Journal of Computer and System Sciences 81 (2015) 1027–1041

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Vulnerability aware graphs for RFID protocol security benchmarking

Shan Chang [a], Li Lu [b,*], Xiaoqiang Liu [a], Hui Song [a], Qingsong Yao [c]

[a] School of Computer Science and Technology, Donghua University, 2999 North Renmin Road, Shanghai, 201620, China
[b] School of Computer Science & Engineering, University of Electronic Science and Technology of China, No. 4, Section 2, North Jianshe Road, Chengdu, 610054, China
[c] School of Computer Science and Technology, Xidian University, 2 South Taibai Road, Xi'an, 710071, China

ARTICLE INFO

Article history:
Received 21 April 2014
Received in revised form 22 September 2014
Accepted 2 October 2014
Available online 30 December 2014

Keywords:
RFID
Security protocol
Vulnerability aware graphs
Benchmarking

ABSTRACT

Security and privacy issues in Radio Frequency Identification (RFID) systems mainly result from limited storage and computation resources of RFID tags and unpredictable communication environment. Although many security protocols for RFID system have been proposed, most of them have various flaws. We propose a random graph-based methodology enabling automated benchmarking of RFID security. First, we formalize the capability of adversaries by a set of atomic actions. Second, Vulnerability Aware Graphs (VAGs) were developed to elaborate the interactions between adversaries and RFID systems, which are used to discover the potential attacks of adversaries via some paths on the graphs. The quantitative analysis on VAGs can predict the probability that the adversary leverages the potential flaws to perform attacks. Moreover, a joint entropy-based method is provided to measure the *indistinguishability* of RFID tags under passive attacks. Analysis and simulation were conducted to show the validity and effectiveness of VAGs.

1. Introduction

Radio frequency identification (RFID) is an emerging technology that has posed great benefit to many applications such as logistics [1], manufacture [2], retailing [3], transportation [4], anti-counterfeiting [5], etc. When attached to objects or persons, RFID tags can be used for identification and verification. It is important to prevent RFID systems from leaking confidential information of their users, and provide security guarantee for each participant in the system [6]. To this end, security protocols have been extensively studied to provide secure guarantee for data delivery between readers and tags, thwarting attackers.

RFID tags usually have tough constraints on the storage and computing resources (due to the cost consideration), which implies that the security protocols applied to RFID systems are expected to be lightweight, and many conventional encryption schemes used for wired systems, for example asymmetric key based ciphers, are not suitable for RFID systems. Therefore, RFID oriented security protocols tend to have much more flaws than those used for wired environments or other wireless applications. Worse yet, the common environments wherein RFID tags are deployed are open and unpredictable. An adversary can launch sophisticated attacks to penetrate the protocols, raising serious threats to RFID systems.

* Corresponding author.
E-mail address: changshan@dhu.edu.cn (S. Chang).

http://dx.doi.org/10.1016/j.jcss.2014.12.015
0022-0000/

For modeling the security of RFID systems, some researchers have proposed either formal definitions of the privacy and security [7–10], or formalization of adversaries [11]. However, designing and analyzing RFID security protocols are still challenging since existing models are high-leveled and coarse-grained. With previous models, it is difficult to detect exact flaws in RFID protocols or attacks based on those flaws. Moreover, the security of RFID protocols as well as the impact of attacks cannot be quantitatively analyzed. In fact, the security analysis of RFID protocols is still on a pedestrian level. Thus, developing automated analysis and benchmarking measures for security protocols are extremely important to enhance the security of RFID systems. In this paper, we propose a random graph-based method coordinated with a fine-grained adversary model enabling automated analysis and benchmarking of RFID security protocols. Our contribution mainly comes from five aspects.

First, we propose an adversary model with a more flexible, fine-grained, and realistic structure. We formalize the capability of an adversary by a set of atomic actions. Any attack launched by an adversary can be accurately represented via some of the atomic actions. Unlike previous models, atomic actions defined by our model are more accurate in simulating adversaries in real world.

Second, for the purpose of expressing how an adversary interacts with a tag or a reader separately, we do not generate a single state transition graph to show interactions between tag and reader. Instead, we propose a novel random graph-based method, termed VAGs (Vulnerability Aware Graphs), which uses a pair of graphs, Tag Graph (TG) and Reader Graph (RG), to reflect the interactions between the adversary and tag, the adversary and reader, respectively. VAGs enable the detection of the potential flaws in security protocols and the analysis of the adversary's attack patterns. By utilizing VAGs, an attack can be mapped to a harmful path on either TG or RG. We develop a set of rules to define the harmful paths. Each path refers to a flaw of the protocol. Each edge on a harmful path denotes an action taken by the adversary. Thus, by finding the harmful paths, we can derive the attack patterns.

Third, an adversary can hardly utilize a flaw which is extremely imperceptible. It is reasonable to calculate the probability that the adversary can detect and make use of the certain flaw, i.e., the probability he or she can walk along the path representing the flaw. We further analyze the probability an adversary can exploit a flow and launch an attack quantitatively.

Forth, since the passive adversaries, which do not disturb the communication between tags and readers, can only pose a threat to the *indistinguishability* of tags, the VAGs of a passive adversary are reduced to a pair of graphs indicating the interactions between a pair of reader and tag. We propose another joint entropy-based method to measure the *indistinguishability* of tags under passive attacks.

Finally, we achieve the whole benchmarking methodology for RFID security protocols.

The rest of the paper is organized as follows. In Section 2, we discuss the related work. We introduce the preliminaries in Section 3 and present the attack model in Section 4. We formalize the problem in Section 5 and detail our VAGs methodology in Section 6. Section 7 discusses the measuring of *indistinguishability* of tags under passive attacks. Section 8 shows the case study. We conclude our work in Section 9.

2. Related work

In this section, we will briefly introduce the previous researches closely related to our work, including RFID security models, graph-based methods for network security analysis, RFID benchmarking and security evaluation of RFID protocols. Table 1 shows the differences of existing work and the proposed one from several aspects, including proposing security models, conducting security benchmarking, automatic protocol analyzing, quantitative analyzing and type of attacks defending.

2.1. RFID security models

Some researchers are working on modeling the security or privacy of RFID systems [7–11]. A. Juels et al. [9] define the strong privacy of RFID system used for basic analysis of RFID systems. Whether a protocol is strong privacy can be analyzed manually. However, how to automatically analyze protocols with the definition is not mentioned. X. Zhang et al. [7] describe some security requirements, including privacy of tag data, privacy of ownership, integrity of tag data, and availability of tag identity in RFID system. It then formalizes the definitions of these requirements. I. Damgard et al. [10] model the behaviors of adversaries by specifying the actions she can take, but the adversary model is coarse-grained and inflexible for characterizing realistic adversaries. G. Avoine [11] also introduces an adversary model suitable for RFID environments. The formalization of an adversary can only be used to analyze the protocols in terms of traceability. S. Vaudenay [8] proposes a model and definition for anonymous identification of RFID system based on [9]. It considers the situation that tags within an RFID system hold dependent keys, and discusses the security–efficiency tradeoff.

2.2. Graph-based methods for network security analysis

Both scenario graph [12] and attack graph [13] (which is a specific kind of scenario graph) are developed to evaluate the security of networked systems. There are a lot of work both on analyzing network vulnerabilities by using attack graph and generating attack graph [13–16]. Attack graph can show the relationship of vulnerabilities on different hosts in network. It takes into account the interrelationship of vulnerabilities activated by the interactions between hosts, since

Table 1
The differences between existing work and the proposed VAGs.

Achieves	Security models	Security benchmarking	Automatically protocol analysis	Quantitative analysis	Attacks (active or passive)
[7–11]	yes	no	no	no	both
[17,18]	no	no	no	yes	none
[27]	no	yes	no	yes	passive
[28,29]	no	yes	no	no	both
[30]	yes	yes	no	no	both
VAGs	yes	yes	yes	yes	both

several vulnerabilities on different hosts can be utilized together to achieve certain attack objects. However original attack graph cannot sufficiently express the relationships that may be utilized by adversaries in RFID systems. In RFID systems, the adversary can not only utilize the relationship between the secrets of tags in the system (actually, in many RFID system, tags are independent for the consideration of strong privacy [8], and our work is focused on such RFID systems), More important is the adversary can utilize the relationship between several executions of the protocol on the same tag which cannot be described by an attack graph.

2.3. RFID benchmarking

There are also some existing archives on RFID benchmarking [17,18], but they mainly focus on the performance and reliability of MAC or physical layer of tags. They aim to evaluate the impacts of surrounding objects, readers' power, the distance between tags and readers, or the physical orientation of tags on the performance of tags, such as read rate.

2.4. Security evaluation of RFID protocols

M. Alizadeh et al. analyzed the security (in terms of confusion and diffusion) of several lightweight encryption algorithms used in RFID applications. The authors only concern passive attacks on confidentiality of tags [27]. Some researchers performed the security and privacy analyses on lightweight protocols proposed recently and discuss their advantages and security issues [28,29]. Changshe Ma et al. proved that *ind-privacy* is weaker than *unp-privacy*. The necessary and sufficient condition on RFID tags to achieve *unp-privacy* is determined [30]. A pseudo-random function family is the minimal requirement on an RFID tag's computational power for enforcing strong RFID system privacy.

3. Preliminaries

3.1. RFID system

RFID tag can be divided into two types: *passive* and *active* tag. The main difference between them is that the passive tag does not have internal power, while the active tag does [19]. The passive tag makes use of RF waves emitted from the reader as the power supply. It thereby cannot afford complex microprocessors that require relatively high power to work. In order to reduce the cost of production, the storage capacity of passive tag is also extremely limited (current passive tag commonly has couples of kilobytes as the storage). Due to the above hardware constraints, security risks on passive tags are extremely high, and security protocols of passive tags tend to have more flaws than that of active tags. In this paper, we focus on the security issues on passive tags. Note that our method can be compatibly applied to the protocols of active tags.

In RFID systems, the reader is a device used to interrogate RFID tags. For passive tags, the reader is also responsible for activating the tag's microchip. The reader does not store sensitive information about the tags or systems. After obtaining responses of tags, it simply transmits the result to backend server for further processing. Generally, the channel between a reader and backend server is considered safe. Tags and readers communicate via two open unsecure wireless channels, i.e., reader-to-tag (forward channel) and tag-to-reader (backward channel) communications. Communication distance of forward channel is larger than that of backward channel, since the reader's transmission power is much higher than that of tags. Correspondingly, the eavesdropping distance of adversaries on the forward channel is also much larger than that on the backward channel. The back-end server processes the information related to tags in the system and is powerful in terms of computational resource. Hence, it is considered as a secure and trusted entity, which cannot be compromised by attackers. In this paper, we consider the 'reader' as a single unit combining the RFID reader and backend server.

The communication between the passive tag and reader adopts the challenge–response mechanism. The reader initiates a challenge (interrogation). After receiving the challenge, the tag computes a result, i.e. response, and sends it back to the reader. The security protocol executed between the reader and tags is also based on the challenge–response mechanism. A successful execution of protocol called a *session*. In this paper, we denote m a message delivered between a tag and a reader, s_i is the ith session of a RFID security protocol, and tag_j is the jth tag in the system. The automatic challenge–response communication mechanism used by RFID tags and readers leaves a security flaw, with which the attacks can easily gain unauthorized access to RFID data. Therefore, RFID security and privacy are increasingly important.

3.2. Security issues of RFID system

A tag indicates the type of an object to which it is attached, or even uniquely identifies the object. Therefore, the data a tag carries on is highly related to the confidential information of the object's owner. Concerns of an adversary $\mathcal{A}$ aiming at breaking down an RFID system include *privacy*, *security*, and *functionality* of it. We formally define the concerns as follows.

Definition 1 *(Privacy).* The privacy problem of an RFID system comes from two aspects: one is data leakage of RFID-tagged belongings, and the second is behavioral tracking or personal identification by tracing tag IDs.

In privacy-guaranteed RFID system, anonymity and untraceability should be considered. Anonymity indicates that a tag's responses should be unlinkable to the ID of the tag. Untraceability requires that $\mathcal{A}$ cannot distinguish a particular response of targeted tag from those of other tags (otherwise $\mathcal{A}$ can trace the tag). Indeed, anonymity and untraceability can be satisfied by *indistinguishability* [20], hence we focus on the *indistinguishability* in the paper.

In order to ensure *indistinguishability*, it is necessary that transmitted tag's information should not be the same or correlate with each other from the perspective of inappropriate parties. The two most significant threats on *indistinguishability* are tracking and hotlisting.

Definition 2 *(Security).* Security is focused on the tag impersonation problem in RFID systems, meaning that $\mathcal{A}$ is not able to cheat a reader, i.e., to make the reader to accept responses from a faked *tag*. (This definition is consistent with the definition of security in [10].)

Security refers to the certainty that the transmitted tag's information is not tampered with during or after challenge–response processes. It implies that the data will not be modified or destroyed by unauthorized parties. The *security* of the RFID challenge–response processes can be compromised through relay, replay, message reconstruction, data modification/insertion attacks.

Definition 3 *(Functionality).* Functionality denotes that all the functions provided by the RFID systems can be correctly conducted. Towards an RFID system that has a functionality guarantee, the adversary cannot disturb its functions, for example, making a legitimate reader to reject a legitimate tag, or vice versa.

Functionality requires that the information is available when it is needed. In order for an RFID system to demonstrate *functionality*, it must have properly functioning computing systems, security controls and communication channels. Denial of Service (DoS) attacks are one of the most challenging threats against *functionality* since they can be easily deployed while they are hard to defend against. This type of attacks includes passive degrading the RF signal, and active jamming or disrupting communications. De-synchronous attack is another challenging threat which causes reader reject legitimate tags.

Notice that the above issues we concerns are similar to the classical CIA (Confidentiality, Integrity, and Availability) requirements, where *privacy* to confidentiality, *security* to integrity, and *functionality* to availability.

4. Attack modeling

In this section, we discuss the behaviors of adversaries, and construct several atomic actions on which any attack can be built. We then model how an adversary can intervene in challenge–response processes between a tag and a reader.

An adversary $\mathcal{A}$ is an entity that attempts to thwart the *security* and *privacy*, or disable the system *functionality*. We assume that the adversary knows the security scheme used by the RFID system, i.e., the specification of security protocol. $\mathcal{A}$ does not know the secret shared between the tag and reader (backend server), unless he or she compromises the tag. $\mathcal{A}$ can compromise a tag via some kinds of physical attacks, but these physical attacks are destructive, which means the tag become useless after the compromising. Due to the fact that the wireless communication channels between the tag and reader are open, $\mathcal{A}$ is able to acquire partial or full control over the messages transmitted on both forward and backward channels, which is dependent on the capabilities of $\mathcal{A}$.

$\mathcal{A}$ can launch various attacks, such as eavesdropping which is the interception of communication between a legitimate reader and tag, rogue scanning which involves a malicious reader that could access a tag without its owner's permit, tracking, hijack, physical attacks, spoofing (cloning, swapping, relay, replay etc.), and DoS attacks.

Although the purposes and patterns of those attacks are diverse, the behavior of each attack can be resolved into a finite set of atomic actions. In other words, any attack comprises of a successive series of atomic actions. We consider those actions as oracles (each oracle can be considered as a theoretical black box, which is able to solve certain decision problems) that can be accessed by $\mathcal{A}$.

Definition 4 *(The atomic actions).* A set of oracles that an adversary $\mathcal{A}$ can access:

- **SendToReader** (m, s_i): this oracle formalizes the model of $\mathcal{A}$ sending message m to the reader in session s_i of the protocol through backward channel.

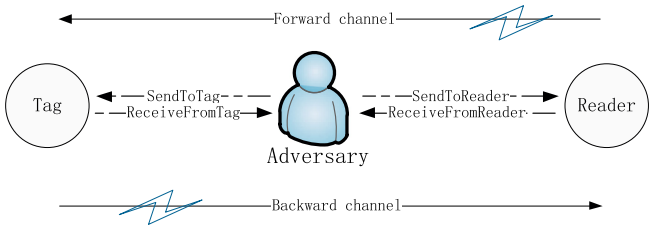

Fig. 1. The adversary serves as a part of RFID system.

- **SendToTag** (m, tag_j, s_i): this oracle formalizes the model of $\mathcal{A}$ sending message m to tag_j in session s_i of the protocol through forward channel.
- **ReceiveFromTag** (m, tag_j, s_i): this oracle formalizes the model of $\mathcal{A}$ receiving message m from tag_j in session s_i of the protocol through backward channel.
- **ReceiveFromReader** (m, s_i): this oracle formalizes the model of $\mathcal{A}$ receiving message m from the reader in session s_i of the protocol through forward channel.
- **Corrupt** (tag_j): this oracle formalizes the model of $\mathcal{A}$ compromising tag_j. $\mathcal{A}$ can obtain the current secret of tag_j, and make the tag disabled, and hence all the atomic actions corresponding to tag_j cannot be used any more.
- **SideChannel** (tag_j, s_i): this oracle formalizes the model of $\mathcal{A}$ observing the result of tag_j in session s_i of the protocol.

The capability of $\mathcal{A}$ is a subset of the oracles defined in Definition 4. Now we elaborate the capabilities of $\mathcal{A}$ and how $\mathcal{A}$ interferes with RFID systems in our model. We consider $\mathcal{A}$ serving as a relaying forwarder similar to a router in a conventional IP network.

- By accessing the **SendTo*** and **ReceiveFrom*** oracles in Definition 4, $\mathcal{A}$ can control all the communications between the tag and reader (as shown in Fig. 1) and decide whether a message is relayed faithfully, distorted, or dropped (i.e. intercepted).
- $\mathcal{A}$ accesses **SendToReader** and **SendToTag** oracles to send a message that complies with the specification of the protocol to the reader and tag, respectively. **ReceiveFromReader** and **ReceiveFromTag** represent that $\mathcal{A}$ receives the messages transmitted on the two channels. Note **ReceiveFrom*** implies that not only can $\mathcal{A}$ eavesdrop on the messages transmitted on channels, but also $\mathcal{A}$ fully controls those messages. For example, intercepting a message means that $\mathcal{A}$ accesses **ReceiveFrom** while ignoring **SendTo***.
- By accessing **Corrupt**, $\mathcal{A}$ can obtain the current secret stored in the tag, and compute the corresponding information using compromised secret and historical observations.
- **SideChannel** allows $\mathcal{A}$ to obtain the result of protocol execution of a given session, i.e., accepting or rejecting a tag.

Particularly, the oracles above can model not only active but also passive adversaries, who only listen on communication between tags and readers, but do not modify the message stream in any way. For a passive adversary, the actions of he or she should follow the following principles:

- The adversary cannot access the **Corrupt** oracle.
- In session s_i, **ReveiveFromReader** (m, s_i) and **SendToTag** (m, tag_j, s_i) should always access in pairs. Furthermore, **ReveiveFromReader** (m, s_i) always follows with **SendToTag** (m, tag_j, s_i) (similarly, **ReveiveFromTag** (m, tag_j, s_i) always follows with **SendToReader** (m, s_i)), which means that the adversary faithfully relay message m.

Above oracle designs pose several advantages to our model, such as the flexibility, fine-grained structure, and accurate definitions for real adversary's capabilities. For example, different from existing adversary models, our model resolves the interaction between $\mathcal{A}$ and reader (or tag) into oracles **ReceiveFromReader** (or **ReceiveFromTag**) and **SendToReader** (or **SendToTag**) instead of treating the entire interaction as a single oracle. Similarly, we should also split the **ReceiveFrom*** oracle used in previous approaches into two more fine-grained oracles, **ReceiveFromReader** and **ReceiveFromTag**. Hence, an adversary who can access oracle **ReceiveFromReader** may not be able to access oracle **ReceiveFromTag**, which is more close to reality. It is because in real RFID systems, the eavesdropping range is relatively larger than the scanning range. The eavesdropping range of tag-to-reader and eavesdropping range of reader-to-tag are also different (the former one is smaller than the latter one). On the other hand, the reader and tag will not always response after receiving a message, which is dependent on the certain protocol. To reflect those incomplete responses, we also need a more flexible definition for the actions of $\mathcal{A}$.

5. Problem formulation

In this section, we first describe the state transition model that represents the interactions between the reader and $\mathcal{A}$, the tag and $\mathcal{A}$, respectively. The couple of state transition model describe how $\mathcal{A}$ intervenes in the communications between the tag and reader. It also describes how state transitions of the tag or reader proceed under the actions launched by $\mathcal{A}$. Then we introduce the formal definition of VAGs, which is a random graph-based methodology used for automated analysis and benchmarking of RFID security protocols.

5.1. Formalizations of state transition model

We model both the state transitions of the tag and reader forced by $\mathcal{A}$'s actions using an uniform transition system $M = (S, I, R, E)$ where:

a) $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n\}$: a set of states, s.t. $\boldsymbol{S} \neq \emptyset$.
b) $\boldsymbol{I}$: the initial state
c) $\boldsymbol{R} \subseteq \boldsymbol{S} \times \boldsymbol{S}$: a set of transitions
d) $\boldsymbol{E} \subseteq \boldsymbol{S}$: a set of end states.

We treat tag and reader as counterparts to each other. Particularly, $M_T = (S_T, I_T, R_T, E_T)$ is the state transition model of the tag with the state set $S_T = \{s_1^t, s_2^t, \ldots, s_n^t\}$.$M_R = \{S_R, I_R, R_R, \emptyset\}$ is the state transition model of the reader with the state set $S_R = \{s_1^r, s_2^r, \ldots, s_n^r\}$. Then, we will focus on the details of M_T and M_R.

- **State:** a *state* of a reader (or tag) is defined by its *Internal State* and *External Output*. Specifically, the state of a tag and reader can be written as shown below:

$$s_i^r = \left(s_i^r \cdot ins, s_i^r \cdot chlg\right)$$
$$s_i^t = \left(s_i^t \cdot ins, s_i^t \cdot resp\right)$$

- **Internal State** (*Ins*): an *Internal State* includes a set of variables, which are secrets either only stored in a tag (or reader) or shared with its counterparts. The secrets can be a counter,session ID, key, etc. Note that although there are different kinds of information stored in tags or readers, for a particular protocol, only those related to state transitions should be concerned.
- **External Output**: the *External Output* of a reader is its challenge sent to a tag (*Chlg*), while that of a tag is its response to a challenge (*Resp*).
- $\boldsymbol{s_i^t \cdot resp}$ (*Resp*): *Resp* denotes a response returned by a tag at state s_i^t. If $\mathcal{A}$ accesses **SendToTag**, the tag at state s_i^t may transfer to a new state s_j^t by calculating a new *Resp* as $s_j^t \cdot resp$, while update $s_i^t \cdot ins$ to $s_j^t \cdot ins$. $\mathcal{A}$ can retrieve $s_j^t \cdot resp$ by accessing **ReceiveFromTag**.
- $\boldsymbol{s_i^r \cdot chlg}$ (*Chlg*): *Chlg* denotes a challenge generated by a reader at s_i^r. *Chlg* can be the first message launched by the reader in a session proactively. For a protocol involving several challenge–response interactions, *Chlg* can also be generated according to the message m in **SendToReader** that $\mathcal{A}$ accesses. Hence, as the reaction of **SendToReader**, the reader at s_i^r transfers to a new state s_j^r, by preparing $s_j^r \cdot chlg$, and updating $s_i^r \cdot ins$ to $s_j^r \cdot ins$. $\mathcal{A}$ takes over *Chlg* by accessing **ReceiveFromReader**.
- **The initial state** (*Ini*): *Ini* represents a special state with the original internal condition before the interaction.
- $R(s_i, s_j)$ represents a transition from state s_i to s_j. In our model, $R_T(s_i^t, s_j^t)$ exists if $\mathcal{A}$ can take either a **SendToTag** or **Corrupt** action to trigger the state transition of the tag from s_i^t to s_j^t. $R_R(s_i^r, s_j^r)$ exists if $\mathcal{A}$ can take a **SendToReader** action to trigger the state transition of the reader from s_i^r to s_j^r. Since M_T and M_R reflect the interactions between the tag and reader, any tag's response representing a state in tag's model must correspond to at least one transition of the reader's model. On the other hand, any reader's challenge representing a state in reader's model must correspond to at least one transition of the tag's model.
- E_T represents the end state of a tag caused by a **Corrupt** action. A tag in E_T cannot perform functions any more. Because $\mathcal{A}$ can never compromise a reader, the reader does not have such states.

Definition 5 *(Size of state space).* The size of a model $M = (S, I, R, E)$ is defined as $|S|$, i.e. the size of its state space.

5.2. Vulnerability aware graphs

Although M_T and M_R describe how each component of RFID system responses and updates its states under the actions of attackers, they are still inconvenient to express the attacking procedure. Moreover, it is difficult to find all the potential attacks with M_T and M_R only. Hence, we develop a random graph-based methodology, termed as Vulnerability Aware Graphs (VAGs). VAGs explicitly depicts the relevance of states in M_T and M_R from three aspects: the causal relationship of

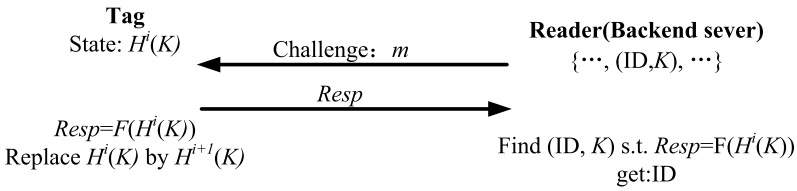

Fig. 2. OSK protocol.

a pair of challenge and response within one session, relevance of several sessions which may be utilized by $\mathcal{A}$, interactions between tag and reader with the existence of $\mathcal{A}$.

There are many non-equivalent ways to define random graphs. In this paper, we adopt the definition used in [21] that the formation of a random graph is a random process, which has advantages on describing the generation of VAGs. Moreover we extend the standard model of the random graphs by allowing slings and directed edges.

VAGs contains a pair of directed random graph, one reflects the interaction between tag and the adversary called *Tag Graph* (TG), while another reflects the interaction between reader and the adversary called *Reader Graph* (RG). The two graphs are constructed according to their state transition models. The vertices in TG and RG denote the states in M_T and M_R, respectively. Edges in the graphs represent state transitions caused by **SendTo*** or **Corrupt** actions taken by $\mathcal{A}$. Each possible edge occurs in a random graph with a certain probability. Each graph has a starting vertex without predecessor, representing the initial state I.

Specifically, for $M_T = (S_T, I_T, R_T, E_T)$, each $s_i^t \in S_T$ corresponds to a vertex on TG. $R_T(s_i^t, s_j^t)$ corresponds to a directed edge which starts from vertices s_i^t and ends at $s_j^t.e_t \in E_T$ is a vertex without an edge beginning from it. For $M_R = \{S_R, I_R, R_R, \emptyset\}$, each $s_i^r \in S_R$ corresponds to a vertex on RG, $R_R(s_i^r, s_j^r)$ corresponds to a directed edge which starts from vertices s_i^r and ends with s_j^r.

M_T and M_R can be constructed as discrete time step models to create TG and RG, respectively. The formation of VAGs is as follows: each graph starts with a starting vertex (I). At each time step, all the possible edges starting from the existing vertex set are added, and then all the corresponding vertices at the end of the edges that have not been created yet are generated as their successors. These vertices then are labeled with ⟨*Internal State, External Output*⟩, and added into existing vertices set. The iterations continue until no new edges can be added. Each directed edge between two vertices stands for an action taken by $\mathcal{A}$ to make a state transition. The probability that an edge occurs in the graph equals to the probability that $\mathcal{A}$ can take the corresponding action.

Definition 6 *(Path).* A path π in VAGs indicating the transition system $M = (S, I, R, E)$ is either an infinite or finite sequence of state transitions $R(s_i, s_{i+1}) \in R$, which connects a sequence of states $\{s_0, \ldots, s_i, \ldots, s_k\}, s_i \in S, 0 \le i \le k$.

VAGs aim to automatically analyze and benchmark RFID security protocols, to detect the flaw of the protocols and point out potential attacks. Any interactions series between a tag (or reader) and $\mathcal{A}$ can be viewed as a path on TG (RG).

We draw the VAGs of the well-known OSK protocol [20] as an example. The OSK protocol is shown in Fig. 2. Both $H(.)$ and $F(.)$ are hash functions. Initially, tag has initial information K, and backend server stores all the (ID, K) pairs, which are different for each tag in a system. For the first time a tag receives a challenge m from a reader, it calculates $F(K)$, where K is its current secret. Then it sends $RespF(K)$ to the reader, and updates its secret $K = H(K)$. A tag repeats above operations each time it gets a challenge.

For example, after i session, the reader sends the $(i+1)$-th challenge. After receiving m, the tag computes $F(.)$ using its current secret, which is $H^i(K)$, $(H^i(K) = H(H(\ldots H_i(K))))$. Then the tag sends *Resp*, i.e., $F(H^i(K))$, to the reader, and updates its secret to $H^{i+1}(K)$. The reader receives *Resp* from the tag, and calculates $F(H^i(K))$ for each (ID, K) pair. If $F(H^i(K)) = Resp$, the tag will be accepted. Note that OSK proposes a fixed upper bound n of the number of time steps over which tags are operated. After the n-th interrogation, the tag yields random output, which cannot be accepted by reader.

The VAGs of OSK is shown in Fig. 3, we set n as three for simplicity, which won't distort the expression of OSK. We use **STT**, **STR**, and **Cor** to denote **SendToTag**, **SendToReader**, and **Corrupt**, respectively.

On the Tag Graph, the tag is on state I_t initially with Internal State K. If $\mathcal{A}$ knows the specifications of OSK, $\mathcal{A}$ can generate the challenge m, and execute **SST** (m) to send a message m to the tag. When the tag is on state I_t, **SST** (m) results in a state transition $R_T(I_t, s_1^t)$ from I_t to s_1^t on the tag. Upon reaching s_1^t, the tag returns a response $F(K)$ and updates its Internal State to $H^1(K)$. Once $\mathcal{A}$ execute **SST** (m) again, the tag is forced to transit its state from s_1^t to s_2^t. Upon reaching s_2^t, the tag returns a response $F(H^1(K))$ and then updates the Internal State to $H^2(K)$. When a tag reaches the state s_3^t, **STT** taken by $\mathcal{A}$ only causes a transition of $R_T(s_3^t, s_3^t)$, which means that the tag does not change the state. s_4^t stands for the state that the tag is compromised by $\mathcal{A}$ using **Corrupt**. **Corrupt** does not change tag's state, but alters the type of state from normal to end.

On the Reader Graph, initially, the reader is on state I_r with Internal State equal to I_t and a challenge m at the beginning of state transitions. Any legitimate response will cause the reader leaves I_r to other states. In this example, **STR**

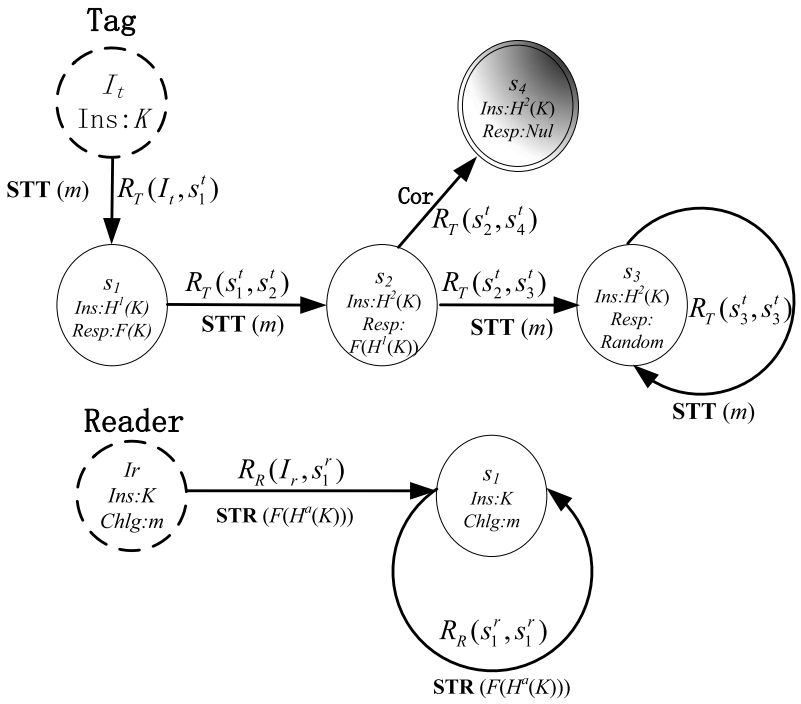

Fig. 3. VAGs of OSK protocol.

$(F(H^a(K))), (a = 0, 1, 2)$, represents the action that sends message $F(H^a(K))$ to reader. This action triggers the reader's operation of transferring the state from I_r to s_1^r. Then, the reader will not change the state until receiving **STR** $(F(H^a(K)))$ again.

6. Vulnerability aware graphs

We consider the passive and active adversaries separately. For an active adversary $\mathcal{A}$, we benchmark the safety properties of RFID security protocols from *indistinguishability*, *security*, and *functionality*. From the perspective of the graphs, actions of $\mathcal{A}$, constituting an attack, can be mapped to a harmful path on either TG or RG. Different from traditional attack graphs or models, we consider the violation of safety properties as some kinds of paths, which show harmful characteristics, instead of checking whether the paths will reach certain unsafe states.

In this section, we present our benchmarking methodology, which detects the potential flaws of the RFID protocol and define the harmful paths on VAGs. We also discuss how to quantify the probability that $\mathcal{A}$ can exploit the flaw to launch an attack. For a passive adversary, it can only pose a threat to the *indistinguishability* of tags. We discuss the benchmarking methodology for passive attacks in the next section.

6.1. Indistinguishability

Indistinguishability is merely related to TG, because it only concerns the relationship between tag's responses and adversary's challenges. On TG, it can be observed that *indistinguishability* of an RFID security scheme depends on whether there is a distinguishable path on TG. A distinguishable path exists if relations between challenges and responses on the path can be recognized as a characteristic by $\mathcal{A}$. Here we just give several popular characteristics on the graph that might be exploited by adversary. In practice, more characteristics can be found using our model.

- **Self-loop:** a self-loop on TG indicates that there is an $R_T(s_i^t, s_i^t)$. It implies $\mathcal{A}$ can take some actions to make the tag return the same response.
- **Loop:** a loop on TG indicates that there is a series of successive $R_T(s_i^t, s_j^t)$. that begins and ends with the same state. It implies $\mathcal{A}$ can take some actions to make the tag return the same response periodically.

Definition 7 *(Distinguishable path).* A path on TG where the External Outputs (Resps) on it show some certain characteristics, for example self-loop or loop.

If there exists a distinguishable path on a TG, $\mathcal{A}$ may be able to walk along the path by performing a series of attack actions, and hence can observe the characteristics. Then $\mathcal{A}$ can distinguish the tag from other tags based on the observation. Although the characteristics can be multifarious, it must be noticed that if there exists such characteristics on TG, the only thing that $\mathcal{A}$ should do is to find the path holding the characteristics. Since different challenges will lead to different responses represented by different paths on TG, to observe the characteristic of a path, $\mathcal{A}$ should decide the challenge (message m in **SendToTag**) in each interaction to the tag, so that the responses of the tag are in accordance with those responses at each tag's state along the path. Hence, we get the following conclusion:

Proposition 1. *The ability that $\mathcal{A}$ can distinguish a tag from others is defined as the probability that he or she can choose the correct challenges along the distinguishable paths on TG.*

6.2. Security

Security means that $\mathcal{A}$ is not able to cheat a reader, or make the reader to accept responses from fictitious tags. If the reader accepts a legitimate (uncompromised) tag, but the tag does not have a matching interaction with the reader, it means that $\mathcal{A}$ breaks down the security of the RFID system. A matching interaction (successful execution of the protocol) means that the reader and tag exchange challenges and responses well interleaved and faithfully (but may be with some time delay) over the execution of a session [10]. We make the following definitions to describe that if a tag and reader are under a proper execution of the protocol.

Definition 8 *(Half-matching states).* Suppose state s_t on TG and s_r on RG, if $s_t \cdot resp$ is related to a transition starting from s_r on RG, **OR** the $s_r \cdot chlg$ is related to a transition pointing to s_t on tag graph, these two states are half-matching states, and denoted as $\langle s_t \cdot resp, s_r\rangle$ or $\langle s_r \cdot chlg, s_t\rangle$. We say there exists a half-matching between s_r and s_t if at least one of the above relation is true.

Definition 9 *(Matching states).* Suppose state s_t on TG satisfies $\langle s_r \cdot chlg, s_t\rangle$, **AND** s_r on RG satisfies $\langle s_t \cdot resp, s_r\rangle$. The two states are defined as matching states. We say there exists a matching between s_t and s_r. The reader and tag on matching states imply that the interaction between them is legal to the protocol.

Definition 10 *(Mismatching states).* Suppose state s_t on TG and s_r on RG. If $s_t \cdot resp$ is **NOT** related to any transition pointing to s_r on RG and $s_r \cdot chlg$ is **NOT** related to any transition pointing to s_t on TG, s_t and s_r are named mismatching states. If a tag and a reader are on such states, they should not be in the same session of the protocol. We say that s_t and s_r are mismatching.

A pair of reader and tag is on certain matching states before $\mathcal{A}$ launches an attack. $\mathcal{A}$ makes the reader to accept a fictitious tag by counterfeiting the response of a legitimate tag. A successful execution of the protocol can be mapped to several pairs of matching states which have $\langle s_t \cdot resp, s_r\rangle$, $\langle s_r \cdot chlg, s_t\rangle$ on VAGs. Generally, there are two ways to counterfeit the response. First, $\mathcal{A}$ records responses generated by the legitimate tag, and replays them later directly. Second, $\mathcal{A}$ guesses a message that is identical to a real one according to the knowledge she knows about the system. In order to get enough information to answer the challenge, $\mathcal{A}$ should find a path which contains the information $\mathcal{A}$ needed on the TG.

Proposition 2. *Given a matching state $\langle s_t, s_r\rangle$, if $\mathcal{A}$ can find some paths on TG from state $s_{t'}$ which is half-matching with s_r, i.e. $\langle s_{t'} \cdot resp, s_r\rangle$, to state s_t, $\mathcal{A}$ may counterfeit the response of s_t, and make the fake response to be accepted by reader.*

6.3. Functionality

Functionality means correctness of the protocol execution. Since the reader is powerful and placed in somewhere safety, $\mathcal{A}$ cannot break it down. However $\mathcal{A}$ can disturb the function of the RFID system, e.g. making the reader to reject a legitimate tag or preventing a tag from recognizing a legitimate challenge. It implies that $\mathcal{A}$ is capable of enforcing a series of actions both on TG and RG, so that the tag cannot recognize the reader, or vice versa, on their current states.

Proposition 3. *The attack starts from a pair of matching states. If $\mathcal{A}$ can find a path on TG or RG, such that the end states of the path are mismatching states, then functionality was crashed at the end states.*

There are two methods that $\mathcal{A}$ deliberately makes the tag unrecognizable to the reader. One is to find some paths on TG from current half-matching state $\langle s_t \cdot resp, s_r\rangle$, to another state $s_{t'}$ that is mismatch with s_r. Another is to find some paths on RG from $\langle s_t \cdot resp, s_r\rangle$ to a state $s_{r'}$ that is a mismatching state with $s_t \cdot resp$. $\mathcal{A}$ then attempts to take a series of actions represented by the directed edges along the path. If no interaction between the tag and reader happens, the reader's state does not change while the tag's state is driven to the destination state of the path by $\mathcal{A}$.

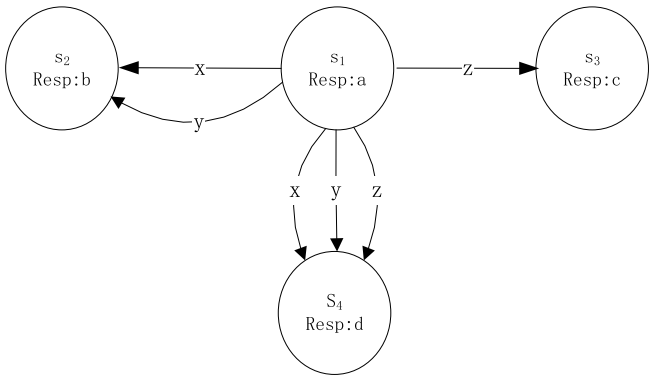

Fig. 4. Example of path probability calculation.

The strategies of $\mathcal{A}$ making tag to deny the recognition of the reader are similar to above two methods. The difference is that upon current state $\langle s_r \cdot chlg, s_t \rangle$, $\mathcal{A}$ should find a path on which the end state is either mismatch with $s_r \cdot chlg$ on RG or mismatch with state s_t on TG.

6.4. Quantitative

As mentioned above, flaws on *privacy*, *security* and *functionality* can all be represented as harmful paths. Therefore, in our quantitative analysis, we only consider the statistics of all harmful paths, without differentiating the paths by flaw types. However, our measurement can also be used to analyze a certain kind of flaws represented by a type of paths. We define three safety factors which are used to evaluation security protocols of RFID systems:

- *The number of flaws* (f_n): indicated by the number of harmful paths on VAGs.
- *The imperceptibility of a flaw* (f_i): the probability that a flaw can be utilized by $\mathcal{A}$.
- *Flaws rate* (f_r): the total number of states which are on harmful paths in VAGs.

It must be emphasized that all the three factors determine the safety of a protocol together. Impacts of each factor on the safety of a protocol should be considered comprehensively. Since all the harmful paths can be located on VAGs, we can calculate f_n *and* f_r easily. In the following, we discuss how to calculate f_i of a path.

A flaw being utilized by $\mathcal{A}$ means that $\mathcal{A}$ can walk along the harmful path representing the flaw with a high probability. The probability that $\mathcal{A}$ can obstacle the safety of a protocol by doing a series of actions along a harmful path on VAGs is:

$$C_{path:s_i \to s_j} = \prod_{si \to s_p \in s_i \to s_j} \Pr\big(R_{VAGs}(s_i, s_p)\big) \tag{1}$$

$\Pr(R_{VAGs}(s_i, s_p))$ is the probability that $\mathcal{A}$ drives the state transition on VAGs from s_i to s_p. It is determined by two conditional probabilities: the probability of an action that can make transition from s_i to s_p, and the probability that $\mathcal{A}$ may carry out the action. The probability that $\mathcal{A}$ can carry out an action for state transition is defined as the probability that $\mathcal{A}$ can choose the related correct message m.

$$\Pr\big(R_{VAGs}(s_i, s_p)\big) = \sum_{m} \Pr\big(s_i \to s_p \mid \mathbf{STT}(m)\big) \times \Pr(m \mid \mathbf{RFT}, \mathbf{RFR}) \tag{2}$$

$\Pr(s_i \to s_p \mid \mathbf{STT}(m))$ is the probability of state transition from s_i to s_p on the condition that $\mathcal{A}$ uses message m to access **STT**. It can be obtained from the graphs as one over the number of transitions related to m which starts from s_i.

$\Pr(m \mid \mathbf{RFT}, \mathbf{RFR})$ is the probability that $\mathcal{A}$ chooses the correct m and used it to access oracle **SendToTag** based on the information received from previous interactions. $\mathcal{A}$ can obtain the information by accessing **ReadFromTag** (**RFT**) and **ReadFromReader** (**RFR**) to eavesdrop challenges from the reader and responses from the tag. Note that the amount of messages obtained by $\mathcal{A}$ is determined by the adversary's capability. After obtaining messages from the tag and reader, $\mathcal{A}$ executes the information and has a probability to guess the desired message, which depends on the relevance of messages sent by tag and reader previously under a certain protocol. If there is strong relevance among the messages, $\mathcal{A}$ would have high probability to guess the desired message.

We give an example to illustrate how to calculate $\Pr(R_{VAGs}(s_i, s_p))$, as shown in Fig. 4. x, y, and z are three messages in **STT** that can incur the transition of state. Assume they are independent and $\mathcal{A}$ has no knowledge on them, so $\mathcal{A}$ guesses each message with the same probability of 1/3. From state s_1, x can cause $R_{VAGs}(s_1, s_2)$ or $R_{VAGs}(s_1, s_4)$, y can cause $R_{VAGs}(s_1, s_2)$ or $R_{VAGs}(s_1, s_4)$, and z can cause $R_{VAGs}(s_1, s_3)$ or $R_{VAGs}(s_1, s_4)$. We calculate the probabilities that $\mathcal{A}$ can trigger a certain state transition at state s_1 as follows:

$$\Pr\big(R_{VAGs}(s_1, s_2)\big) = \sum_m \Pr\big(s_1 \to s_2 \mid \mathbf{STT}(m)\big) \times \Pr(m) = \frac{1}{3} \times \frac{1}{2} + \frac{1}{3} \times \frac{1}{2} = \frac{1}{3}$$

$$\Pr\big(R_{VAGs}(s_1, s_3)\big) = \sum_m \Pr\big(s_1 \to s_3 \mid \mathbf{STT}(m)\big) \times \Pr(m) = \frac{1}{3} \times \frac{1}{2} = \frac{1}{6}$$

$$\Pr\big(R_{VAGs}(s_1, s_4)\big) = \sum_m \Pr\big(s_1 \to s_4 \mid \mathbf{STT}(m)\big) \times \Pr(m) = \frac{1}{3} \times \frac{1}{2} + \frac{1}{3} \times \frac{1}{2} + \frac{1}{3} \times \frac{1}{2} = \frac{1}{2}$$

7. Measuring indistinguishability of tags under passive attacks

Since the passive adversaries only eavesdrop on the forward and backward channels between tags and readers, which does not disturb the communication between tags and readers, they can only pose a threat to the *indistinguishability* of tags. In this section, we represent how to measure the *indistinguishability* of RFID tags under passive attacks.

Although passive adversaries can be formalized well by using the atomic actions in Definition 4, and passive attacks can certainly be represented by VAGs, passive attackers, in essence, do not take any action to force tags or readers make responses to counterfeit challenges. It implies the VAGs will degenerate to a pair of graphs indicating the interactions between a pair of reader and tag. Thus, we propose another convenient method to measure the *indistinguishability* of tags under passive attacks.

From the perspective of passive adversaries, a tag is considered as a black-box in the sense that it only knows the challenge–responses pairs, but knows nothing about the internal state of it. For any challenge–response mechanism, we first measure the *indeterminacy* (uncertainty) of the challenges and responses, and the *mutual dependence* of the challenge–response pairs respectively as follows.

Definition 11 *(Indeterminacy of challenges).* The indeterminacy of challenges C of a tag is the entropy of it, denoted by $H(C)$ as follows

$$H(C) = -\sum_{c \in \partial} p(c) \log p(c),$$

where ∂ is the domain of the challenges, c is one of the challenges, $p(c)$ is the probability c occurs. $H(C)$ can be used to measure the feasibility that an adversary guesses a challenge. By pretending a legal challenge, the adversary can get a particular response.

Definition 12 *(Indeterminacy of responses).* The indeterminacy of a response R is the entropy of it, denoted by $H(R)$ as follows.

$$H(R) = -\sum_{r \in \phi} p(r) \log p(r),$$

where ϕ is the domain of the response, r is one of the responses, $p(r)$ is the probability r occurs. $H(R)$ can be used to measure the feasibility that an adversary guesses a response. By pretending a legal response, the adversary can cheat the reader.

Definition 13 *(Mutual dependence of a challenge–response pair).* The mutual dependence of a challenge–response pair (C, R) is the mutual information between them, denoted by $I(C, R)$ as follows

$$I(C, R) = \sum_{r \in \phi} \sum_{c \in \partial} p(c, r) \frac{\log p(c, r)}{p(c)p(r)},$$

where $p(c, r)$ is the joint probability distribution of c and r. $I(C, R)$ is used to measure how much a certain challenge tells us about the corresponding response. i.e., how much the entropy of a response is reduced if we know a certain challenge?

Then given a pair of tag and reader, we measure the *indistinguishability* of the tag as follows:

Definition 14 *(Indistinguishability).* We use the joint entropy to measure the indistinguishability a challenge–response pair (C, R), denoted by $H(C, R)$, as follows

$$H(C, R) = -\sum_{c \in \partial} \sum_{r \in \phi} p(c, r) \log p(c, r).$$

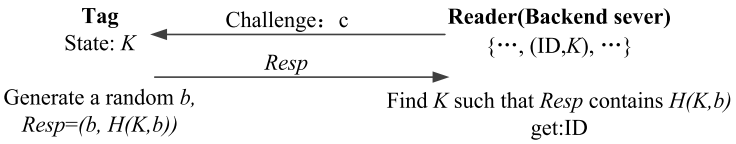

Fig. 5. WRSE's randomized hash-lock protocol.

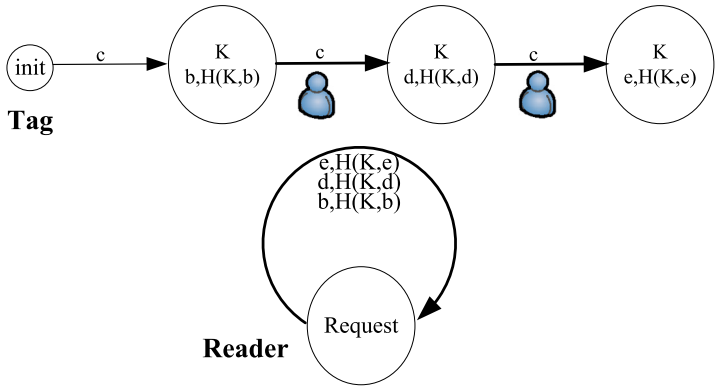

Fig. 6. VAGs of WRSE.

We use the joint entropy to measure how much entropy is contained in RFID systems which use challenge–response mechanism. C and R describe related events, the total entropy of the systems should be as follows:

$$H(C, R) = H(C) + H(R) - I(C, R).$$

By measuring the challenge–response pairs of a certain tag, we can evaluate the *indistinguishability* of such tag. In any RFID system, for a tag and reader pair, a large $H(C, R)$ implies high *indistinguishability* of the tag.

8. Benchmarking case study

In this section, we first apply the benchmarking methodology to WRSE's randomized hash-lock scheme [ ], and a variant of OSK protocol [8,22] (VOSK), to show the correctness and effectiveness of our method. We generate the VAGs of WRSE and VOSK and present our analysis. The security parameter k is related to the size of state space. For a practical protocol used in RFID systems, k should be sufficiently large to guarantee that an adversary cannot exhaust all the state space via brute force attacks. The generation of VAGs will suffer the state explosion problem as k increases. In this paper, we do not aim to develop a method to resolve the state explosion issue, instead we can use the existing abstraction techniques for model checking [23,24] to generate an abstract model that contains a smaller set of states, while preserving safety properties of original graph. We can also use on-the-fly techniques [25,26] to avoid generating the paths that may occur with extremely small probabilities.

8.1. WRSE's randomized hash-lock protocol

We illustrate WRSE's randomized hash-lock (WRSE [31]) in Fig. 5. $H(.)$ is a hash function. Each tag has secret information K, and backend server stores all the (ID, K) pairs, which are different for each tag in a system. Reader sends a constant challenge c. After receiving c, tag chooses a random k-bit *nonce*, and calculates $H(K, nonce)$. Then it sends *Resp* as $(nonce, H(K, nonce))$ to reader. Reader searches for K among all the (ID, K) pairs such that r_0 contains $H(K, nonce)$. Reader accepts the tag if such a K exists and rejects otherwise.

For providing an intuitive impression of WRSE, we first give a VAGs of WRSE with quite small state space in Fig. 6. We set that the nonce is chosen form $X = \{b, d, e\}$. We depict all the states and transitions of WRSE under the setting of above parameters.

On TG of Fig. 6, we use bold arrow lines to represent the paths that $\mathcal{A}$ can find. $\mathcal{A}$ gets the challenge c on RG by accessing **RFR**, then accesses **STT** (c) to query tag repeatedly. As a result, $\mathcal{A}$ can force the tag to transmit its state from s_1 to s_2 and s_3, and launch a spoofing attack to disrupt the *security* of the system; that is to say, $\mathcal{A}$ can access **STR** to cheat the reader to accept a fake tag.

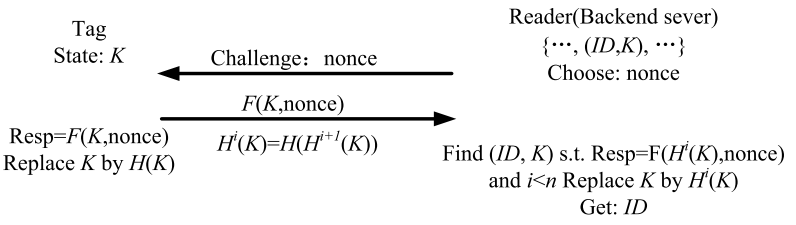

Fig. 7. A variant of OSK protocol.

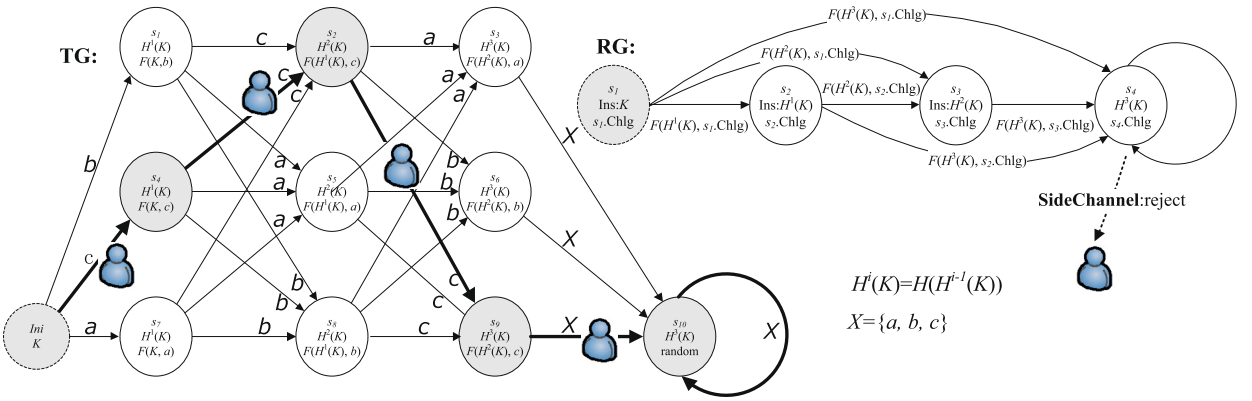

Fig. 8. VAGs of VOSK.

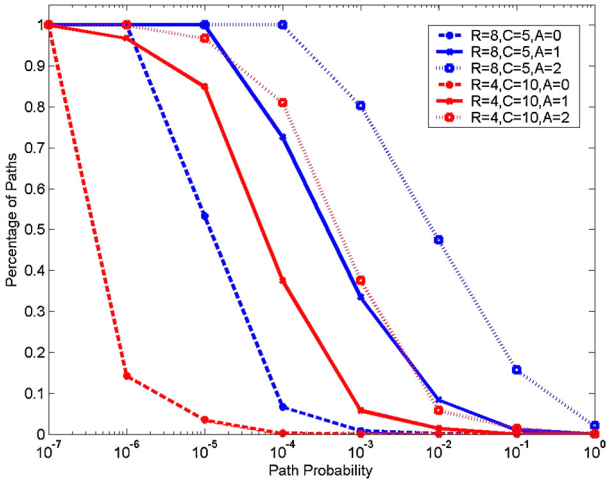

Fig. 9. A variant of OSK protocol.

8.2. A variant of OSK protocol

We illustrate VOSK in Fig. 7, which has two differences compared with original OSK. First, the reader sends the challenge with a nonce (a random α-bit string). Second, after the backend server accepts a tag, the reader updates current K.

For providing an intuitive impression of VAGs, we first give a VAGs of VOSK with quite small state space in Fig. 8. We set n as 4, and the nonce is chosen form $X = \{a, b, c\}$. We depict all the states and transitions of VOSK under the setting of above parameters.

On TG of Fig. 8, we use bold arrow lines to represent the paths that $\mathcal{A}$ can find. $\mathcal{A}$ gets the $s_1 \cdot Chlg$ on RG by accessing **RFR**, then accesses **STT** $(s_1 \cdot Chlg)$ to query tag repeatedly. As a result, $\mathcal{A}$ can find the path $Ini \to s_4 \to s_2 \to s_9 \to s_{10}$ and launch a DoS attack to disrupt the *functionality* of the system. Moreover, $\mathcal{A}$ can access **SideChannel** to learn that the tag is rejected by reader, which is also a distinguishable characteristic for tracing.

We design an algorithm to generate VAGs and calculate the probability that each path is found by $\mathcal{A}$. Fig. 9 shows the TG of VOSK with 50 states generated by our algorithm. The path with red solid line indicates a path located by $\mathcal{A}$ to launch a DoS attack.

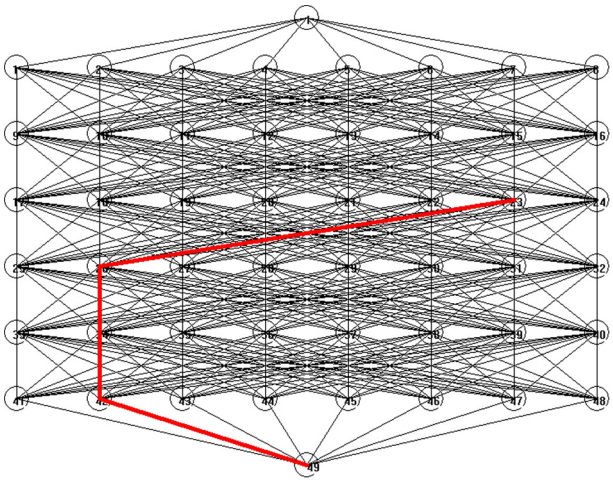

Fig. 10. VAG of VOSK generated automatically.

In Fig. 10, the probability for the adversary to locate each path is calculated, and we compare the number of paths with different probabilities when changing the size of state space and the amount of knowledge about the system that $\mathcal{A}$ already has. In Fig. 8, R is related to the length of nonce (α), C denotes the value of n, A indicates the number of actions $\mathcal{A}$ has taken. It can be observed that in VOSK protocol, after one attack action, the probability that adversary can launch an attack has greatly increased. It also can be seen that under the same state space, a larger n will result in higher security in VOSK protocol.

9. Conclusion

In this paper, we propose a random graph-based methodology to facilitate the benchmarking and quantitative evaluation for RFID security protocols. The methodology includes a fine-grained and accurate model for expressing adversary's capabilities, so that the attacks launched by adversaries can be resolved into a set of atomic actions. We formalize how the adversary intervenes in the RFID system by using the state transition models for both the tag and reader. Based on these models, we propose a random graph-based method, VAGs, to reflect the interactions between the adversary and tag, adversary and reader, respectively. By investigating the characteristic that a harmful path should hold, we use VAGs to exploit the flaws in RFID security protocols and to analyze the attack patterns of adversary. Moreover, we discuss the probability an adversary can successfully find the flaws and launch an attack to the RFID system.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61300199, 61303221, 61173171, 61472068), the Fundamental Research Funds for the Central Universities (233201400111, ZYGX2012J072), Innovation Program of Shanghai Municipal Education Commission (12ZZ060), and China Postdoctoral Science Foundation funded project (2014M550466).

References

[1] R. Oh, J. Park, A development of active monitoring system for intelligent RFID logistics processing environment, in: International Conference on Advanced Language Processing and Web Information Technology, 2008, pp. 358–361.
[2] Y. Lee, F. Cheng, Y. Leung, Exploring the impact of RFID on supply chain dynamics, in: Conference on Winter Simulation, 2004, pp. 1145–1152.
[3] M. Bertolini, G. Ferretti, G. Vignali, A. Volpi, Reducing out of stock, shrinkage and overstock through RFID in the fresh food supply chain: evidence from an Italian retail pilot, Int. J. RF Technol., Res. Appl. 4 (2013) 107–125.
[4] H. Hsu, H. Liao, A mobile RFID-based tour system with instant microblogging, J. Comput. Syst. Sci. 77 (4) (2011) 720–727.
[5] T. Staake, F. Thiesse, E. Fleisch, Extending the EPC network: the potential of RFID in anti-counterfeiting, in: ACM Symposium on Applied Computing, 2005, pp. 1607–1612.
[6] J. Jang-Jaccard, S. Nepal, A survey of emerging threats in cybersecurity, J. Comput. Syst. Sci. 80 (5) (2014) 973–993.
[7] X. Zhang, B. King, Modeling RFID security, in: Conference on Information Security and Cryptology, in: LNCS, vol. 3822, 2005, pp. 75–90.
[8] S. Vaudenay, On privacy models for RFID, in: ASIACRYPT, in: LNCS, vol. 4833, 2007, pp. 68–87.
[9] A. Juels, S. Weis, Defining strong privacy for RFID, in: Pervasive Computing and Communications Workshops, 2007, pp. 342–347.
[10] I. Damgard, M. Pedersen, RFID security: tradeoffs between security and efficiency, in: CT-RSA, 2008.
[11] G. Avoine, Adversarial model for radio frequency identification, IACR E-print, vol. 49, 2005.
[12] S. Jha, J. Wing, Survivability analysis of networked systems, in: International Conference on Software Engineering, 2001, pp. 307–317.
[13] O. Sheyner, Scenario graphs and attack graphs, Thesis, Carnegie Mellon University, 2004.
[14] P. Ammann, D. Wijesekera, S. Kaushik, Scalable, graph-based network vulnerability analysis, in: ACM Conference on Computer and Communications Security, CCS, 2002, pp. 217–224.

[15] X. Ou, W. Boyer, M. McQueen, A scalable approach to attack graph generation, in: ACM Conference on Computer and Communications Security, CCS, 2006, pp. 336–345.
[16] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J. Wing, Automated generation and analysis of attack graphs, in: IEEE Symposium on Security and Privacy, S&P, 2002, pp. 273–284.
[17] S. Cheung, W. Chan, P. Lee, L. Ni, P. Ng, A combinatorial methodology for RFID benchmarking, in: RFID Academic Convocation, 2006, pp. 1–6.
[18] M. Buettner, D. Wetherall, An empirical study of UHF RFID performance, in: ACM International Conference on Mobile Computing and Networking, Mobicom, 2008, pp. 223–234.
[19] A. Juels, RFID security and privacy: a research survey, IEEE J. Sel. Areas Commun. 24 (2006) 381–394.
[20] M. Ohkubo, K. Suzuki, S. Kinoshita, Cryptographic approach to "privacy-friendly" tags, in: RFID Privacy Workshop, 2003.
[21] P. Erdos, A. Renyi, On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci 5 (1960) 17–61.
[22] G. Avoine, E. Dysli, P. Oechslin, Reducing time complexity in RFID systems, in: Selected Areas in Cryptography, SAC, 2006, pp. 291–306.
[23] A. Gupta, Learning abstractions for model checking, Thesis, Carnegie Mellon University, 2002.
[24] M. Lohrey, S. Maneth, M. Schmidt-Schauß, Parameter reduction and automata evaluation for grammar-compressed trees, J. Comput. Syst. Sci. 78 (5) (2012) 1651–1669 (Special Issue: Cloud Computing).
[25] R. Gerth, D. Peled, M. Vardi, P. Wolper, Simple on-the-fly automatic verification of linear temporal logic, in: International Symposium on Protocol Specification, Testing and Verification, PSTV, 1995, pp. 3–18.
[26] A. Bouajjani, S. Tripakis, S. Yovine, On-the-fly symbolic model checking for real-time systems, in: Real-Time Systems Symposium, RTSS, 1997.
[27] M. Alizadeh, M. Salleh, M. Zamani, J. Shayan, S. Karamizadeh, Security and performance evaluation of lightweight cryptographic algorithms in RFID, in: Recent Researches in Communications and Computers, 2012.
[28] E. Vahedi, R.K. Ward, I.F. Blake, Security analysis and complexity comparison of some recent lightweight RFID protocols, in: Computational Intelligence in Security for Information Systems, CISIS 2011, in: LNCS, vol. 6694, 2011, pp. 92–99.
[29] B. Niu, H. Li, X. Zhu, C. Lv, Security analysis of some recent authentication protocols for RFID, in: International Conference on Computational Intelligence and Security, CIS, 2011, pp. 665–669.
[30] C. Ni, Y. Li, R. Deng, T. Li, RFID privacy: relation between two notions, minimal condition, and efficient construction, in: ACM conference on Computer and Communications Security, CCS, 2009, pp. 54–65.
[31] S. Weis, S. Sarma, R. Rivest, D. Engels, Security and privacy aspects of low-cost radio frequency identification systems, in: International Conference on Security in Pervasive Computing, SPC, in: LNCS, vol. 2802, 2003, pp. 454–469.