Minimizing Information Leakage of Tree-based RFID Authentication Protocols using Alternate Tree-Walking

Kaleb Lee Juan Gonzalez Nieto Colin Boyd
{leekj, j.gonzaleznieto, c.boyd}@qut.edu.au

April 25, 2012

Abstract

The privacy of efficient tree-based RFID authentication protocols is heavily dependent on the branching factor on the top layer. Indefinitely increasing the branching factor, however, is not a viable option. This paper proposes the alternate-tree walking scheme as well as two protocols to circumvent this problem. The privacy of the resulting protocols is shown to be comparable to that of linear-time protocols, where there is no leakage of information, whilst reducing the computational load of the database by one-third of what is required of tree-based protocols during authentication. We also identify and address a limitation in quantifying privacy in RFID protocols.

1 Introduction

RFID is a wireless technology designed for convenient automatic identification of physical objects, originally intended to replace barcodes. However this convenience also comes risk ones privacy, as these objects are more commonly carried used in our daily lives. In particular it is possible to track people based on RFID attached objects they carry. Thus it is critical to ensure that only the minimum amount privacy is leaked when using this technology.

The most commonly used RFID networks are low-cost RFID networks. Such networks typically consists of three components, a back-end database, multiple readers and a large number of RFID tags. Whereas the database and reader are usually workstation class devices, tags are severely limited in terms of computational power and storage. A large number of research has been focused on increasing the privacy of in low-cost RFID networks, however an increase in privacy often comes at the cost of efficiency on the database. Of particular interest is tree-based protocols which provides requires comparatively very little computation but was later shown to be susceptible to information leakage. This paper presents a scheme to minimize the leakage of privacy leakage in tree-based protocols before presenting two protocols. In addition, this paper also identify a limitation, and thus propose an extension, of a current method of measuring piracy leakage in RFID protocols. The protocols proposed are shown to leak significant less information compared to current proposed protocols whilst only requiring one-third of the computation when pre-computation is used. Also discussed is the work required to attack the protocols, it is shown that an attacker is required to perform significantly more work to succeed.

2 Previous Work

This section will first outline a metric of measuring leakage of privacy in RFID protocols before discussing currently proposed protocols. Each of the current protocol types will also be analyzed using this measurement metric. Also of interest, is the amount of computation required for each of the scheme to achieve its level of privacy.

2.1 Measuring Privacy Leakage

The amount of privacy leakage has been analyzed and quantified in a number of works including [1, 2, 3, 4]. This paper will focus on the work in [2]. In the work, leakage is measured in the probability of an adversary distinguish between two tags in a privacy attack experiment. The experiment can be considered as follows:

1. The adversary draws one tag, T_0, and obtains it's full set of secrets. The tag is put back into circulation.
2. The adversary is than randomly given a tag T_j and is allowed to query the tag as much as it wants. However, the adversary is not allowed to reveal the secrets of the tags.
3. The adversary is now given two tags, T_1 and T_2 such that $T_j \in \{T_1, T_2\}$. The adversary wins the experiment if it can output i, such that $T_i = T_j$.

In the experiment, the adversary is only given the power to query tags, i.e. pretending to be a reader to interact with a tag. It should also be noted that the method is based on the following assumptions. Firstly, all secrets are chosen from uniformly random distribution, thus secrets should not be correlated with each other. Secondly, the adversary cannot carry out an exhaustive search over all possible values. Finally, the only way for an adversary to obtain the secret of a tag is by tempering with a tag.

2.2 Linear Protocols

Linear protocols as the name suggests requires a linear amount of computations per authentication attempt. As such, it is one of the most inefficient types of RFID protocol proposed. In the protocols, each tag shares a single unique secret, k, with the reader. A generic construction of a linear-time protocol based on the randomized hash-lock is shown in table 1. For every run of the protocol, the reader/database generates a random nonce, N_T and is sent to the tag as a the start of a protocol. The tag in turn generates another nonce, N_T, which is sent along with $H(N_R \Vert N_T \Vert k)$ where H is a pseudo-random function. The reader/database is than required to perform an exhaustive search of $H(N_R \Vert N_T \Vert k)$ for all secrets k in the system. Thus the database is required to perform a total of N operations per authentication attempt, where N is the total number of tags in the system.

Reader		**Tag**
k		k
	$\xrightarrow{N_R}$	
	$\xleftarrow{H(N_R \Vert N_T \Vert k), N_T}$	

Table 1: Linear Time Protocol

Aside from the high computational requirement, however, linear protocols have the advantage of no privacy leakage as secrets stored are not related between tags. Using the experiment from section 2.1, the only cases where the adversary is able to win is when either T_1 or T_2 is T_0. As such the cases are as follows:

- C$_1$: $(k^0 = k^1) \wedge (k^0 \neq k^2)$ then the attack succeeds,
- C$_2$: $(k^0 \neq k^1) \wedge (k^0 = k^2)$ then the attack succeeds,
- C$_2$: $(k^0 \neq k^1) \wedge (k^0 \neq k^2)$ then the attack definitely fails,

Thus, the probability of winning is:

$$\begin{aligned} \Pr(win) &= \Pr(C_1) \vee \Pr(C_2) \\ &= \frac{2N-2}{N^2} \end{aligned}$$

2.3 Tree-Based Protocols

Tree-based authentication was proposed in 2004 by Molnar et al. [5]. They introduced the concept of a hierarchical arrangement of tags that significantly improves the efficiency of the authentication protocol at the reader. The main motivation behind this approach is to solve the scalability issue associated with linear protocols. This approach has since been considered as one of the most efficient methods of private authentication proposed [2], however it is not without limitations and drawbacks. In the following section will discuss both the advantages and limitations of such approach when compared to traditional approaches that does not use of any explicit tag organizing structure.

The key difference between the two approaches is the method of authentication, whereas tree-based protocols take a top-down approach, i.e walk the tree layer-by-layer from the top as shown in figure 1, other protocols take a horizontal or linear approach, i.e. using the same analogy but with only one layer. Thus, each every tag in the same branch shares the same secret. In the end, it is possible that a tag does not share any unique secret with the database, but instead a unique set of secrets.

As shown in figure 1, authentication is now completed in a number of steps. During each step, the database has to perform an exhaustive within a branch (instead of the entire set of tags). Hence the number of computations required on the reader/database of systems using a tree-based protocol has been reduced to *b(log$_b$ N)*.

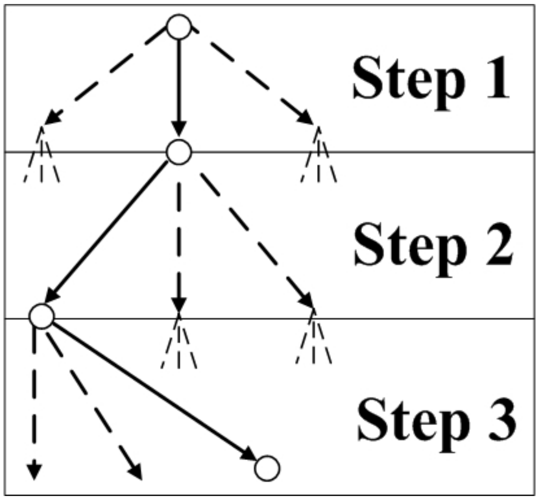

Figure 1: M-ary Tree Walking

2.3.1 CR/MW Protocol

Along with the scheme discussed above, the CR/MW protocol shown in table 2, was proposed. Since tags are arranged in a tree, and that each tag is to store multiple secrets each corresponding to its branch in the tree, the protocol has to be repeated for each layer in the tree. For each layer an exhaustive search of the secrets within the branch is there required. For every protocol run the reader generates a random nonce N_R sent to the tag as a query. The tag in turn generates a random nonce, N_T, which is sent back along with $k_i \oplus f_s(0\|N_R\|N_T)$, where f_s is a pseudo-random function. Finally the reader replies back with $k_i \oplus f_s(1\|N_R\|N_T)$ to finish mutual authentication before the protocol is executed again for the next layer.

When using the CR/MW scheme tags can be authenticated within *b(log$_b$ N)* computations, where N is the total number of tags in the system, and b is the branching factor of the tree. This is a dramatic improvement to the computational efficiency for the reader and/or database, especially in large networks where a large amount of tags are present. Whereas traditionally decrease in operational load on the reader typically implies an increase on the tag, it is not the case in this situation. The number of operations required per iteration of the protocol remains constant.

Reader		**Tag**
		$k_1, k_2, k_3, ...k_*$
	$\xrightarrow{N_R}$	
	$\xleftarrow{k_i \oplus f_s(0 \Vert N_R \Vert N_T), N_T}$	
	$\xrightarrow{k_i \oplus f_s(1 \Vert N_R \Vert N_T)}$	

Table 2: CR/MW Scheme

Whilst there are significant advantages when using a tree-based structure, it is not without drawbacks. Several potential drawbacks of a tree-based system includes the leakage of information, and the number of messages required per authentication session.

One of the most significant drawbacks of using a M-ary tree based approach is the leakage of information when secrets of tags are revealed. Where in linear protocols there exists only one unique secret per tag, in tree-based protocols there are multiple secrets per tag, of which most are shared among other tags. Thus revealing the secrets of one or multiple tags will dramatically reduce the privacy of the system as a whole.

Thus the possible outcomes of the experiment are as follows:[1]:

- $C_i^1 = ((k_i^0 = k_i^1) \wedge (k_i^0 \neq k_i^2))$ then the attack succeeds,
- $C_i^2 = ((k_i^0 \neq k_i^1) \wedge (k_i^0 = k_i^2))$ then the attack succeeds,
- $C_i^3 = ((k_i^0 \neq k_i^1) \wedge (k_i^0 \neq k_i^2))$ then the attack definitely fails,
- $C_i^4 = (k_i^0 = k_i^1 = k_i^2)$ then the attacks fails at level i but can move onto the next level $i + 1$

The probability of an adversary winning the experiment is as below, where δ denotes the branching factor of the tree.

$$\begin{aligned}
Pr(win) &= \Pr(C_1^1 \vee C_1^2) + \sum_{i=2}^{l} \left(\Pr(C_i^1 \vee C_i^2) \times \prod_{j=1}^{i-1} \Pr(C_j^4) \right) \\
&= \frac{2(\delta - 1)}{\delta^2} + \sum_{i=1}^{l} \left(\frac{2(\delta - 1)}{\delta^2} \left(\frac{1}{\delta^2} \right)^{i-1} \right) \\
&= 2(\delta - 1) \frac{1 - \left(\frac{1}{\delta^2} \right)^l}{1 - \frac{1}{\delta^2}} \frac{1}{\delta^2}.
\end{aligned}$$

[2] also showed that the probability of the adversary winning can be lowered significantly if the branching factor, δ, sufficiently large.

2.4 Group Protocols

A proposal to improve the tree-based scheme was proposed by Avoine et al. [6] in the form of group protocols. In group protocols, tags are arranged into different groups instead of a tree. It can also be considered that in this approach, tags are arranged into a two-layer tree. The number of secrets required to be stored by the tag has also been reduced to only 2 a group secret k^1 and a tag secret k^2. Structurally, assuming that all groups are balanced, in a system with N tags, there are to be $N^{\frac{1}{2}}$ groups with $N^{\frac{1}{2}}$ elements(tags) each.

[1] The following cases and probabilities are extracted from [2], for further detail please refer back to the original work.

The group protocol proposed by Avoine et al. is shown in table 3. Notice that instead of pseudo-random functions of the previous two protocols, the group protocol makes use of symmetric encryption, but its functioning is nevertheless similar. For each protocol execution the reader generates a random nonce, N_R, which is sent to the tag. The tag than also generates a nonce, N_R, and replies with message $E_{k_1}(N_R \| N_T), E_{k_2}(N_R \| N_T)$. After receiving the message, the database tries to decrypt $E_{k_1}(N_R \| N_T)$ trying all group secrets, k_1. Once a match is found, the database continues to decrypt $E_{k_2}(N_R \| N_T)$ using all k^2s within the group. Thus the number of computations required by the database is now $2N^{\frac{1}{2}}$.

Reader		**Tag**
k		k
	$\xrightarrow{N_R}$	
	$\xleftarrow{E_{k_1}(N_R \Vert N_T), E_{k_2}(N_R \Vert N_T)}$	

Table 3: Group Protocol

The possible outcomes of the privacy experiment are now as follows:

- $C_1 = ((k_1^0 = k_1^1) \wedge (k_1^0 \neq k_1^2))$ then the attack succeeds,
- $C_2 = ((k_1^0 \neq k_1^1) \wedge (k_1^0 = k_1^2))$ then the attack succeeds,
- $C_3 = ((k_1^0 \neq k_1^1) \wedge (k_1^0 \neq k_1^2)) \wedge ((k_2^0 \neq k_2^1) \wedge (k_2^0 \neq k_2^2)))$ then the attack definitely fails,
- $C_4 = ((k_2^0 = k_2^1) \wedge (k_2^0 \neq k_2^2)))$ then the attack succeeds,
- $C_5 = ((k_2^0 \neq k_2^1) \wedge (k_2^0 = k_2^2)))$ then the attack succeeds

In order to obtain the minimum leakage of privacy, it is assumed that tag secrets,k_2, are unique within the entire system. Thus when there are N tags, the number possible k_1 values are $N^{\frac{1}{2}}$ and the number of possible k_2 values are N. Under the given assumptions, it is possible to obtain the following:

$$\begin{aligned} \Pr(win) &= \Pr(C_1) \vee \Pr(C_2) \vee \Pr(C_4) \vee \Pr(C_5) \\ &= \frac{2N^{\frac{1}{2}} - 2}{N} + \frac{2N - 2}{N^2} \end{aligned}$$

3 Alternate-Tree Walking (ATW)

The privacy of the CR/MW scheme is heavily dependent on the branching factor on the top layer [4], however it is not feasible to indefinitely increase the branching factor. Thus this paper proposes the alternate-tree walking scheme circumvent this problem. The resulting scheme significantly reduces the amount of leakage compared to traditional tree-based(and group) protocols whilst maintaining a reasonable amount of computational load on the database. The core concept of alternate-tree walking is to start authentication from a layer in-between the top and bottom layers of a tree, as opposed to authenticating sequential from the top to bottom. Although the concept can be applied to any tree with more than three layers, for simplicity, the rest of the paper will consider a tree with only three layers.

Before further detailing the approach, we outline the structural differences between an M-ary tree and a 3-layer tree. Thus each tag only has to store 3 secrets, K_1, K_2, K_3. Whereas in a balanced 3-layer tree the branching factor is always $N^{\frac{1}{3}}$, where N is the total number of tags in the system, an M-ary-tree does not set any limits on its branching factor and consequently the number of layers, making direct comparisons difficult. Therefore the remainder of this section will compare the approaches under a 3-layer tree structure. It should be emphasized that this is to the advantage of the traditional tree based approach due to the fact that this is when its branching factor is maximized, thus leakage is also minimal.

3.1 ATW in a 3-layer tree

As shown in figure 2, authentication is completed in 3 steps starting from the middle layer. Whereas traditional tree-based approach authenticate sequentially down the tree, this approach starts from the middle back to the top before working down the tree. In essence, this approach is to achieve benefits of a large branching factor without altering the structure of the tree. In a tree structure it is required that all secrets be unique within a branch, thus in traditional tree-based approach the number of possible unique secrets in the initial branch is limited to its branching factor. However, by starting authentication from a middle layer, the number of possible unique secret values has increased to $N^{\frac{2}{3}}$ from $N^{\frac{1}{3}}$ of the traditional approach, and $N^{\frac{1}{2}}$ of group protocols.

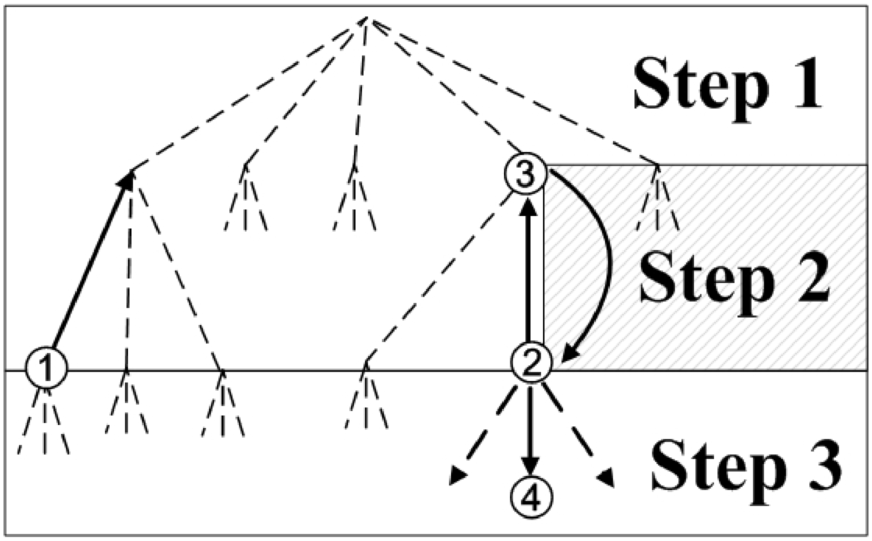

Figure 2: Alternate Tree-Walking

Also suggested in figure 2 is the possibility of shared layer-two secrets. Since benefit of reduced information leakage comes at the expense of computations required, by allowing secrets to be shared between branches it is possible to establish a trade-off between computations required and information leaked. This trade-off will be further explored further in the next section.

3.2 Comparison

In this section the proposed approach and previous proposals will be compared using the experiment from section 2.1. For the rest of the section all tree-based approaches are compared using the same 3-layer tree, thus when there are N tags in the system the branching factor will be $N^{\frac{1}{3}}$. It is also assumed that in this section the ATW scheme also uses the CR/MW protocol.
In the ATW scheme, the possible outcomes of the experiment are as follows:

- $C_1 = ((k_2^0 = k_2^1) \wedge (k_2^0 \neq k_2^2))$ then the attack succeeds,
- $C_2 = ((k_2^0 \neq k_2^1) \wedge (k_2^0 = k_2^2))$ then the attack succeeds,
- $C_3 = ((k_2^0 \neq k_2^1) \wedge (k_2^0 \neq k_2^2))$ then the attack definitely fails,
- $C_4 = ((k_2^0 = k_2^1 = k_2^2)$ then following cases are possible,
  - $C_{4.1} = ((k_1^0 = k_1^1) \wedge (k_1^0 \neq k_1^2))$ then the attack succeeds,
  - $C_{4.2} = ((k_1^0 \neq k_1^1) \wedge (k_1^0 = k_1^2))$ then the attack succeeds,
  - $C_{4.3} = ((k_1^0 \neq k_1^1) \wedge (k_1^0 \neq k_1^2))$ then the attack definitely fails,

- $C_{4.4} = ((k_1^0 = k_1^1 = k_1^2)$ then following cases are possible,
  - $C_{4.4.1} = ((k_3^0 = k_3^1) \wedge (k_3^0 \neq k_3^2))$ then the attack succeeds,
  - $C_{4.4.2} = ((k_3^0 \neq k_3^1) \wedge (k_3^0 = k_3^2))$ then the attack succeeds,
  - $C_{4.4.3} = ((k_3^0 \neq k_1^1) \wedge (k_1^0 \neq k_3^2))$ then the attack definitely fails.

From the above, it is possible to obtain the following probability:

$$\begin{aligned}
Pr(win) &= Pr(C_1) \vee Pr(C_2) \vee Pr(C_4 \wedge (Pr(C_{4.1}) \vee Pr(C_{4.2}) \vee Pr(C_{4.4} \wedge (Pr(C_{4.4.1}) \vee Pr(C_{4.4.2}))) \\
&= \frac{2}{N^{\frac{2}{3}}}(\frac{N^{\frac{2}{3}}-1}{N^{\frac{2}{3}}}) + \frac{1}{N^{\frac{4}{3}}}(\frac{2}{N^{\frac{1}{3}}}(\frac{N^{\frac{1}{3}}-1}{N^{\frac{1}{3}}}) + \frac{1}{N^{\frac{2}{3}}}(\frac{2}{N^{\frac{1}{3}}}(\frac{N^{\frac{1}{3}}-1}{N^{\frac{1}{3}}}))) \\
&= \frac{2N^{\frac{2}{3}}-2}{N^{\frac{4}{3}}} + \frac{1}{N^{\frac{4}{3}}}(\frac{2N^{\frac{1}{3}}-2}{N^{\frac{2}{3}}} + \frac{2N^{\frac{1}{3}}-2}{N^{\frac{4}{3}}})
\end{aligned}$$

Similarly, the probability of the traditional approach in a 3-layer tree is given by the probability below:

$$= \frac{2N^{\frac{1}{3}}-2}{N^{\frac{2}{3}}} + \frac{2N^{\frac{1}{3}}-2}{N^{\frac{4}{3}}} + \frac{2N^{\frac{1}{3}}-2}{N^{\frac{6}{3}}}$$

Also included in the comparison is group and linear protocols. It should be noted in consistency with previous sections, only group protocols is assumed to have N k_2 values, whereas both tree schemes are assumed to have $N^{\frac{1}{3}}$ k_1, k_2, and k_3 values.

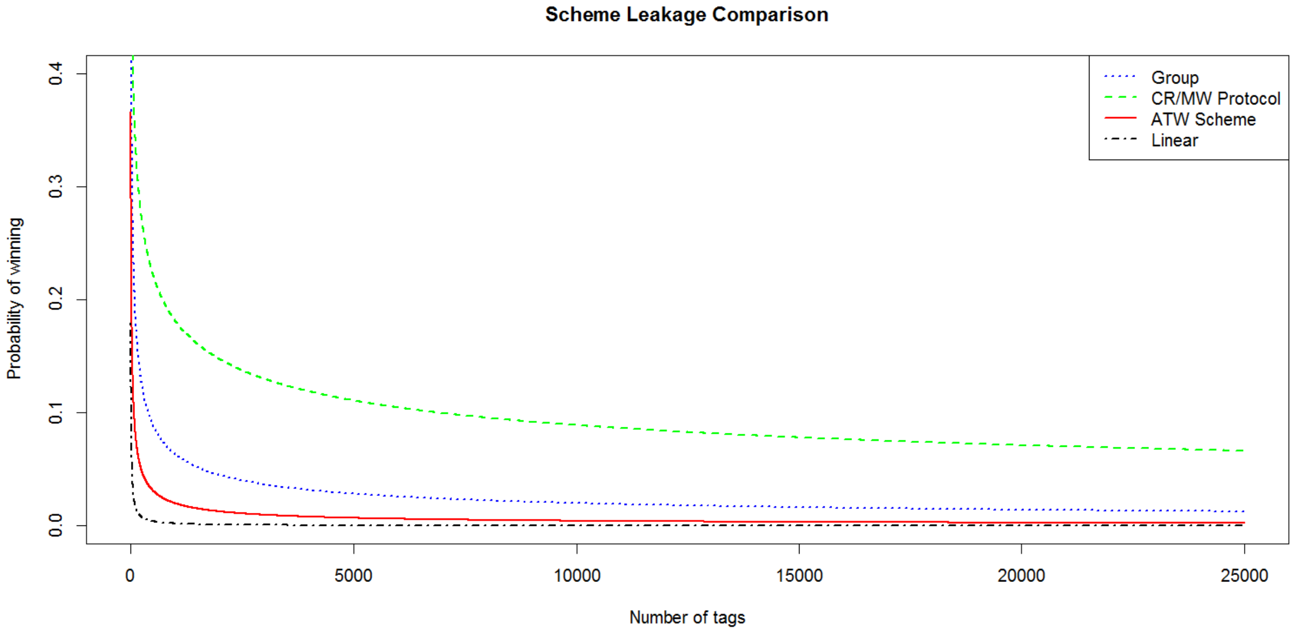

Figure 3: Leakage Comparison between Schemes

The comparison of the adversary winning are shown in figure 3 and table 4. Evidently, the leakage of ATW is notably lower compared to previous proposals. Particularly the level of leakage is approaching that of linear protocols, as shown in figure 3.

No# Tags	**CR/MW**	**Group**	**Linear**	**ATW**
100	35%	20%	2%	9%
500	22%	8.9%	0.4%	3%
1000	18%	6.3%	0.2%	2%
5000	11%	2.8%	0.04%	0.7%

Table 4: Scheme Leakage Comparison

4 Limitations of Privacy Leakage Measurement

The experiment from [2], detailed in section 2.1, does not take into consideration the ability for an adversary to view transcripts of past successful sessions, a potential major increase of leakage of information. In a library scenario, transcripts can be obtained by an adversary eavesdropping communication between a reader (typically close to doors) and tags (attached to books) as they are carried out. Having access to transcripts significantly increases the probability of the adversary winning in the CR/MW scheme as the adversary is able to compare all secrets as the same time, as opposed to given the next iteration of the protocol only when $(k_i^0 = k_i^1 = k_i^2)$. Thus this section proposes an extension to the experiment that also considers the adversaries ability to view successful protocol transcripts. The experiments would subsequently be compared.

In the new experiment, only step 2 and 3 need to be changed. The experiment is now as follows:

1. The adversary draws one tag, T_0, and obtains it's full set of secrets, K_1^0, K_2^0, K_3^0, etc... The tag is put back into circulation.

2. The adversary is than randomly given a tag T_j and is allowed to query the tag and request protocol past successful transcripts as much as it wants. However, the adversary is not allowed to reveal the secrets of the tags.

3. The adversary is now given two tags, T_1 and T_2 such that $T_j \in \{T_1, T_2\}$. and is allowed to query both tags and request protocol past successful transcripts as much as it wants. The adversary wins the experiment if it can output i, such that $T_i = T_j$.

4.1 Comparison of experiment

In this section, the proposed extension of the experiment will be compared with the original using the CR/MW scheme. Using the same 3-layer tree from previous sections, the possible outcomes of the extended experiment in the CR/MW scheme would became as follows:

- $C_1 = ((k_1^0 = k_1^1) \wedge (k_1^0 \neq k_1^2))$ then the attack succeeds,

- $C_2 = ((k_1^0 \neq k_1^1) \wedge (k_1^0 = k_1^2))$ then the attack succeeds,

- $C_3 = ((k_2^0 = k_2^1) \wedge (k_2^0 \neq k_2^2)))$ then the attack succeeds,

- $C_4 = ((k_2^0 \neq k_2^1) \wedge (k_2^0 = k_2^2)))$ then the attack succeeds

- $C_5 = ((k_3^0 = k_3^1) \wedge (k_2^3 \neq k_3^2)))$ then the attack succeeds,

- $C_6 = ((k_3^0 \neq k_3^1) \wedge (k_3^0 = k_3^2)))$ then the attack succeeds,

- $C_7 = ((k_1^0 \neq k_1^1) \wedge (k_1^0 \neq k_1^2)) \wedge ((k_2^0 \neq k_2^1) \wedge (k_2^0 \neq k_2^2))) \wedge ((k_3^0 \neq k_3^1) \wedge (k_3^0 \neq k_3^2)))$ then the attack definitely fails.

Thus:

$$\Pr(win) \quad = \quad \Pr(C_1) \vee \Pr(C_2) \vee \Pr(C_3) \vee \Pr(C_4) \vee \Pr(C_5) \vee \Pr(C_6)$$

For a more meaningful comparison, two cases of the CR/MW scheme will be considered in the new experiment, its best case and worst case scenario. In its best case scenario, all k^3 and k^2 secrets are assumed to be unique with its layer. Thus there are N possible values of k^3 and $N^{\frac{2}{3}}$ possible values of k^2. In its worst case scenario, however, all secrets are assumed to be unique only within its branch. As such there are $N^{\frac{1}{3}}$ possible values of k^3 and $N^{\frac{1}{3}}$ possible values of k^2. The possibilities are as follows:

$$\begin{aligned} Pr(win - best) &= \frac{2N^{\frac{1}{3}} - 2}{N^{\frac{2}{3}}} + \frac{2N^{\frac{2}{3}} - 2}{N^{\frac{4}{3}}} + \frac{2N - 2}{N^2} \\ Pr(win - worst) &= \frac{6N^{\frac{1}{3}} - 6}{N^{\frac{2}{3}}} \end{aligned}$$

No# Tags	OldExp	NewExp-Best	NewExp-Worst
100	35%	45%	99%
500	22%	26%	66%
1000	18%	20%	54%
5000	11%	12%	33%

Table 5: Comparison of Experiments

The results of the comparison are presented in table 5 and figure 4. It should be emphasized that in the old experiment, the values presented uses the same tree-structure as the worst case scenario in the extended experiment. In the table it is shown throughout the experiments under the same conditions, probability of the adversary winning when given successful protocol transcripts increases by three-fold. Interestingly, it can be observed that, even in the best case scenario, the probability of the adversary with transcripts winning is still higher than worst case of the one without. Nevertheless, it is apparent that this additional power of the adversary does not have any effect on group and linear protocols.

On the side note, by giving the adversary transcripts the ATW scheme does not gain any privacy advantage over the CR/MW scheme. For the rest of the paper the best case scenario of from the extended experiment will be used as a baseline for the CR/MW scheme.

4.2 The ATW Protocol

As one may realize, using the ATW scheme with the CR/MW protocol does not gain any privacy advantage over using the CR/MW scheme. Simply by analyzing protocol transcript it is possible for an adversary to gain the same amount of information as the original approach. Thus the remainder of this section will be used to propose protocols that designed to take advantage of the reduced information leakage of the ATW scheme.

Reader		**Tag**
		k_1, k_2, k_3
	$\xrightarrow{N_R}$	
	$\xleftarrow{H(N_T \Vert N_R \Vert k_2) \Vert H(N_R \Vert k_1) \oplus H(N_T \Vert k_3), N_T}$	

Table 6: Alternate-Tree Walking Protocol (ATW-Protocol)

Table 6 shows the ATW protocol designed to take advantage of the alternate-tree walking scheme. Evidently, transcripts of this protocol does not leak any more information than required. In the protocol,

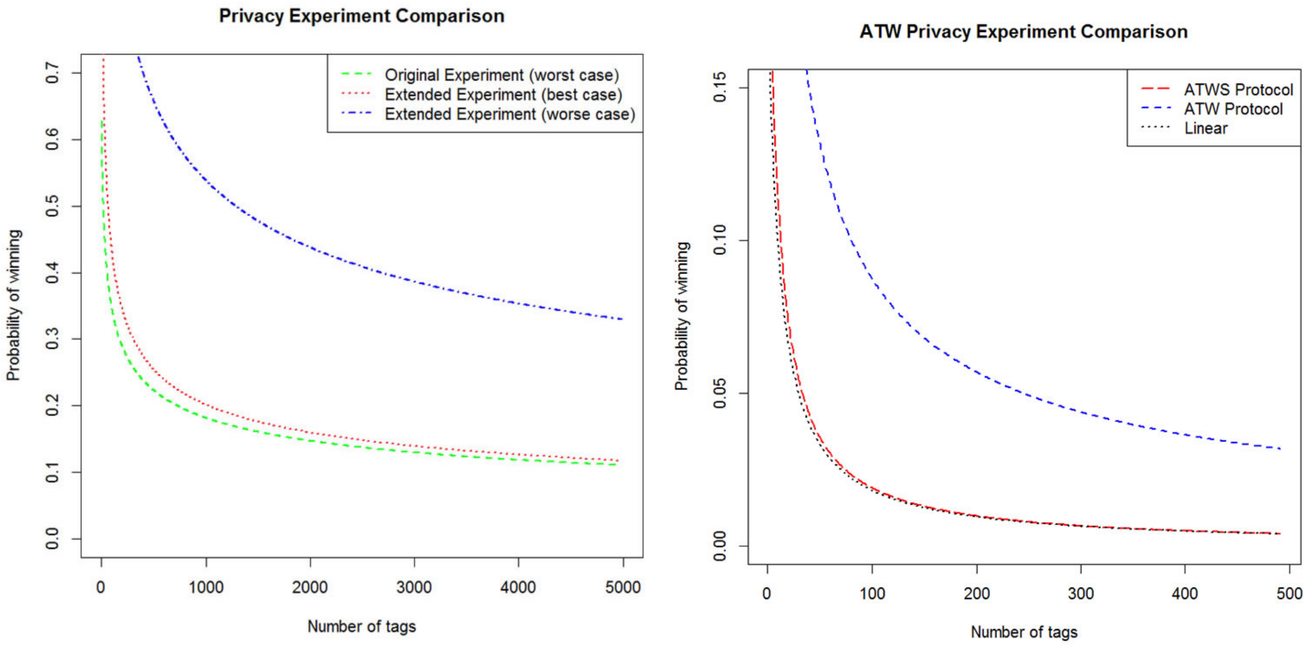

Figure 4: Comparison of Experiments

the database first verifies the message by computing $H(N_T \| N_R \| k_2)$ for all k_2 after which checks for K_1 where the value of k_2 exists. The database than only computes the values of $H(N_R \| k_1) \oplus H(N_T \| k_3$ for which $k_2 \in k_1$ and $k_3 \in k_2$.

Due to the use of $H(N_R \| k_1) \oplus H(N_T \| k_3$ to authenticate layer 1 and 2, it should be noted that the leakage of information using this protocol is less than that given in the previous section. If $D = H(N_R \| k_1) \oplus H(N_T \| k_3)$, the possible outcomes of winning are:

- C$_1$: $(k_1^0 = k_1^1) \wedge (k_1^0 \neq k_1^2)$
- C$_2$: $(k_1^0 \neq k_1^1) \wedge (k_1^0 = k_1^2)$
- C$_3$: $(D_0 = D_1) \wedge (D_0 \neq D_2)$
- C$_4$: $(D_0 \neq D_1) \wedge (D_0 = D_2)$

For consistency the best case tree-structure, where there are N possible values of k^3 and $N^{\frac{2}{3}}$ possible values of k^2, will be used. The overall probability of winning:

$$\begin{aligned} Pr(win) &= Pr(C_1) \vee Pr(C_1) \vee Pr(C_2) \vee Pr(C_3) \vee Pr(C_4) \\ &= 2(\frac{1}{N^{\frac{2}{3}}})(\frac{N^{\frac{2}{3}}-1}{N^{\frac{2}{3}}}) + 2(\frac{1}{N^{\frac{4}{3}}})(\frac{N^{\frac{4}{3}}-1}{N^{\frac{4}{3}}}) \\ &= \frac{2N^{\frac{2}{3}}-2}{N^{\frac{4}{3}}} + \frac{2N^{\frac{4}{3}}-2}{N^{\frac{8}{3}}} \end{aligned}$$

The resulting leakages are shown in table 7, however, as the plot of the overall results resembles that of figure3, it will not be repeated. The results are in particular interesting as the leakage of the ATW protocol has been reduced by more than 50% than when the ATW scheme was compared using the CR/MW protocol. Nevertheless, the protocol still leakes twice as much information as that of protocols.

No# Tags	CR/MW Protocol	Group	Linear	ATW Protocol
100	9%	20%	2%	4%
500	3%	8.9%	0.4%	0.08%
1000	2%	6.3%	0.2%	0.04%
5000	0.7%	2.8%	0.04%	0.08%

Table 7: Protocol Leakage Comparison

4.2.1 Further Reduction of Privacy Leakage for Small Networks

This section proposes a modification of the protocol proposed in the previous section that aims to minimize the leakage of information for small networks. The protocol is shown in table 8.

Reader		**Tag**
		k_1, k_2, k_3
	$\xrightarrow{N_R}$	
	$\xleftarrow{H(N_R\|S_2)\oplus H(N_T\|S_1)\|H(N_T\|S_2)\oplus H(N_R\|S_3)),N_T}$	

Table 8: Alternate-Tree Walking Protocol for Small Networks (ATWS-Protocol)

By requiring the database to compute $H(N_R\|S_1)\oplus H(N_T\|S_2)$ first, followed by $H(N_R\|S_2)\oplus H(N_T\|S_3)$ the probability of the adversary winning has decreased to:

$$Pr(win) \quad = \quad \frac{4N-4}{N^2}$$

Nevertheless, the number of required computations have increased to $N^{\frac{2}{3}} + 2N^{\frac{1}{3}}$. However even though this scheme is aimed at small networks, the increased computational requirement is comparatively minimal compared to linear protocols. A comparasion is shown in the left plot in figure 5, it can be seen that also the the number of computations reqiured is increased slightly, the increase decrease in privacy is significant as shown in the right plot of figure 4. More detailed results are shown in table 9, evidently the ATWS provides a comparable leverl of privacy to linear protocols at significantly less computational cost.

No# Tags	ATW	Linear	ATWS
100	9.3%	2%	2.1%
200	5.8%	1%	1%
300	4.5%	0.7%	0.7%
400	3.7%	0.5%	4%

Table 9: ATW and ATWS Protocol Results Comparison

4.2.2 Further Reduction of Computation using Pre-Computation

This section discuss the use of pre-computation to reduce the amount of computations required by the database. By pre-computing the values of $H(N_R\|k^*)$, it is possible to decrease the amount of time and computation required during authentication. Although the method can be applied to both proposed schemes, it would be most useful when applied to the ATWS-Protocol. By pre-computing the values of $H(N_R\|S_2)$ and $H(N_R\|S_3)$ in $H(N_R\|S_2) \oplus H(N_T\|S_1)\|H(N_T\|S_2) \oplus H(N_R\|S_3))$, it is possible to reduce the number of computations during authentication from $N^{\frac{2}{3}} + 2N^{\frac{1}{3}}$ to $N^{\frac{1}{3}}$, allowing the protocol to complete authentication with less computations then the CR/MW scheme. A comparasion is shown in the left plot in figure 5, it can be seen that after pre-compuation the Precomputated-ATWS(P-ATWS) protocol requires the least number of computations.

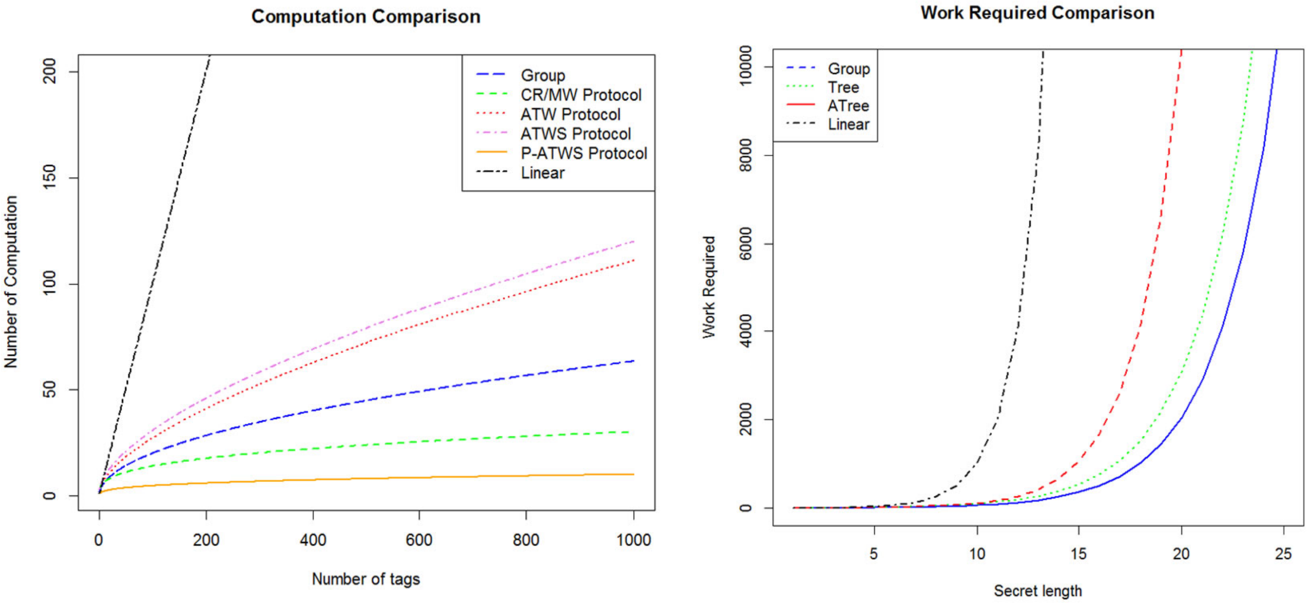

Figure 5: Comparison of Computations Required between Protocols

4.3 Other Considerations

An interesting, but seldom discussed, aspect of protocols is the amount of work required by the adversary to attack the scheme. Namely the amount of work required by the adversary to brute-force the secrets from protocol messages. This aspect is comparatively more important for RFID protocols due to their limited storage, making brute-force attacks seemingly more attractive than for traditional key-exchange protocols. Assuming that the total amount of memory given to a tag for storing secrets is K bits, the amount of work required is shown in table 10, and in the right graph of figure 5. It can be observed that linear protocols require the most amount of work to attack followed by the ATW-Scheme. It should be noted that both protocols based on the ATW-Scheme require the same amount of work to attack.

	Tree	**Group**	**Linear**	**ATree**
Work Required	$3(2^{\frac{K}{3}})$	$2(2^{\frac{K}{2}})$	2^K	$2^{\frac{K}{3}}+2^{\frac{2K}{3}}$

Table 10: Work Comparison

5 Conclusion

This paper analyzed the leakage of information in linear, tree-based and group-based RFID protocols as well addresses a limitation of a current privacy measurement method. The paper also proposed two protocols, the ATW protocol and ATWS protocol, which were showed to leak substantially less privacy compared to analyzed protocols. The increased computational requirement of the proposed protocols can also be offloaded though the use of pre-computation, the resulting protocol can be completed using less computations than that of tree-based protocols. The resulting P-ATWS protocol was able to match linear protocols in terms of privacy but at the same time only require one-third of what is required of tree-based protocols.

References

[1] X. Huang, "Quantifying Information Leakage in RFID Systems," in *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, vol. 1, Feburary 2008, pp. 84–89.

[2] G. Avoine, E. Dysli, and P. Oechslin, "Reducing Time Complexity in RFID Systems," in *Selected Areas in Cryptography*, ser. Lecture Notes in Computer Science, B. Preneel and S. Tavares, Eds., vol. 3897. Springer Berlin / Heidelberg, 2006, pp. 291–306.

[3] K. Nohl and D. Evans, "Quantifying Information Leakage in Tree-Based Hash Protocols," in *Information and Communications Security*, ser. Lecture Notes in Computer Science, P. Ning, S. Qing, and N. Li, Eds. Springer Berlin / Heidelberg, 2006, vol. 4307, pp. 228–237.

[4] L. Buttyn, T. Holczer, and I. Vajda, "Optimal key-trees for tree-based private authentication," in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, G. Danezis and P. Golle, Eds. Springer Berlin / Heidelberg, 2006, vol. 4258, pp. 332–350.

[5] D. Molnar and D. Wagner, "Privacy and Security in Library RFID: Issues, Practices, and Architectures," in *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2004, pp. 210–219.

[6] G. Avoine, L. Buttyán, T. Holczer, and I. Vajda, "Group-based private authentication," in *IEEE International Workshop on Trust, Security, and Privacy for Ubiquitous Computing – TSPUC*, IEEE. Helsinki, Finland: IEEE Computer Society, June 2007, pp. 1–6.