Quantifying Information Leakage in Tree-Based Hash Protocols (Short Paper)

Karsten Nohl and David Evans

University of Virginia, Computer Science Department
{nohl, evans}@cs.virginia.edu

Abstract. Radio Frequency Identification (RFID) systems promise large scale, automated tracking solutions but also pose a threat to customer privacy. The tree-based hash protocol proposed by Molnar and Wagner presents a scalable, privacy-preserving solution. Previous analyses of this protocol concluded that an attacker who can extract secrets from a large number of tags can compromise privacy of other tags. We propose a new metric for information leakage in RFID protocols along with a threat model that more realistically captures the goals and capabilities of potential attackers. Using this metric, we measure the information leakage in the tree-based hash protocol and estimate an attacker's probability of success in tracking targeted individuals, considering scenarios in which multiple information sources can be combined to track an individual. We conclude that an attacker has a reasonable chance of tracking tags when the tree-based hash protocol is used.

1 Introduction

Radio Frequency Identification (RFID) systems provide more precise identification (right down to the item-level) and superior reliability over existing tracking systems, as well as the possibility of strong authentication. Their capabilities, however, also pose a threat to individual privacy. Several schemes have been proposed that preserve consumer privacy by obfuscating the tag identity from rogue readers. Some proposed schemes, such as Weis et al.'s [13] and Ohkubo et al.'s [8], provide strong privacy but cannot scale to large RFID systems because the workload for the backend system scales linearly with the number of tags in the system. Other schemes, such as the tree-based hash protocol first proposed by Molnar and Wagner [5], provide scalability but sacrifice some privacy. We focus on this protocol and describe it in Section 2.

Avoine et al. analyzed the degree to which privacy is scarified in the tree-based protocol and concluded that a serious privacy threat exists [1]. In this paper, we revisit their assumptions and derive a different attacker model that we believe better captures possible capabilities and motives of real-world attackers. In our model, the attacker wants to track a tag through the system and needs to distinguish that tag from all other tags. We find that the threat to privacy is even higher than Avoine et al.'s estimate.

We assume an active attacker who can send arbitrary messages to readers and tags of the system, but cannot invert the hash function. We also assume our

P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 228–237, 2006.

attacker can extract secrets from a limited number of tags. The attacker tries to learn as many bits of information as possible about each tag's identity with the final goal of distinguishing among passing tags. The amount of information that the adversary needs to successfully launch an attack depends on properties of the system and environment. The attack becomes harder if the system has more tags and also if more of these appear in the limited environment that the attacker probes. Our attacker model is different from previous models in that we consider the case in which the attacker sees only a subset of all tags in the system and tries to distinguish among those. The probability an attack will be successful increases with the amount of information that each tag leaks, and with the number of tags that are likely to stay together as a group. Our model does not make assumptions about how the attacker learns information about the tags other than that the attacker can extract all the key material from a number of captured tags. We focus on information leaked from the protocol layer; information leaked through side channels may further increase the risk of privacy compromise.

Our main contributions are an improved metric for information leakage that allows us to combine different information sources and that better follows the proposed attacker model (Section 3), an analysis of the tree-based protocol based on this metric (Section 4), and an analysis of the relevance of our results o realistic RFID systems 5. We conclude that the privacy risks associated with the tree-based hash protocol are more severe than previously thought.

2 Private Authentication Protocols

Several protocols have been proposed through which a tag can identify itself to a legitimate reader while preserving the customer's privacy against rogue readers.

Public-key cryptography would provide a clear solution to the privacy problem, but is usually too expensive to implement on RFID tags. All the protocols we consider employ symmetric cryptographic hash functions in which keys are shared between the tag and legitimate readers.

Weis et al. proposed a privacy-preserving RFID protocol in which the tag hashes a random value (nonce) with a secret key that is only known to the tag itself and all legitimate readers [13]. This linear hash protocol provides strong privacy (as defined in Section 3) but fails to provide the needed scalability for large RFID systems. The reader stores one key per tag and has to try all possible keys in the database. Every tag authentication requires $O(N)$ hashing operations where N is the number of tags in the system. Since RFID systems must scale to millions of tags, this cost becomes excessive.

To achieve scalability, Molnar and Wagner [5] proposed a protocol that achieves sub-linear workload in the backend system[1]. The main drawback is that secrets are shared among several tags. Hence, an attacker who can extract secrets from a given

[1] Other protocols have been devised that sacrifice reliability for better scalability [8][12]. Since we believe that most RFID applications require high availability, we do not consider these protocols viable solutions.

tag also learns some of the secrets stored on other tags. The tags are structured in a tree where each tree leaf is a tag. Secrets are assigned to each tree branch and every leaf stores all secrets on the path from the root to itself. The tree has a depth d. Each node in the tree (except for the leaves) has k children. Each tag holds d secrets, one for each level of the tree. Using the notation from Avoine et al. [1] we denote the ith secret on the j level of the tree as $r_{j,i}$. The secrets on each tag correspond to a unique path through the tree; hence, every tag has at least one secret that is not shared with any other tag.

To authenticate a tag in the tree, the reader initiates the protocol by sending a nonce, N_R. The tag responds with a second nonce, N_T, and one hash for each level of the tree. The tag response is: $N_T, H\left(r_{1,i}||N_T||N_R\right), \cdots, H\left(r_{d,i}||N_T||N_R\right)$, where H is a cryptographic hash function (our analysis assumes the attack cannot compromise H). The nonce provided by the tag provides privacy by making consecutive responses from the same tag unlinkable. The reader-supplied nonce prevents replay attacks. On reception, the backend system generates hashes for all possible secrets corresponding to the first-level branches with the two session-specific nonces. The one hash that matches the transmitted hash on this level points to a node on the next level. This step is repeated until a leaf is reached.

If the tree is balanced, it holds $N = k^d$ tags. On each of the d levels, up to k hashing operations are needed to find the responding secret. Hence, the database performs up to $dk = d\sqrt[d]{N}$ hashing operations. The tag performs d times the number of hash operations than were required with the linear hash protocol, and the transmitted response is approximately d times larger (ignoring the framing and protocol overhead that does not grow with d). The memory needed to store and process the hashes grows with d. Therefore, it is necessary to keep d small to minimize the tag processing and memory requirements. The parameter k does not affect the tag, but determines the computational cost in the backend database. When designing a tree it can be chosen more freely than d. It can also be dynamically adapted to a changing system size.

Different tags can have different probabilities of being broken. Tags that are more likely to be broken should have fewer secrets (i.e., be placed higher in the tree) than tags that are know to be hard to break. The question of the optimal number of secrets was answered by Poovendran and Baras [9] in the context of multicast keys. If tag i has a probability of being broken of p_i then the optimal number of secrets for this tag is $d_i = -log_k(p_i)$. Note that for the special case in which all tags have the same probability of being broken ($\forall i : p_i = \frac{1}{N}$), this resolves to a balanced tree as introduced earlier $d = -log_k(\frac{1}{N}) = log_k(N)$. The rest of this paper assumes equal probabilities of being captured for all tags and a balanced tree.

3 Privacy Definition

Several different notions of RFID privacy have been developed. The first papers that targeted RFID privacy [13][8] focused on the requirement that tags should protect product information from being disclosed. This is a weak notion because

it leaves tags traceable. A stronger property, *unlinkability*, means that an adversary should not be able to differentiate between readings that originated from the same tag and readings that originated from different tags.

A system achieves *strong privacy* when an adversary cannot distinguish between two tags with a probability better than random guessing [4]. Since scalable protocols have to sacrifice strong privacy, we need a more flexible measure of privacy. Our notion captures shades of privacy where a tag can be distinguishable from some tags but not from others.

Our notion of privacy is closely related to anonymity, which has been studied in the context of mix-nets [10][3]. Mix-nets try to make sender and recipient of a message anonymous. The anonymity set is defined as the set of all potential senders of a given message. The degree to which anonymity is achieved depends on the size of the anonymity set. Perfect anonymity is achieved if the set includes all members capable of sending messages in the system. The metric used by Serjantov and Danezis is similar to the metric we propose in this paper. Both are based on Shannon's information theory [11]. They use entropy to describe the number of possible elements in a group (in our case, the set of RFID tags in the system). Nohara et al. were the first to use entropy in the analysis of the tree-based RFID protocols [6][2]. They only considered the case of a single compromised tag and concluded that almost no information is leaked if the number of tags in the system is large enough. Our results are consistent with this, but extend to the more likely scenario where multiple tags are compromised.

Buttyán, Holczer and Vajda recently published an analysis of the privacy of tree-based hash protocols also employing an information-theoretic metric similar to ours [2]. Their notion of privacy is different from ours in that they employ the average anonymity set size as their metric. In this metric the impact of decreasing the anonymity set size is independent of the initial set size. We believe that the attacker's actual incentive is better modeled by a logarithmic measure. Decreasing the size from 100 to 50 should have the same impact as from 2 to 1 since both advances help to distinguish tags twice as well.

Measuring Privacy. We define privacy as the degree to which two authentication sessions of the same tag are not linkable. An authentication session is the interaction between a reader (legitimate or rogue) and a tag at the protocol level. Sessions are unlinkable if an attacker cannot discover whether two responses originated from the same tag with a probability better than random guessing. The highest degree of unlinkability exits if any pair of tags is indistinguishable. The metric that we derive in this paper measures the unlinkability as a value between zero (unlinkable) and $log_2(N)$ (all tags linkable). Our metric closely follows our attacker model as described in Section 1.

We measure privacy as the degree to which a member of the group is indistinguishable from other elements of the group. The degree to which elements in the group are distinguishable can be measured in bits. If we have a group of size N and the adversary can, with absolute certainty, divide our group into two

[2] Poovendran and Baras use entropy to analyze multicast keys [9].

disjoint subgroups of size $\frac{N}{2}$ each then we have disclosed 1 bit of information. We can extend this to two arbitrarily sized subgroups, S_1 and S_2, where $\frac{N}{s}$ tags are placed into group S_1 and the remaining $\left(1-\frac{1}{s}\right)N$tags are placed into S_2. The adversary can place every tag in either S_1 or S_2. We use I to denote average amount of information disclosed (that is, the amount of information that can be learned about all tags divided by the number of tags). The information disclosed is: $I = \frac{1}{s} \cdot log_2(s) + \frac{s-1}{s} \cdot log_2\left(\frac{s}{s-1}\right)$.

In general, an attacker will be able to split the group of all tags, G, into k disjoint groups, S_i, of arbitrary size. Then, the information disclosed is:

$$I = \sum_{i=1}^{k} \left(\frac{|S_i|}{|G|} \cdot log_2\left(\frac{|G|}{|S_i|}\right)\right). \tag{1}$$

The amount of disclosed information increases when there are more groups and is maximized when the groups are equal in size (This is consistent with Shannon's information theory that states that the entropy of a source grows as the probabilities of possible symbols become more similar [11]). Information theory also gives us that a $log_2(N)$-bit identifier uniquely identifies elements in a group of size N. The values of I range from $I = 0$ (strong privacy) to $I = log_2(N)$ (no privacy). In the latter case we can identify each tag uniquely, which means that we have N groups of size 1.

4 Information Leakage

The tree-based protocol shares secrets among tags, so extracting the secrets from one tag compromises the privacy of other tags. This section analyzes the amount of information that can be gathered by an adversary. The amount of information depends on the tree-structure and the tree positions of broken tags. We first look at the worst case in which the adversary can select the tags to compromise based on their tree position in Section 4.1 and then at the random case in Section 4.2.

4.1 Selected Tags Scenario

In the selected tag scenario, the attacker can select which tags to compromise. This enables the attacker to select tags such that the number of redundant secrets is minimized, thereby maximizing the information leakage. We consider the information leaked when an attacker breaks b tags, and denote the broken tags as $t_1, t_2, \cdots, t_b$. The first broken tag, t_1, always reveals d new secrets to the adversary. The second through k^{th} broken tags (recall that k is the number of children of a node), can each reveal between 1 and d new secrets. The number of new secrets depends on how many branches are shared between the broken tag and previously broken tags. This can be as few as one new secret if the tags are siblings in the tree. Assuming the worst case the tags $t_{(k^i+1)} \cdots t_{k^{i+1}}$ reveal $d-i$ new secrets each — that is, all secrets at level i are known to the attacker and each newly broken tag adds one secret to each level below i.

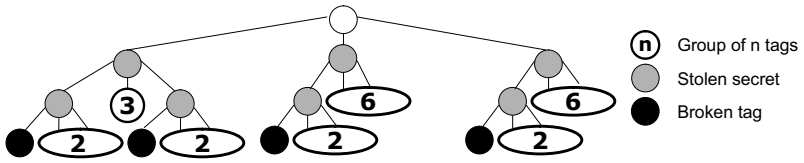


Fig. 1. Distinguishable groups of tags after 4 tags have been broken

For the purpose of our analysis we assume a completely filled k-ary tree with depth d, containing $N = k^d$ tags. The secrets have been extracted from b tags. The adversary always selects tags to break that maximize the number of secrets learned. We define level j of the tree as the deepest level on which all secrets are known: $j = \lfloor log_k(b) \rfloor$ On the next level of the tree, level $j+1$, the adversary knows b secrets. Recall, that we are considering the worst case first in which there exists as little redundancy among the secrets as possible. Each of these secrets is the root to a subtree with height $d - j$ with one known path from the root to one of the leaves. Each of these subtrees split the leaves of the tree into subgroups of size $\frac{N\cdot(k-1)}{k^{j+2}}, \frac{N\cdot(k-1)}{k^{j+3}}, \cdots, \frac{N\cdot(k-1)}{k^d}, \frac{N}{k^d}$. Maximum information is disclosed if the groups of tags are of similar size. Therefore, the remaining tags cluster in groups of only two sizes. These sizes are the ones closest to the average size.

Figure 1 shows an example of the maximal information leakage in a 3-ary tree, in which 4 tags have been broken. For the subgroups in which one of the leaves has been broken, the final level is either the broken tag, or one of two unbroken tags. The remaining unbroken keys at level 2 correspond to tag groups of size 3 and 6. The next broken tag should be selected from one of the groups of size 6.

The unbroken level j keys correspond to tag groups of two sizes, c_1 and c_2, where r_1 and r_2 are the numbers of times these groups appear. The overall number of groups add up to the number of keys at level j ($r_1 + r_2 = k^j$), because each node on level j has exactly one group (potentially with size 0) below it. Thus, $r_1 = b \bmod k^j$ and $r_2 = k^j - r_1$.

The number of nodes on the next level, k^{j+1}, is equal to the number of groups times their sizes plus the number of broken tags:

$$k^{j+1} = c_1 \cdot r_1 + c_2 \cdot r_2 + b$$

$$c_2 = c_1 + 1 \text{ and } c_1 = \frac{k^{j+1} - b - r_2}{r_1 + r_2}$$

For the example in Figure 1, we get one group of size 3 ($j = 1$; $r_1 = 1$; $c_1 = 1 = \frac{3}{k^j}$), two groups of size 6 ($r_2 = 2$; $c_2 = 2 = \frac{6}{k^j}$); in addition, there are 4 groups of size 2, and 4 groups of size 1 at level $j + 1$.

Using equation 1 we can compute the worst case average information leakage as

$$I(k,d,b) = b \cdot \left(\sum_{i=j+2}^{d} \left(\Psi\left(\frac{k-1}{k^i}\right)\right) + \Psi\left(\frac{1}{k^d}\right)\right) + r_1 \cdot \Psi\left(\frac{c_1}{k^{j+1}}\right) + r_2 \cdot \Psi\left(\frac{c_2}{k^{j+1}}\right)$$

where the information leakage due to a group of size σ is $\Psi(\sigma) = \frac{1}{\sigma} \cdot log_2(\sigma)$.

The first term quantifies the information leakage due to b subtrees, each of which contains one broken tag. The second and third term denote the leakage due to the groups of tags that are not part of these subtrees.

For the example in Figure 1, this formula resolves to $I = 3.132$ bits. After just 4 of the 27 tags in the tree have been broken (which means that 11 of the 39 secrets have been revealed), a significant portion of the maximally achievable information ($= log_2(27)$, approximately 4.75, bits) is disclosed.

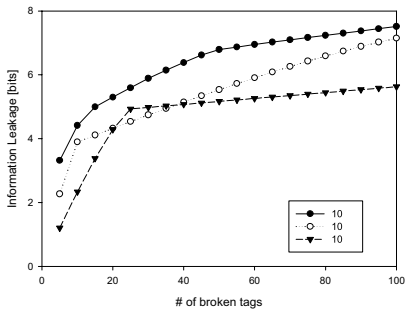
The information leakage for a few example cases is shown in Figure 2(a). The figure shows the amount of information leakage over the number of broken tags for several different system sizes. An attacker who compromises 20 tags in a system with 100,000 tags obtains 2.9 bits of information when a tree with depth 3 is used and 4.3 bits when a tree of depth 5 is used. These values are small enough to only allow tracking of individuals in very limited environments.

The worst case scenario will only occur if the attacker can select tags that maximize the number of different secrets compromised. This is entirely possible if the attacker has access to many tags. The attacker could probe every tag for secrets on the tag that match those that were already extracted from other tags, thus identifying a tag to break that has a high number of unknown secrets.

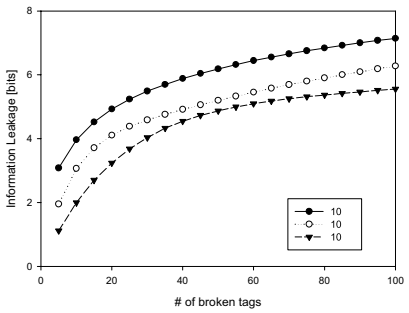
4.2 Random Tags Scenario

The attacker in this scenario breaks tags that are chosen at random. The information disclosure of this scenario cannot be easily captured in a closed-form equation. We choose to simulate this case instead.

We simulated the random case for systems with system sizes in between $N = 10^3$ and $N = 10^7$, and a tree depth $d = 5$, and number of broken tags up to $b = 100$. The results are shown in Figure 2(b). The difference between this simulated random case leakage and the selected tag leakage (Figure 2(a)) is at most 34% (for $N = 10^7$ and $b = 25$) and typically less than 10%. The average difference



(a) Tree with d=5, selected tags scenario



(b) Tree with d=5, random tags scenario

Fig. 2. Information leakage in the tree-based hash-protocol

over all simulated cases is 9%. Simulations of trees with different heights lead to similar results [7].

The information leakage in the random tags scenario is always upper-bounded by the selected tags scenario. Our results suggest that the closed-form solution for the selected tags scenario is tight enough, typically within a ten percent, for cases where attackers have no control over which tags they break. Since a smart attacker with access to many tags could obtain nearly the worst case information leakage, the derived closed-form solution can be used to analyze the information leakage in the tree-based protocol in nearly all scenarios.

5 Relevance

An attacker can only track people whose tags can be distinguished from all other people's tags. This definition is different from Avoine et al.'s [1]. They considered an attack to be successful if an attacker can distinguish between two tags. In our model an attacker needs to be able to distinguish a tag from all other tags for a tracking attack to be successful which we believe better captures a realistic attacker. In this section, we estimate the likelihood of a successful attack for different key parameters.

Our threat model and the described tracking attack are not limited to information disclosed at the protocol layer. The most notable additional source of information is the physical layer of a tag. Different tags have different physical characteristics[3].

Few bits of information are encoded in the number of tags that an individual carries. Additional information could be encoded in the timestamp of readings (e.g. if the same tag was always read at around the same time of the day).

Our analysis in limited to the information leaked on the protocol layer. Privacy on this layer can be seen as a required but not sufficient property of RFID privacy.

For simplicity of the analysis we assume that the tags are partitioned into $g = \lfloor 2^I \rfloor$ groups of equal size where I is the amount of information leakage. A second parameter of our attack, η, is the number of tags in the focus of the attacker (e.g. all the tags that have entered the subway system at a given day). Note that this number is typically much smaller than the total number of tags in the system. A tag can be uniquely identified if it is the only tag in one of the g groups. First we look at the case where every individual carries exactly one tag and then we consider the case in which multiple tags stay together as a group.

5.1 Tracking Single Tags

The probability that at least one tag can be uniquely identified (that is, this tag can be distinguished from all other tags) is

[3] Based on radio characteristics, several additional bits of information may be extracted from the tag. Our experiment and preliminary results are reported in an extended version of this paper [7].

$$P_1(g,\eta) = \left(\frac{g-1}{g}\right)^{\eta-1}.$$

The probability that at least j tags of the η tags can be identified is

$$P(g,\eta,j) = \prod_{i=1}^{j}\left(\frac{g-i}{g-i+1}\right)^{\eta-i}.$$

Given a system with 100,000 tags of which 20 have been broken, and a tree with depth 5, we get 20 groups ($g = 20$). The probability that in small group of tags ($\eta = 10$), half of the tags can be uniquely identified is 14%. As the number of tags grows, this probability becomes smaller.

5.2 Tracking Collections of Tags

For many RFID applications, it is common for each individual to carry several tags. Even if a given RFID application gives individuals only a single tag, other tags they carry for different RFID applications are equally helpful to the attacker in distinguishing the individual. We assume that these collections comprise randomly selected tags. The number of ways in which l tags can fall into the g groups is given by $\binom{g}{l}$. When combined with the earlier result, the probability that in a group of η individuals who each carry l tags, at least j can be uniquely identified is

$$P(g,\eta,l,j) = \prod_{i=1}^{j}\left(\frac{\binom{g}{l}-i}{\binom{g}{l}-i+1}\right)^{\eta-i}.$$

Looking at the example from the last section with $N = 10^5$, $d = 5$, $\eta = 10$ but now assuming two tags per individual ($l = 2$), the attacker can uniquely identify 5 individuals ($j = 5$) with a probability of 83%. If each individual carries 5 tags ($l = 5$), this probability exceeds 99%. Looking at an example of larger attack, we assume 50 compromised tags ($b = 50$ and $l = 5$); the probability of identifying half of 1,000 individuals ($\eta = 1000$, $l = 500$) is 88%.

These results illustrate that tracking attacks on large groups of individuals are practical under the assumption that each individual carries a fixed collection of tags.

6 Conclusion

The resource constraints of RFID tags, combined with the strict requirements for large-scale scalability and high availability, mean that strong privacy is not possible. All proposed protocols that provide strong privacy fail to scale to large systems or suffer from a degraded availability. The tree-based protocol provides a trade-off between privacy and scalability, but raises the need to better quantify the amount of privacy compromised.

Privacy must be measured in a way that accounts for a realistic attacker's ability to combine partial information to compromise individuals' privacy without necessarily being able to uniquely distinguish tags. Our proposed metric for information leakage provides useful guidance for estimating the privacy a system provides. An attacker is not likely to distinguish between individuals that each carry only a single tag, but is very likely to be successful in distinguishing individuals that carry several tags. If additional information sources are factored into the attack tracking of very large tag populations becomes entirely possible. Our results indicate that protocol designs previously considered to provide adequate privacy, may in fact be insufficient against more realistic threat models. Designers of RFID applications must be careful to balance the needs for scalability with realistic assessments of the threats of privacy compromise.

References

1. Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing time complexity in RFID systems. In *Selected Areas in Cryptography – SAC*, 2005.
2. Levente Buttyán, Tamás Holczer, and István Vajda. Optimal key-trees for tree-based private authentication. In *Privacy Enhancing Technologies Workshop – PET*, 2006.
3. Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Privacy Enhancing Technologies Workshop – PET*, 2002.
4. Ari Juels and Stephen Weis. Defining strong privacy for RFID, 2006.
5. David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *Conference on Computer and Communications Security – ACM CCS*, 2004.
6. Yasunobu Nohara, Sozo Inoue, Kensuke Baba, and Hiroto Yasuura. Quantitative evaluation of unlinkable id matching schemes. In *Workshop on Privacy in the Electronic Society – WPES*, 2006.
7. Karsten Nohl and David Evans. Quantifying information leakage in tree-based hash protocols. Technical Report CS-2006-20, University of Virginia, Computer Science Department, October 2006.
8. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to "privacy-friendly" tags. In *RFID Privacy Workshop*, 2003.
9. Radha Poovendran and John S. Baras. An information-theoretic approach for design and analysis of rooted-tree-based multicast key management schemes. *IEEE Transactions on Information Theory*, 2001.
10. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Privacy Enhancing Technologies Workshop – PET*, 2002.
11. C. E. Shannon. A mathematical theory of communication. 1948.
12. Gene Tsudik. YA-TRAP: Yet another trivial RFID authentication protocol. In *PerCom*, 2006.
13. Stephen Weis, Sanjay Sarma, Ronald Rivest, and Daniel Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *International Conference on Security in Pervasive Computing – SPC*, 2003.