the RFID system and both valid readers and tags of the system have no prior information about the entity that is interacting with themselves. We also suppose that each experiment always starts with executing the algorithm SetupReader thus, K_p, K_s and 1^α are already generated. We consider that K_p and 1^α are already available to ***Adv*** but K_s is kept secret because ***R*** cannot be corrupted. Furthermore, we assume that there are no tags in the system at the beginning of each experiment and ***Adv*** is allowed to call $\mathcal{O}^{CreateTag}$ oracle to add new tags to the system.

According to Vaudenay's model [14], a tag is considered as either a free tag or a drawn tags. Drawn tags are the set of tags that ***Adv*** has visual contact and communicates ***Adv*** cannot interact with initially free tags. When ***Adv*** calls the $\mathcal{O}^{CreateTag}$ oracle, she generates a new tag whose status is free. The following oracles are used by the adversary ***Adv*** to interact with the RFID system. First of all, ***Adv*** setups a new tag of identifier *ID*.

- $\mathcal{O}^{CreateTag}(ID, b)$: It creates a free tag $\boldsymbol{T}$ with a unique identifier ID using SetupTag. $\boldsymbol{T}$ is legitimate when $b = 1$, otherwise $b = 0$ and $\boldsymbol{T}$ is not valid. It also inserts (ID, K) into $\boldsymbol{DB}$. b is implicitly 1 when neglected.

Then, the adversary may change the status of the tag from free to drawn by calling the following oracle.

- $\mathcal{O}^{DrawTag}(distr, n) \rightarrow (\psi_{\boldsymbol{T}_1}, b_1, \ldots, \psi_{\boldsymbol{T}_n}, b_n)$: It randomly selects n free tags among all existing ones with distribution probability of the given $distr$. The oracle assigns a new pseudonym, $\psi_{\boldsymbol{T}_i}$ for each tag and changes their status to drawn. Hence, the oracle returns an array of fresh pseudonyms $\left(\psi_{\boldsymbol{T}_1}, \psi_{\boldsymbol{T}_2}, \ldots, \ldots, \psi_{\boldsymbol{T}_n}\right)$ of the tags ($\psi_{\boldsymbol{T}_n}$ is the pseudonym of the nth tag). The pseudonyms are always changed from session to session so that the adversary may interact to drawn tags for only one single session. The relations $(\psi_{\boldsymbol{T}_i}, ID_i)$ are stored in a hidden table tbl such that $tbl(\psi_{\boldsymbol{T}i}) = ID_i$. This oracle also returns a bit array $(b_1, b_2 \ldots, \ldots, b_n)$ where b_i of the ith tag shows whether it is legitimate or not. Furthermore, the oracle may return $\perp$ if the querying tags are already drawn or there are no existing tags.

When the tag is drawn, the adversary is only able to interact to the tag with pseudonym $\psi_{\boldsymbol{T}}$. $\psi_{\boldsymbol{T}}$ is defined as a temporary identifier of a tag and used for pointing to the tag anonymously. In this case the following oracles can be called.

- $\mathcal{O}^{Free}(\psi_{\boldsymbol{T}})$: This oracle changes the state of tag $\boldsymbol{T}$ that is represented by the pseudonym $\psi_{\boldsymbol{T}}$ from drawn to free. Afterwards $\boldsymbol{Adv}$ is no longer able to interact with $\boldsymbol{T}$. The secret key of the tag with the pseudonym $\psi_{\boldsymbol{T}}$ is denoted as $key[\psi_{\boldsymbol{T}}]$. The adversary can corrupt the drawn tags by using the following oracle and obtain the internal values of the tag including its secret key.
- $\mathcal{O}^{Corrupt}(\psi_{\boldsymbol{T}}) \rightarrow S$: S is the whole memory of $\psi_{\boldsymbol{T}}$. $\boldsymbol{Adv}$ obtains the $key[\psi_{\boldsymbol{T}}]$. Eventually, the tag $\boldsymbol{T}$ with the pseudonym $\psi_{\boldsymbol{T}}$ is destroyed and $\boldsymbol{Adv}$ cannot interact to $\boldsymbol{T}$ any more.
- $\mathcal{O}^{Launch}() \rightarrow \pi$: This makes the reader $\boldsymbol{R}$ start a new Ident protocol with transcript π.
- $\mathcal{O}^{SendReader}(m, \pi) \rightarrow m'$: This sends the message m to the reader $\boldsymbol{R}$ in the protocol transcript π with outputs the response m'.
- $\mathcal{O}^{SendTag}(m, \pi) \rightarrow m'$: This sends the message m to $\boldsymbol{T}$ and outputs the response m'. Also, $\boldsymbol{Adv}$ asks for the reader's result of the protocol transcript π. The adversary can use the corresponding oracle to change the state of the tag so she can start to interact with the tag change, the state to drawn or she can free the tag (after which she communicate). anymore.
- $\mathcal{O}^{Execute}(\psi_{\boldsymbol{T}}) \rightarrow (\pi, transcript)$: This executes a complete protocol between the reader and the tag with pseudonym $\psi_{\boldsymbol{T}}$. It returns the transcript of the protocol instance that is the list of all successive messages of the protocol.
- $\mathcal{O}^{Result}(\pi) \rightarrow x$: This returns $x = 1$ when π completes successfully after the Ident returns $Output \neq \perp$ which means that the tag $\boldsymbol{T}$ is identified. Otherwise, if $\boldsymbol{T}$ is not identified and $Output = \perp$, this oracle returns $x = 0$.

Finally, we introduce a new oracle called RNG oracle, $\mathcal{O}^{RNG}$ as follows. The adversary $\boldsymbol{Adv}$ is allowed to obtain the results of the RNG bit string used in the protocol by a tag $\boldsymbol{T}$ by querying the following oracle. For simple explanation, π_i denotes the ith protocol instance, s_i is the corruption state of the RNG of a tag $\boldsymbol{T}$ for the ith protocol instance. If $s_i = 0$, $\boldsymbol{Adv}$ does not corrupt $\boldsymbol{T}$ but if $s_i = 1$, she corrupts $\boldsymbol{T}$ and captures the $key[\psi_{\boldsymbol{T}}]$ for the protocol instance π_i. The array of (π_i, s_i) values is denoted by $\theta_\pi := \{(s_1, \pi_1), (s_2, \pi_2), \ldots, (s_n, \pi_n)\}$

and θ_π defines the sufficient number of n tuples where each tuple includes the protocol transcript and tag corruption information.

- $\mathcal{O}^{RNG}(\theta_\pi, \psi_{\boldsymbol{T}}) \rightarrow (RNG_1, RNG_2 \ldots, RNG_i, \ldots, RNG_n)$: This outputs the set of the *RNG* bit string used on the tag $\boldsymbol{T}$ with the unique identifier $ID_{\boldsymbol{T}}$ for each protocol instance π_i and the state s_i. The oracle returns $\perp$ for any protocol instance π_i, when the RNG used in this instance cannot be obtained.

Adv performs her attack by running an experiment or playing a game and obeying the corresponding rules. Firstly, she constructs an RFID system and uses the oracles and gets a result. She wins or looses depending on the corresponding rules.

4.3 Definition of the Adversary Classes

We define different adversary classes for playing security games. The definition includes Vaudenay's model [14] and our own novel adversary class.

Definition 1 *(Adversary Classes).* An adversary ***Adv*** against an RFID system who has an arbitrary number of accesses to the above oracles except the $\mathcal{O}^{RNG}$ oracle is regarded to be in one of the following classes.

- STRONG ***Adv*** uses all oracles without any restrictions.
- DESTRUCTIVE ***Adv*** cannot use an oracle against a tag after using $\mathcal{O}^{Corrupt}$ oracle (i.e. the tag has been killed).
- FORWARD ***Adv*** can only use $\mathcal{O}^{Corrupt}$ oracle after her first call to this oracle.
- WEAK ***Adv*** uses all oracles except $\mathcal{O}^{Corrupt}$ oracle
- NARROW ***Adv*** has no access to $\mathcal{O}^{Result}$ oracle.
- RANDOMEYE ***Adv*** can access the RNG oracle $\mathcal{O}^{RNG}$, and extracts the random number(s) used in a tag. This is a novel class introduced in this paper.

4.4 Security Notions

Some security properties of an RFID system such as completeness and soundness are visited below.

Definition 2 *(Completeness).* An RFID system is complete if the reader $\boldsymbol{R}$ of the system returns the tag identifier *ID* at the end of the protocol (IDENT) for a legitimate tag $\boldsymbol{T}$ with very high probability.

Definition 3 *(Strong Completeness).* An RFID system is complete if the reader $\boldsymbol{R}$ of the system returns the tag identifier *ID* at the end of the protocol (IDENT) for a legitimate tag $\boldsymbol{T}$ with very high probability although the RFID scheme has been already attacked.

According to Vaudenay's model, security is a vital property and should be withheld against every attack by the strongest adversary. But it is obvious that the security of a scheme is violated when tag impersonation occurred if the adversary uses $\mathcal{O}^{Corrupt}$ oracle. Hence, the model permits an adversary to use all oracles except the $\mathcal{O}^{Corrupt}$ oracle.

Definition 4 *(Soundness).* An RFID system is said sound if an adversary ***Adv*** impersonates a legitimate tag $\boldsymbol{T}$ with a negligible probability [16].

4.5 Privacy

Vaudenay defines a privacy notion that is the deducing ability of an adversary to obtain the *ID* relations of a tag from its protocol instances. He explains *anonymity* and *untraceability* properties under the privacy notion in that one is about unveiling the *ID* of tags and the other is about indistinguishability of any two tags, respectively [14].

In the RFID literature, there are two types of untraceability notions: *forward untraceability* and *backward untraceability*. If an RFID system provides the forward untraceability feature, an adversary ***Adv*** who compromises a legitimate tag at a time t, cannot trace the future interactions of the tag, $t' > t$. If an RFID system provides the backward untraceability feature, ***Adv*** cannot trace past interactions of the tag, $t' < t$. The backward untraceability property is also referred to as *forward privacy* or *forward secrecy* and this notion is more important than forward untraceability for real life scenarios. Vaudenay also considers the privacy of the RFID system based on the adversary classes in Definition 1. In his model, he presents a blinded adversary called blinder ***B***.

Definition 5 *(Blinder, trivial adversary).* A blinder ***B*** for an adversary ***Adv*** is a polynomial-time algorithm that observes the same messages as ***Adv*** and simulates LAUNCH, SENDREADER, SENDTAG, and RESULT oracles without having access to the secret keys nor the database of the system. The adversary ***Adv*** uses all outputs of the oracles. A blinded adversary $\boldsymbol{Adv^B}$ is an adversary who never uses LAUNCH, SENDREADER, SENDTAG, and RESULT oracles. An adversary ***Adv*** is said to be *trivial* if there exists a blinded adversary $\boldsymbol{Adv^B}$ such that $|\, Prob[\boldsymbol{Adv}\ wins] - Prob[\boldsymbol{Adv^B}\ wins] \,|$ is negligible.

If the success probability of the simulator and the blind adversary is nearly the same, this means that the blind adversary has attack ability at least as high as the simulator of the system (except using the secret keys). Hence, the authentication and identification of a tag can be considered private. Vaudenay says that an adversary accomplishes his attack (plays a security game) into two phases. In the first phase, she queries the allowed oracles and collects the outputs. In the second phase, she analyses the obtained results without using any oracle. Between the two phases, she also has access to the hidden table *tbl* of the $\mathcal{O}^{DrawTag}$ oracle. If she outputs true from her analysis, then she wins the game.

Definition 6 *(Privacy).* An RFID system is P-private if all the adversaries who belong to class P are trivial following Definition 5 [14].

The following well-known links between Vaudenay's privacy classes which are rather obvious by definition.

$$\begin{array}{ccccccc} \text{STRONG} & \Longrightarrow & \text{DESTRUCTIVE} & \Longrightarrow & \text{FORWARD} & \Longrightarrow & \text{WEAK} \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow \\ \text{NARROW-STRONG} & \Longrightarrow & \text{NARROW-DESTRUCTIVE} & \Longrightarrow & \text{NARROW-FORWARD} & \Longrightarrow & \text{NARROW-WEAK} \end{array}$$

4.6 The Proposed RANDOMEYE Adversary Class

Now we are ready to explain our RANDOMEYE adversary class and its relationship to the other adversary classes. The RANDOMEYE adversary class formalizes the weakness and/or misuse of random number generators for real life RFID systems. Tangibly, an adversary ***Adv*** that can query the $\mathcal{O}^{RNG}$ oracle, might learn the random numbers used in the authentication protocol. If ***Adv*** cannot infer the *ID* of the tag by using this information, we

consider that the protocol is RANDOMEYE private. Hence Vaudenay's original adversary classes are not complete and the relationship between them has changed with the newly introduced class. Therefore, we give the new link for the STRONG class as follows for clear comprehensibility:

RANDOMEYE-STRONG	$\Longrightarrow$	STRONG
$\Downarrow$		$\Downarrow$
NARROW-RANDOMEYE-STRONG	$\Longrightarrow$	NARROW-STRONG

5 Case Study Protocols

In this section, we consider two popular existing RFID schemes to apply our new model and provide analysis. We first briefly introduce Song and Mitchell's and Akgün et al.'s schemes. Then, we explain how an adversary attacks and break the schemes step by step. Our analysis further shows that the schemes do not provide security and privacy properties with respect to the presented weakness. Hence, according to our improved model, the protocols are not RANDOMEYE private.

5.1 First Study Example: Song and Mitchell's Protocol

Firstly, we investigate the scheme designed by Song and Mitchell (SM) [30] to provide private and secure authentication between low-cost RFID tags. Their protocol is depicted below.

In this protocol the reader generates a nonce r_1 and sends it to the tag to start the protocol. The tag receives the nonce and generates a random bit string, r_2 as a temporary secret for the protocol instance. The tag computes $M_1 = r_1 \oplus tid_i$ and $M_2 = f_{tid_i}(r_1 \oplus r_2)$. Then, the tag sends M_1 and M_2 to the reader. The reader evaluates and searches its database by using M_1, M_2 and r_1. If the reader does not find any match, it will stop the session. In case of a successful match, the reader authenticates the tag and updates the tag information which is $(u_i)_{old}$ and $(tid_i)_{old}$. Then it computes $M_3 = u_i \oplus (r_2 \gg l/2)$ and sends M_3 message to the tag. The tag computes u_i using M_3 and checks that $h(u_i) = t_i$. If a match is obtained, the tag authenticates the reader and updates its u_i and t_i values. Otherwise, the tag does not update the current values. This process is shown in Fig. 1.

We prove below that a RANDOMEYE adversary can trace a tag in this protocol without corrupting it.

Theorem 1 *The SM protocol does not ensure the RANDOMEYE-WEAK privacy.*

Proof An adversary ***Adv*** can perform the following attack.

1. ***Adv*** creates two legitimate tags by using $\mathcal{O}^{CreateTag}(tid_1, 1)$ and $\mathcal{O}^{CreateTag}(tid_2, 1)$ oracles. Then, ***Adv*** draws two tags from the system by calling $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$oracle and obtains two pseudonyms $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$. At this point, ***Adv*** does not know tid_1 and tid_2 that are the identifiers of the $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ tags respectively.
2. ***Adv*** calls $\mathcal{O}^{Execute}(\boldsymbol{T}_1)$ and gets $\theta_\pi = (0, \pi_1)$ for $\boldsymbol{T}_1$.

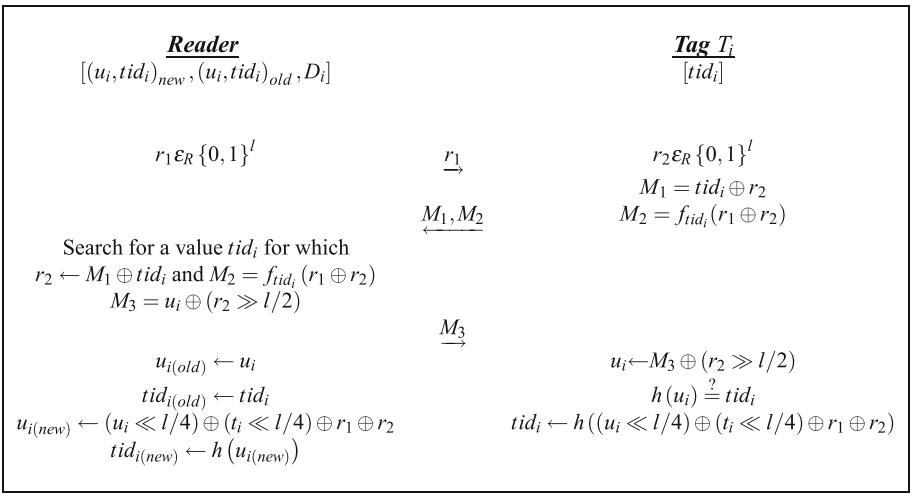

Fig. 1 Song and Mitchell's Protocol

3. Then, ***Adv*** **requests** $\mathcal{O}^{RNG}[\theta_\pi, \boldsymbol{T}_1]$ and obtains $(RNG_1, 1)$ for $\boldsymbol{T}_1$. For this protocol RNG_1 is equal to the random bit strings r_2 generated by the tag, $\boldsymbol{T}_1$. $\mathcal{O}^{RNG}$ oracle performs the following procedures:
   (a) It generates all possible random strings for r_2 with respect to the seed of the RNG used in the tag. Lets call the list $\boldsymbol{R} = \left[r_2^1, r_2^2, \ldots, r_2^j, \ldots r_2^{|K|}\right]$ where $|K|$ is the entropy of the seed.
   (b) It has the list of all the possible $\boldsymbol{X} = \left[tid_1^1, tid_1^2, \ldots, tid_1^j, \ldots tid_1^{|K|}\right]$ values by computing $\boldsymbol{X} = M_1 \oplus \boldsymbol{R}$ because M_1 is obtained within the protocol instance.
   (c) Then, it does the exhaustive search to check for the M_2 messages with computing $f_{\boldsymbol{X}}(r_1 \oplus \boldsymbol{R})$. Finding $M_2 = f_{M_1 \oplus r_2^j}(r_1 \oplus r_2^j)$, ***Adv*** obtains r_2 that is equal to r_2^j.
4. ***Adv*** obtains the tid_1 for tag $\boldsymbol{T}_1$ computing $M_1 \oplus r_2$ and updates the internal values of the tag according to the protocol procedure. Therefore, ***Adv*** has the $tid_{1(new)}$ value of $\boldsymbol{T}_1$.
5. ***Adv*** performs step 2, step 3 and step 4 for the $\boldsymbol{T}_2$ tag. ***Adv*** updates the internal values of the tag and gets the $tid_{2(new)}$ value of $\boldsymbol{T}_2$.
6. ***Adv*** frees both tags with request $\mathcal{O}^{Free}(\boldsymbol{T}_1)$ and $\mathcal{O}^{Free}(\boldsymbol{T}_2)$, then she reaffects only one of them using $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$. She obtains a new $\boldsymbol{T}_3$.
7. ***Adv*** performs step 2, step 3 and step 4 for the $\boldsymbol{T}_3$ tag and obtains tid_3.
8. Then ***Adv*** compares tid_3 with $tid_{1(new)}$ and $tid_{2(new)}$.
9. If $tid_3 = tid_{1(new)}$, ***Adv*** claims that $T_3 = T_1$ else she claims that $T_3 = T_2$.

The success probability of this adversary is equal to 1. Therefore, it is clear that Song and Mitchell's Protocol is not RANDOMEYE-WEAK private. □

5.2 Second Study Example: Akgün et al.'s Scheme

Akgün and Çağlayan [31] introduced a new authentication protocol and claimed that it is the first protocol that provides destructive privacy according to Vaudenay's model with constant identification time. This scheme is a simple challenge/response protocol enhanced with Physically Unclonable Functions (PUFs) in order to achieve higher level of privacy.

This scheme has two phases. In the first phase, the system initializes itself. In this initialization phase, a shared secret S is randomly generated for the back-end server. Two random values, a and b are generated for each tag. Then each tag performs its own PUF $P(.)$ to calculate $c = S \oplus P(a) \oplus P(b)$. The back-end server stores all values $[ID_i, a_i, b_i, DATA_i]$ for each tag where $DATA_i$ contains the information about a tag T_i.

In the second phase called authentication phase, the reader generates a random number r_1 and broadcasts it to the tag.

Secondly, a tag T_i which receives the signal of the reader, generates another random number r_2. The tag also computes $M_1 = H(r_1, r_2, a_i)$, $M_1 = H(r_2, r_1, 1) \oplus ID_i$ and $h = H(r_2, 1, 2)$. Then, it uses PUF to calculate $k = P_i(a_i) \oplus r_2$ and deletes the r_2 and $P_i(a_i)$ values from the volatile memory. The tag updates k by computing $k = k \oplus P_i(b_i) \oplus c_i$ and then $P_i(b_i)$ is deleted from the memory too. The tag transmits M_1, M_2 and k back to the reader.

Thirdly, the reader generates a new random number r_3 and computes $r_2' = S \oplus k$, $ID_i' = M_2 \oplus H(r_2', r_1, 1)$. Then, the reader checks that the M_1 message is equal to $H(r_1, r_2', a_i)$ to authenticate the tag T_i. If the equality is confirmed, then the reader computes $M_3 = H(H(r_2', 1, 2), r_3, b_i)$ and sends r_3 and M_3 to the tag T_i.

Finally, the tag T_i checks that the M_3 message is equal to $H(h, r_3, b_i)$ to authenticate the reader. If the equality is confirmed, the tag authenticates the reader too. Thus, mutual authentication is accomplished and the protocol is terminated successfully. This is shown in Fig. 2.

Akgün et al. claimed that their protocol scheme provides destructive privacy according to Vaudenay's privacy and security model with constant time identification property. Their protocol does not need key-updating mechanism on both, tags and back-end server. The authors use the common secret S to identify a tag with $O(1)$ time complexity. They base the security and privacy of their protocol on the PUFs that are regarded to have robustness, unclonability, unpredictability and tamper-evident properties [31]. We realized that there is a RNG misuse in their protocol design. We can prove that their protocol is neither destructive private nor secure. A RANDOMEYE adversary can trace the past and future transactions of the tag as proven below.

Theorem 2 *Akgün et al.'s protocol does not ensure the RANDOMEYE-WEAK privacy.*

Proof An adversary ***Adv*** can perform the following attack.

1. ***Adv*** creates two legitimate tags by using $\mathcal{O}^{CreateTag}(ID_1, 1)$ and $\mathcal{O}^{CreateTag}(ID_2, 1)$ oracles. Then, ***Adv*** draws two tags from the system by calling $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$oracle and obtains two pseudonyms $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$. At this point, ***Adv*** does not know ID_1 and ID_2 that are the identifiers of the $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ tags respectively.
2. ***Adv*** calls $\mathcal{O}^{Execute}(\boldsymbol{T}_1)$ two times and gets $\theta_\pi = \{(0, \pi_1), (0, \pi_2)\}$ for $\boldsymbol{T}_1$.
3. Then, ***Adv*** **requests** $\mathcal{O}^{RNG}[\theta_\pi, \boldsymbol{T}_1]$. ***Adv*** obtains (RNG_1) and (RNG_2) respectively for $\boldsymbol{T}_1$. For this protocol scheme, RNG_1 is equal to the random bit strings r_2 generated by

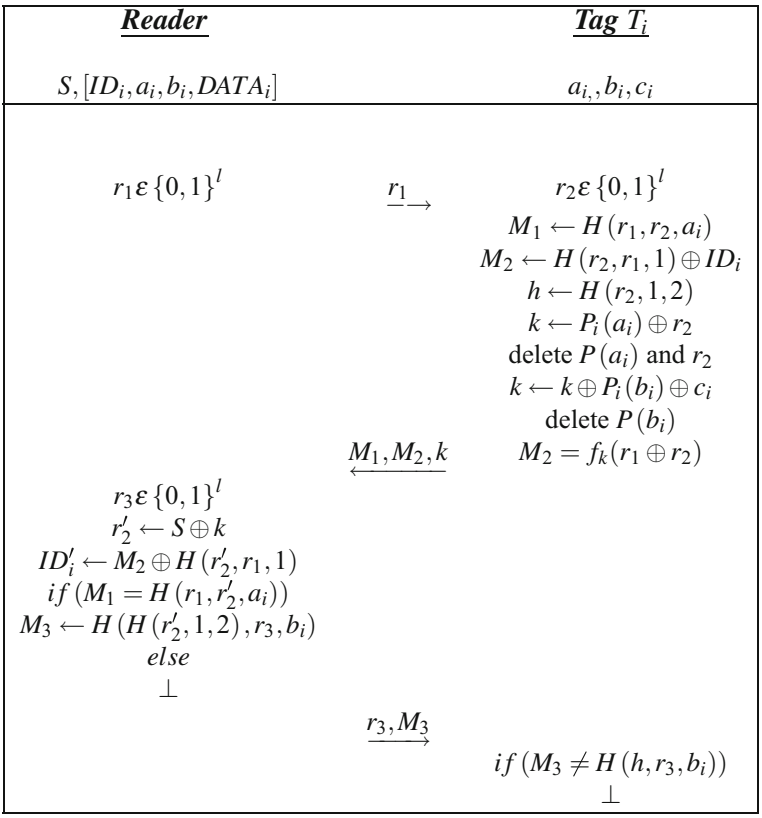

Fig. 2 Akgün et al.'s Authentication Protocol

the tag, $\boldsymbol{T}_1$ for the first protocol instance and RNG_2 is the secondly generated random bit string r_2. $\mathcal{O}^{RNG}$ oracle performs the following procedures:

(a) It generates all possible random strings for r_2 with respect to the seed of the RNG used in the tag. Lets call the list $\boldsymbol{R} = \left[r_2^1, r_2^2, \ldots, r_2^j, \ldots r_2^{|K|}\right]$ where $|K|$ is the entropy of the seed.

(b) It has the list of all the possible $\boldsymbol{X^1} = \left[ID_1^1, ID_1^2, \ldots, ID_1^j, \ldots, ID_1^{|K|}\right]$ values by computing $\boldsymbol{X^1} = M_2 \oplus H(\boldsymbol{R}, r_1, 1)$ because M_2 and r_1 are obtained within the first protocol instance.

(c) It has the second list of all the possible $\boldsymbol{X^2} = \left[ID_1^1, ID_1^2, \ldots, ID_1^j, \ldots, ID_1^{|K|}\right]$ values by computing $\boldsymbol{X^2} = M_2 \oplus H(\boldsymbol{R}, r_1, 1)$ because M_2 and r_1 are obtained within the second protocol instance.

(d) Then, it compares $\boldsymbol{X^1}$ and $\boldsymbol{X^2}$ and defines the identifier of the tag by finding the equal bit string of each list.

(e) Finally, it obtains the random bit string r_2 by using the corresponding identifier of the tag ID_1 .

4. ***Adv*** obtains ID_1 for $\boldsymbol{T}_1$ tag by computing $M_2 \oplus r_2$ using one of the protocol instances.
5. ***Adv*** performs step 2, step 3 and step 4 for the $\boldsymbol{T}_2$ tag. ***Adv*** obtains ID_2 for $\boldsymbol{T}_2$.
6. ***Adv*** frees both tags with request $\mathcal{O}^{Free}(\boldsymbol{T}_1)$ and $\mathcal{O}^{Free}(\boldsymbol{T}_2)$, then she re-affects only one of them using $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$. She obtains a new $\boldsymbol{T}_3$.
7. ***Adv*** performs step 2, step 3 and step 4 for the $\boldsymbol{T}_3$ tag and obtains ID_3.
8. Then ***Adv*** compares ID_3 with ID_1 and ID_2.
9. If $ID_3 = ID_1$, ***Adv*** claims that $\boldsymbol{T}_3 = \boldsymbol{T}_1$ else she claims that $\boldsymbol{T}_3 = \boldsymbol{T}_2$.

Therefore, if the adversary ***Adv*** captures the *IDs*, she can trace the past and future transactions of the tags of the scheme using the unchanging *ID*. Hence, the scheme does not provide forward and backward untraceability properties. □

Theorem 3 *Akgün et al.'s protocol does not ensure the RANDOMEYE-DESTRUCTIVE privacy.*

Proof Akgün et al.'s protocol does not provide WEAK privacy. Hence, it is not DESTRUCTIVE private. □

Theorem 4 *Akgün et al.'s scheme is not secure against RANDOMEYE adversary.*

Proof It is clearly seen that the Akgün et al.'s scheme does not provide RANDOM-WEAK privacy and a passive adversary is able to reveal the *ID* of a tag. Let an adversary ***Adv*** reveals the *ID* of a tag and consequently has the random bit strings r_2. ***Adv*** also has the k value obtained during eavesdropping to the protocol session where $k = P_i(a_i) \oplus r_2 \oplus P_i(b_i) \oplus c_i$. The shared secret S is generated as $S = P_i(a_i) \oplus P_i(b_i) \oplus c_i$ in the initialization according to the protocol description. Thus, the adversary ***Adv*** obtains the shared secret S by computing $S = k \oplus r_2$. The scheme is not longer secure after the shared secret S is obtained and the whole system can be broken by the adversary ***Adv***. □

6 Conclusion and Future Work

In this paper, we focus on the improper usages of RNGs in privacy-friendly RFID authentication protocols and show that misusing RNGs in a protocol design might cause serious security and privacy weaknesses. To prove our claim, we first have revisited and enhanced an RFID privacy and security model proposed by Vaudenay by modeling a new attack based on misusing of the RNGs. In this context, we extend the model by introducing a new RNG oracle and *RANDOMEYE* adversary class. Then, we apply our improved model on recently published lightweight RFID authentication protocols. We show that Song and Mitchell's [30] and Akgün and Çağlayan's [31] schemes are vulnerable to RNG attacks. In our point of view, RNGs should only be utilized to increase the security and privacy level of the protocols instead of becoming a brittle point of the scheme. It is known that a chain is only as strong as its weakest link and we point out that misusing RNGs might be the weakest link in a protocol design. Moreover, for future analysis, a completely new RFID privacy model can be constructed.

References

1. Want, R., Schilit, B. N., & Jenson, S. (2015). Enabling the internet of things. *IEEE Computer*, *48*(1), 28–35.
2. Bilal , Z. (2015). *Addressing security and privacy issues in low-cost RFID systems*. Ph.D. thesis, Royal Holloway, University of London, London, UK.
3. Armknecht, F., Hamann, M., & Mikhalev, V. (2014). Lightweight authentication protocols on ultra-constrained RFIDs—myths and facts. In N. Saxena & A. R. Sadeghi (Eds.), *Radio frequency identification: Security and privacy issues* (pp. 1–18). Cham: Springer
4. Ghaeini, H.R., & Tippenhauer, N.O. (2016). HAMIDS: Hierarchical monitoring intrusion detection system for industrial control systems. In *Proceedings of the 2nd ACM workshop on cyber-physical systems security and privacy, CPS-SPC '16* (pp. 103–111). New York, NY, USA.
5. Juels, A. (2004). Minimalist cryptography for low-cost RFID tags. In C. Blundo & S. Cimato (Eds.), *International conference on security in communication networks—SCN 2004, volume 3352 of of lecture notes in computer science* (pp. 149–164). Amalfi, Italy, Springer.
6. Avoine, G., Bingöl, M. A., Carpent, X., & Kardaş, S. (2013). *Deploying OSK on low-resource mobile devices* (pp. 3–18). Berlin: Springer.
7. Kardas, S., Celik, S., Bingöl, M.A., & Albert, L. (2013). A new security and privacy framework for RFID in cloud computing. In *IEEE 5th international conference on cloud computing technology and science, CloudCom 2013, Bristol, United Kingdom* (Vol. 1, pp. 171–176)
8. Avoine, G. (2017). *RFID lounge*. http://www.avoine.net/rfid/. Accessed March 2.
9. Bilal, Z., Martin, K., & Saeed, Q. (2014). Multiple attacks on authentication protocols for low-cost RFID tags. *Applied Mathematics and Information Sciences*, *9*(2), 561–569.
10. Radványi, T., Biró, C., Király, S., Szigetváry, P., & Takács, P. (2015). Survey of attacking and defending in the RFID system. *Annales Mathematicae et Informaticae*, *44*, 151–164.
11. Alavi, S. M., Baghery, K., & Abdolmaleki, B. (2014). Security and privacy flaws in a recent authentication protocol for EPC C1 G2 RFID tags. *Advances in Computer Science: An International Journal*, *3*(5), 44–52.
12. Avoine, G. (2005). *Cryptography in radio frequency identification and fair exchange protocols*. Ph.D. thesis, EPFL, Lausanne, Switzerland.
13. Juels, A., & Weis, S. (2007). Defining strong privacy for RFID. In *International conference on pervasive computing and communications—PerCom* (pp. 342–347). New York City, New York, USA, IEEE, IEEE Computer Society.
14. Vaudenay, S. (2007). On privacy models for RFID. In K. Kurosawa (Ed.), *Advances in cryptology ASIACRYPT 2007, volume 4833 of of lecture notes in computer science* (pp. 68–87). Berlin: Springer.
15. Avoine, G. (2005). *Adversary model for radio frequency identification*. Technical report, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC).
16. Avoine, G., Coisel, I., & Martin, T. (2010). Time measurement threatens privacy-friendly RFID authentication protocols. In S.B. Ors Yalcin (Ed.), *Workshop on RFID security—RFIDSec'10, volume 6370 of lecture notes in computer science* (pp. 138–157) Istanbul, Turkey, Springer.
17. Ha, J., Moon, S., Zhou, J., & Ha, J. (2008). A new formal proof model for RFID location privacy. *Proceding of the 13th European symposium on research in computer security–ESORICS 2008, volume 6123 of lecture notes in computer science* (pp. 267–281). Malaga, Spain, Springer.
18. Lai, J., Deng, R.H., & Li, Y. (2010). Revisiting unpredictability-based RFID privacy models. In *Proceedings of the 8th international conference on applied cryptography and network security—ACNS 2010, volume 6123 of lecture notes in computer science* (pp. 475–492). Beijing, China, Springer.
19. Akgün, M., & Çağlayan, M. (2011). Extending An RFID security and privacy model by considering forward untraceability. In J. Cuellar, J. Lopez, G. Barthe & A. Pretschner (Eds.), *Security and trust management* (pp. 239–254). Berlin: Springer.
20. Kardaş, S., Çelik, S., Bingöl, M. A., Kiraz, M. S., Demirci, H., & Levi, A. (2014). *k*-strong privacy for radio frequency identification authentication protocols based on physically unclonable functions. *Wireless Communications and Mobile Computing*, *15*, 1–17. https://doi.org/10.1002/wcm.2482.
21. Hermans, J., Peeters, R., & Preneel, B. (2014). Proper RFID privacy: Model and protocols. *IEEE Transactions on Mobile Computing*, *13*(12), 2888–2902.
22. Peinado, A., Munilla, J., & Fúster-Sabater, A. (2013). EPCGen2 pseudorandom number generators: analysis of J3Gen. *IACR Cryptology ePrint Archive*, *2013*, 825.
23. Melia-Segu, J., Garcia-Alfaro, J., & Herrera-Joancomart, J. (2011). A practical implementation attack on weak pseudorandom number generator designs for EPC Gen2 tags. *Wireless Personal Communications*, *59*(1), 27–42.

24. Garcia, F. D., de Koning Gans, G., Muijrers, R., van Rossum, P., Verdult, R., Schreur, R. W., et al. (2008). Dismantling MIFARE classic. In S. Jajodia & J. Lopez (Eds.), *Computer security—ESORICS 2008, volume 5283 of lecture notes in computer science* (pp. 97–114). Berlin: Springer.
25. Bayon, P., Bossuet, L., Aubert, A., Fischer, V., Poucheret, F., Robisson, B., et al. (2012). Contactless electromagnetic active attack on ring oscillator based true random number generator. In W. Schindler & S. Huss (Eds.), *Constructive side-channel analysis and secure design, volume 7275 of lecture notes in computer science* (pp. 151–166). Berlin: Springer.
26. Avoine, G., Dysli, E., & Oechslin, P. (2005). Reducing time complexity in RFID systems. In B. Preneel & S. Tavares (Eds.), *Selected areas in cryptography–SAC 2005, volume 3897 of lecture notes in computer science* (pp. 291–306). Kingston, Canada, Springer.
27. Lim, C. H., & Kwon, T. (2006). Strong and robust RFID authentication enabling perfect ownership transfer. In P. Ning, S. Qing, & N. Li (Eds.), *International conference on information and communications security—ICICS'06, volume 4307 of lecture notes in computer science* (pp. 1–20). Raleigh, North Carolina, USA, Springer.
28. Van Le, T., Burmester, M., & de Medeiros, B. (2007). Universally composable and forward-secure RFID authentication and authenticated key exchange. In F. Bao & S. Miller (Eds.), *ACM symposium on information, computer and communications security—ASIACCS 2007* (pp. 242–252). Singapore, Republic of Singapore, ACM, ACM Press.
29. van Deursen, T., & Radomirović, S. (2012). Insider attacks and privacy of RFID protocols. In *Proceedings of the 8th European conference on public key infrastructures, services, and applications* (pp. 91–105). Springer.
30. Song, B., & Mitchell, J.C. (2008). RFID authentication protocol for low-cost tags. In V.D. Gligor, J.-P. Hubaux, & R. Poovendran (Eds.), *Proceedings of the 1st ACM conference on wireless network security—WiSec'08* (pp. 140–147). Alexandria, Virginia, USA, ACM, ACM Press.
31. Akgün, M., & Çağlayan, M. (2015). Providing destructive privacy and scalability in RFID systems using PUFs. *Ad Hoc Networks*, *32*, 32–42.
32. Lauter, K. (2004). The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless Communications*, *11*(1), 62–67.
33. Yih-Chun, H., & Perrig, A. (2004). A survey of secure wireless ad hoc routing. *IEEE Security Privacy*, *2*(3), 28–39.
34. Altop, D. K., Bingöl, M. A., Levi, A., & Savaş, E. (2017). DKEM: Secure and efficient distributed key establishment protocol for wireless mesh networks. *Ad Hoc Networks*, *54*(C), 53–68.
35. Chien, H.-Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing*, *4*(4), 337–340.
36. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Workshop on RFID security—RFIDSec'06* (pp. 12–14). Graz, Austria, Ecrypt.
37. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2006). M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In J. Ma, H. Jin, L. T. Yang, & J. J. P. Tsai (Eds.), *International conference on ubiquitous intelligence and computing—UIC'06, volume 4159 of lecture notes in computer science* (pp. 912–923). China, Wuhan and Three Gorges, Springer.
38. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M. & Ribagorda, A. (2006). Emap: An efficient mutual-authentication protocol for low-cost rfid tags. In *OTM confederated international conferences" On the move to meaningful internet systems"* (Vol. 4277, pp. 352–361). Springer.
39. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2008). Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In K.-I. Chung, K. Sohn, & M. Yung (Eds.), *Workshop on information security applications—WISA'08, volume 5379 of lecture notes in computer science* (pp. 56–68). Jeju Island, Korea, Springer.
40. EPC Global, (2014). *UHF air interface protocol standard Generation2/Version2*. http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2. Accessed March 2, 2017.
41. Peris-Lopez, P., Lim, T. L., & Li, T. (2008). Providing stronger authentication at a low-cost to RFID tags operating under the EPCglobal framework. In C.-Z. Xu & M. Guo (Eds.), *Embedded and ubiquitous computing—Volume 02—EUC'08* (pp. 159–166). Shanghaim, China, IEEE, IEEE Computer Society.
42. Chien, H.-Y., & Chen, C.-H. (2007). Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. *Computer Standars & Interfaces*, *29*(2), 254–259.
43. Avoine, G., Bingöl, M. A., Carpent, X., & Yalcin, S. B. O. (2012). Privacy-friendly authentication in RFID systems: On sub-linear protocols based on symmetric-key cryptography. *IEEE Transactions on Mobile Computing*, *12*(10), 2037–2049. https://doi.org/10.1109/TMC.2012.174.

44. Menezes, A. J., Vanstone, S. A., & Van Oorschot, P. C. (1996). *Handbook of applied cryptography* (1st edn.). Boca Raton: CRC Press, Inc.
45. Schindler, W., & Killmann, W. (2003). Evaluation criteria for true (physical) random number generators used in cryptographic applications. In *Revised papers from the 4th international workshop on cryptographic hardware and embedded systems, CHES '02* (pp. 431–449). London, UK, Springer.
46. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2009). LAMED—A PRNG for EPC Class-1 Generation-2 RFID specification. *Computer Standards and Interfaces*, *31*(1), 88–97.
47. Melia-Segu, J., Garcia-Alfaro, J., & Herrera-Joancomart, J. (2013). J3Gen: A PRNG for low-cost passive RFID. *Sensors*, *13*(3), 3816–3830.
48. Garcia-Alfaro, J., Herrera-Joancomart, J., & Segu, J. M. (2015). Remarks on Peinado et al.'s analysis of J3Gen. *Sensors*, *15*(3), 6217–6220.
49. Che, W., Deng, H., Tan, W., & Wang, J. (2008). A random number generator for application in RFID tags. In P. H. Cole & D. C. Ranasinghe (Eds.), *Networked RFID systems and lightweight cryptography* (pp. 279–287). Berlin: Springer.
50. ISO/IEC Standard 18000 RFID Air Interface Standard. (2014). http://www.hightechaid.com/standards/18000.htm. Accessed March 2, 2017.
51. Sarma, S., Weis, S., & Engels, D. (2002). RFID systems and security and privacy implications. In B. Kaliski, Ç. Kaya ço, & C. Paar (Eds.), *Cryptographic hardware and embedded systems—CHES 2002, volume 2523 of lecture notes in computer science* (pp. 454–469). Redwood Shores, California, USA, Springer.
52. Barak, B., Shaltiel, R., & Tromer, E. (2003). *True random number generators secure in a changing environment* (pp. 166–180). Berlin: Springer.
53. hashcat. (2015). *Performance*. http://hashcat.net/oclhashcat/. Accessed August 30, 2015.

Atakan Arslan received the B.S. degree in Telecommunication Engineering from Istanbul Technical University, Istanbul, Turkey in 2008 and 2012. He is currently a candidate of Ph.D. degree in Electronics and Telecommunication Engineering, Kocaeli University, Kocaeli, Turkey. His research interests include information security, privacy, RFID systems and cryptographic protocols.

Süleyman Kardaş received the M.S. degree in Computer Engineering from Bilkent University, Ankara, Turkey, in 2009 and the Ph.D. degree in Computer Engineering from Sabanci University, Istanbul, in 2014. He is currently an Assistant Professor with the Computer Engineering Department, Batman University, Batman, Turkey. His research interests include cryptography, information security, secure multi party computation, RFID systems and e-voting

Sultan Aldırmaz Çolak received the B.Sc. degree in Electronics and Telecommunication Engineering from Kocaeli University, Kocaeli, Turkey, in 2004 and the M.S. degree and the Ph.D. degree in Electronics and Telecommunication Engineering from Yildiz Technical University, istanbul, Turkey, in 2006 and in 2012, respectively. She is currently an Assistant Professor with the Department of Electronics and Telecommunication Engineering, Kocaeli University. Her research interests include multi user communication systems, massive MIMO, VLC, OFDM, modulation.

Sarp Ertürk (M'99) received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 1995 and the M.Sc. degree in telecommunication and information systems and the Ph.D. degree in electronic systems engineering from the University of Essex, Colchester, U.K., in 1996 and 1999, respectively. From 1999 to 2001, he carried out his compulsory service at the Army Academy, Ankara. He is currently a Full Professor with the Department of Electronics and Telecommunication Engineering, Kocaeli University, where he was an Assistant Professor between 2001 and 2002 and an Associate Professor between 2002 and 2007. His research interests include digital signal and image processing, video coding, remote sensing, and digital communications.