Group-Based Private Authentication

Gildas Avoine* avoine@mit.edu

Levente Buttyán† buttyan@crysys.hu

Tamás Holczer† holczer@crysys.hu

István Vajda† vajda@crysys.hu

Abstract

We propose a novel authentication scheme that ensures privacy of the provers. Our scheme is based on symmetric-key cryptography, and therefore, it is well-suited to resource constrained applications in large scale environments. A typical example for such an application is an RFID system, where the provers are low-cost RFID tags, and the number of the tags can potentially be very large. We analyze the proposed scheme and show that it is superior to the well-known key-tree based approach for private authentication both in terms of privacy and efficiency.

1 Introduction

The problem of private authentication is to enable the authentication of a party, called the prover, to another party, called the verifier, in such a way that an adversary can neither identify nor track the prover. We consider the private authentication problem in a resource constrained application where only symmetric-key cryptography is feasible and tamper resistance is limited. In addition, we assume that there are many potential provers. A typical example for such an application is an RFID system, where the provers are low-cost RFID tags and the verifier is a back-end system that interacts with the tags via reader devices. Hence, in the rest of the paper, we use the terms *tag* and *reader* instead of the terms *prover* and *verifier*, respectively. However, we emphasize that our work is not restricted to RFID systems, but our results can equally be used in applications with similar characteristics (e.g., in wireless sensor networks).

The problem of private authentication in the setting described above is that although a tag can encrypt its messages to hide its identity from eavesdroppers, it cannot give any hint to the reader regarding the key that it used for encryption, because such a hint could also be used by the adversary to break the tag's privacy. For this reason, the reader must search through a set of candidate keys until it finds the right key that properly decrypts the tag's messages.

Some private authentication schemes require the reader to test $O(N)$ keys to authenticate a tag, where N is the total number of tags in the system. Such a complexity is unmanageable in a large scale environment, such as an automated fare collection system for public transportation, where thousands of electronic tickets must be checked every second. Molnar and Wagner proposed an approach in [5] that reduces the complexity of authentication from $O(N)$ to $O(\log N)$. This reduction is made possible by using a key-tree instead of a flat key space. In the Molnar-Wagner scheme, the tags are assigned to the leaves of a balanced tree with branching factor b at each level of the tree. In addition, each edge of the tree is associated with a unique key. Each tag stores the keys along the path from the root to the leaf corresponding to the given tag (see Figure 2.). The reader possesses all keys in the tree (or it can generate them systematically from a single master key). When authenticating itself, a tag uses all of its keys. The reader identifies which keys have been used by iteratively searching through the keys at the successive levels of the tree. Note, however, that at each level, the reader only needs to consider the subtree that belongs to the path consisting of the already identified keys at the upper layers. In the worst case, the reader needs to test b keys at each level, hence the complexity of the authentication for the reader is $b \log_b N$. For the tag, the complexity is $\log_b N$ in terms of computation, communication, and storage.

However, while the key-tree based private authentication scheme of Molnar and Wagner reduces the complexity of the authentication for the reader, there is a price to be paid for this gain in performance: The level of privacy provided by the scheme is quickly decreasing as more and more tags are compromised (meaning that their keys become known to the adversary). The simple reason for this is that tags that belong to the same subtree of the key-tree share some common keys. Therefore, if one tag is compromised, then this affects the privacy of all other tags that have some common keys with the compromised tag. We note that it is reasonable to assume that some tags become compromised, because tags cannot be assumed to be tamper resistant.

*Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT)

†Laboratory of Cryptography and Systems Security (CrySyS), Budapest University of Technology and Economics (BME), Hungary

Intuitively, key-trees with small branching factors are better in terms of complexity but worse in terms of privacy because keys are shared by more tags. Hence, there is a trade-off between the complexity and the level of privacy provided by the key-tree based scheme. This trade-off is identified and analyzed by Avoine, Dysli, and Oechslin in [1], by Buttyan, Holczer, and Vajda in [2], and more recently by Nohl and Evans in [6]. In particular, these papers introduce privacy metrics and quantify the level of privacy provided by the key-tree based scheme when some tags are compromised. In addition, in [2], the authors observe that key-trees that have different branching factors at different levels of the tree can provide a higher level of privacy, and they propose an algorithm to determine the optimal key-tree for a given number of tags and a given upper bound on the complexity of the authentication. We will rely on these results in this paper.

Although, as we have seen, the key-tree based scheme proposed by Molnar and Wagner is not perfectly privacy-compliant, it has been held in great consideration by the RFID community, because it is the only private authentication protocol so far that could be deployed in large scale in practice. Hence, improving the trade-off between privacy and complexity provided by the key-tree based approach has a great practical importance. This observation serves as the main motivation for our work.

Our contribution in this paper is a novel symmetric-key private authentication scheme that provides a higher level of privacy and achieves better efficiency than the key-tree based approach. More precisely, the complexity of our scheme for the reader can be set to be $O(\log N)$ (i.e., the same as in the key-tree based approach), while the complexity for the tags is always a constant (in contrast to $O(\log N)$ of the key-tree based approach). Hence, our scheme is better than the key-tree based scheme both in terms of privacy and efficiency, and therefore, it is a serious alternative to the key-tree based scheme to be considered by the RFID community.

The organization of the paper is the following: We describe the operation of our scheme in Section 2, and we quantify the level of privacy that it provides in Section 3. We compare our proposed scheme to the key-tree based approach in Section 4, and finally, we conclude the paper in Section 5.

2 The group-based approach

In our proposed private authentication scheme, the set of all tags is divided into groups of equal size, and all tags of a given group share a common group key. Since the group keys do not enable the reader to identify the tags uniquely, every tag also stores a unique identifier. Keys are secret (each group key is known only to the reader and the members of the corresponding group), but identifiers can be public. To avoid impersonation of a tag from the same group, every tag has a unique secret key as well. This key is only shared between the tag and the reader. To reduce the storage demands on the reader side, the pairwise key can be generated from a master key using the identifier of the tag.

In order to authenticate a tag, the reader sends a single challenge to the tag. The answer of the tag has two parts. In the first part, the tag answers to the reader by encrypting with the group key the reader's challenge concatenated with a nonce picked by the tag, and the tag's identifier. In the second part, the tag encrypts the challenge concatenated with the nonce using its own secret key. Encrypting the identifier is needed since the key used for encryption does not identify uniquely the tag. Upon reception of the answer, the reader identifies the tag by trying all the group keys until the decryption succeeds. Then it checks the second part, that it was encrypted by the same tag. Without the second part, every tag could impersonate every other tag in the same group.

The operation of our group-based private authentication scheme is illustrated in Figure 1.

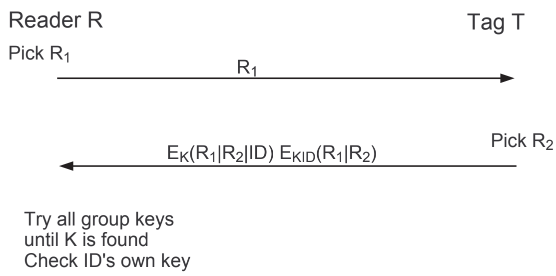

Figure 1. Operation of the group-based private authentication scheme. K is the group key stored by the tag, KID is the tag's own secret key, ID is the identifier of the tag, R_1 and R_2 are random values generated by the reader and the tag, respectively, $|$ denotes concatenation, and $E_K()$ denotes symmetric-key encryption with K.

The complexity of the group-based scheme for the reader depends on the number of the groups. In particular, if there are γ groups, then, in the worst case, the reader must try γ keys. Therefore, if the upper bound on the worst case complexity is given as a design parameter, then γ is easily determined. For example, to get the same complexity as in the key-tree based scheme proposed by Molnar and Wagner, one may choose $\gamma = (b \log_b N) - 1$, where N is the total number of tags and b is the branching factor of the key-tree. The minus one indicates the decryption of the second part of the message.

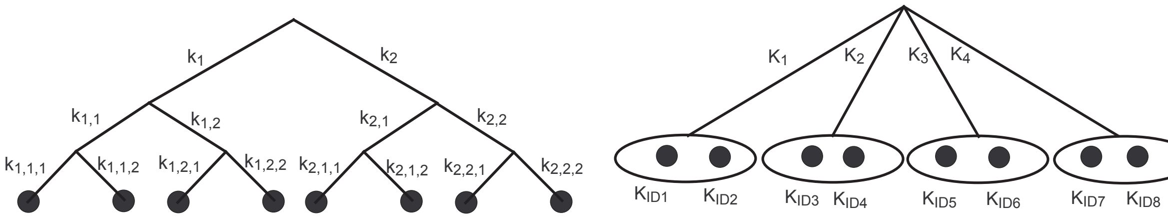

Figure 2. On the left hand side: The tree-based authentication protocol uses a tree, where the tags correspond to the leaves of the tree. Each tag stores the keys along the path from the root to the leaf corresponding to the given tag. When authenticating itself, a tag uses all of its keys. The reader identifies which keys have been used by iteratively searching through the keys at the successive levels of the tree. On the right hand side: In the group-based authentication protocol, the tags are divided into groups. Each tag stores its group key and its own key. When authenticating itself, a tag uses its group key first, and then its own key. The reader identifies which group key has been used by trying all group keys, then it checks the tags own key.

An immediate advantage of the group-based scheme with respect to the key-tree based approach is that the tags need to store only two keys and an identifier. In contrast to this, in the key-tree based scheme, the number of keys stored by the tags depends on the depth of the tree. For instance, in the case of the Molnar-Wagner scheme, the tags must store $\log_b N$ keys. Moreover, by using only two keys, our scheme also has a smaller complexity for the tag in terms of computation and communication.

Besides its advantages with respect to complexity, the group-based scheme provides a higher level of privacy than the key-tree based scheme when some of the tags are compromised. We will show this in Section 4.

3 Analysis

In this section, we want to characterize the level of privacy provided by our group-based scheme as a function of the number of the compromised tags. For this, we need a privacy metric [3, 4, 7]. In this paper, we use the metric proposed by Buttyan, Holczer, and Vajda in [2] for key-tree based private authentication schemes.

The metric proposed in [2] is based on the observation that when some tags are compromised, the set of all tags become partitioned such that the adversary cannot distinguish the tags that belong to the same partition, but she can distinguish the tags that belong to different partitions. Hence, the partitions are the anonymity sets of their members. The level R of privacy provided by the scheme is then characterized as the average anonymity set size normalized with the total number N of the tags. Formally,

$$R = \frac{1}{N}\sum_i |P_i|\frac{|P_i|}{N} = \frac{1}{N^2}\sum_i |P_i|^2 \qquad (1)$$

where $|P_i|$ denotes the size of partition P_i and $|P_i|/N$ is the probability that a randomly chosen tag belongs to partition P_i.

In our group-based scheme, a similar kind of partitioning can be observed when tags become compromised. In particular, when a single tag is compromised, the adversary learns the group key of that tag, which allows her to distinguish the tags within this group from each other (since the tags use their identifiers in the protocol) and from the rest of the tags in the system. This means that each member of the compromised group forms an anonymity set of size 1, and the remaining tags form another anonymity set. In general, when more tags are compromised, we observe that the partitioning depends on the number C of the compromised groups, where a group is compromised if at least one tag that belongs to that group is compromised. More precisely, when C groups are compromised, we get nC anonymity sets of size 1 and an anonymity set of size $n(\gamma - C)$, where γ is the number of groups and $n = N/\gamma$ is the size of a group. This results in the following expression for the level R of the privacy according to the metric (1):

$$R = \frac{1}{N^2}\left(nC + (n(\gamma - C))^2\right) \qquad (2)$$

If tags are compromised randomly, then C, and hence, R are random variables, and the level of privacy provided by the system is characterized by the expected value of R. In order to compute that, we must compute the expected value of C and that of C^2. This can be done as follows: Let us denote by A_i the event that at least one tag from the i-th group is compromised, and let I_{A_i} be A_i's indicator function. The probability of A_i can be calculated as follows:

$$P(A_i) = 1 - \frac{\begin{pmatrix} N-n \\ c \end{pmatrix}}{\begin{pmatrix} N \\ c \end{pmatrix}} = \tag{3}$$

$$= 1 - \prod_{j=0}^{c-1} \left(1 - \frac{n}{N-j}\right) \tag{4}$$

The expected value of C is the expected value of the sum of the indicator functions:

$$E\left[C\right] = E\left[\sum_{i=1}^{\gamma} I_{A_i}\right] = \sum_{i=1}^{\gamma} P(A_i) = \tag{5}$$

$$= \gamma \left(1 - \prod_{j=0}^{c-1} \left(1 - \frac{n}{N-j}\right)\right) \tag{6}$$

Similarly, the second moment of C can be computed as follows:

$$E\left[C^2\right] = E\left[\sum_{i=1}^{\gamma} I_{A_i}\right]^2 = \tag{7}$$

$$= E\left[\sum_{i=1}^{\gamma} I_{A_i}\right] + E\left[\sum_{i \neq j} I_{A_i \cap A_j}\right] = \tag{8}$$

$$= E\left[C\right] + \left(\gamma^2 - \gamma\right) P\left(A_i \cap A_j\right) \tag{9}$$

Finally, probability $P\left(A_i \cap A_j\right)$ can be computed in the following way:

$$P\left(A_i \cap A_j\right) = \tag{10}$$

$$= 1 - P\left(\overline{A_i} \cap \overline{A_j}\right) - 2P\left(A_i \cap \overline{A_j}\right) \tag{11}$$

$$P\left(\overline{A_i} \cap \overline{A_j}\right) = \frac{\begin{pmatrix} N-2n \\ c \end{pmatrix}}{\begin{pmatrix} N \\ c \end{pmatrix}} = \tag{12}$$

$$= \prod_{j=0}^{c-1} \left(1 - \frac{2n}{N-j}\right) \tag{13}$$

$$P\left(A_i \cap \overline{A_j}\right) = P\left(A_i | \overline{A_j}\right) P\left(\overline{A_j}\right) = \tag{14}$$

$$= \left[1 - \prod_{j=0}^{c-1} \left(1 - \frac{n}{N-n-j}\right)\right] \cdot \tag{15}$$

$$\cdot \prod_{j=0}^{c-1} \left(1 - \frac{n}{N-j}\right) \tag{16}$$

Based on the above formulae, we computed the expected value of R as a function of c for $N = 2^{14}$ and $\gamma = 64$. The results are plotted on the left hand side of Figure 3. The same plot also contains the results of a Matlab simulation with the same parameters, where we chose the c compromised tags uniformly at random. For each value of c, we ran 10 simulations, computed the exact values of the average anonymity set size using (1) directly, and averaged the results. As it can be seen in the figure, the analytical results match the results of the simulation. We performed the same verification for several other values of N and γ, and in each case, we obtained the same matching results.

4 Comparison with the key-tree based approach

In this section, we compare our group-based scheme to the key-tree based scheme. Our methodology is the following: For a given number N of tags and upper bound γ on the worst case complexity for the reader, we determine the optimal key-tree using the algorithm proposed in [2]. Then, we compare the level of privacy provided by this optimal key-tree to that provided by our group-based scheme with γ groups and N tags.

The comparison is performed by means of simulations. A simulation run consists in randomly choosing c compromised tags, and computing the resulting normalized average anonymity set size R for both the optimal key-tree and the group-based scheme. For the former, we use the formulae given in [2], while for the latter, we use formula (2) directly. For each value of c, we run several simulation runs, and average the results.

The simulation parameters were the following: For the number N of tags, we considered only powers of 2, because in practice, that number is related to the size of the identifier space, and identifiers are usually represented as binary strings. Thus, in the simulations, $N = 2^x$, and we varied x between 10 and 15 with a step size of 1. The values for the worst case complexity γ (which coincides with the number of groups in the group-based scheme) were 64, 128, and 256. Finally, we varied the number c of compromised tags from 1 to 3γ. For each combination of these values, we ran 100 simulation runs.

The right hand side of Figure 3 shows the results that we obtained for $N = 2^{10}$ and $\gamma = 64$. We do not include the plots corresponding to the other simulation settings, because they are very similar to the one in Figure 3. As we can see, the group-based scheme provides a higher level of privacy when the number of compromised tags does not exceed a threshold. Above the threshold, the key-tree based scheme becomes better, however, in this region, both schemes provide virtually no privacy. Thus, for any practical purposes, our group-based scheme is better than the

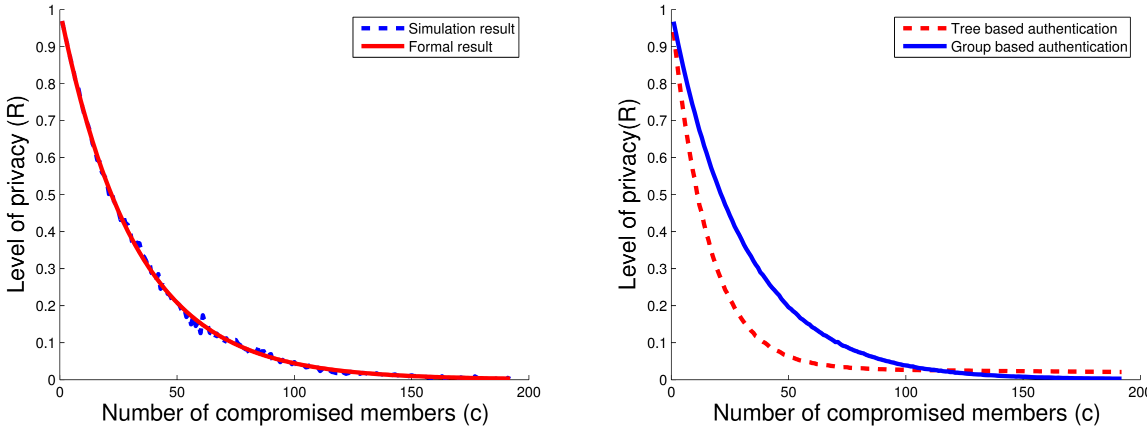

Figure 3. On the left hand side: The analytical results obtained for the expected value of R match the averaged results of ten simulations. The parameters are: $N = 2^{14}$ and $\gamma = 64$. On the right hand side: Results of the simulation aiming at comparing the key-tree based scheme and the group-based scheme. The curves show the level R of privacy as a function of the number c of the compromised tags. The parameters are: $N = 2^{10}$ and $\gamma = 64$. The confidence intervals are not shown, because they are in the range of 10^{-3}, and therefore, they would be hardly visible. As we can see, the group-based scheme achieves a higher level of privacy when c is below a threshold. Above the threshold, the key-tree based approach is slightly better, however, in this region, both schemes provide virtually no privacy.

key-tree based scheme (even if optimal key-trees are used).

5 Conclusion

In this paper, we proposed a novel private authentication scheme based on symmetric-key cryptography for resource constrained applications in large scale environments. We analyzed the proposed scheme and quantified the level of privacy that it provides. We compared our scheme to the key-tree based scheme originally proposed by Molnar and Wagner, and later optimized by Buttyan, Holczer, and Vajda. We showed that our scheme provides a higher level of privacy than the key-tree based scheme. In addition, the complexity of our scheme for the verifier can be set to be the same as in the key-tree based scheme, while the complexity for the prover is always smaller in our scheme. Hence, our scheme is better than the key-tree based scheme both in terms of privacy and efficiency. The primary application area of our scheme is that of RFID systems, but it can also be used in applications with similar characteristics (e.g., in wireless sensor networks).

Acknowledgements

This work has partially been supported by the Hungarian Scientific Research Fund (T046664), the Mobile Innovation Center, Hungary (www.mik.bme.hu), and the SeVeCom project (IST-027795).

References

[1] G. Avoine, E. Dysli, and P. Oechslin. Reducing time complexity in RFID systems. In B. Preneel and S. Tavares (Eds.), *Selected Areas in Cryptography*, Springer, LNCS 3897, pp 291–306, 2005.

[2] L. Buttyan, T. Holczer, and I. Vajda. Optimal key-trees for tree-based private authentication. In G. Danezis and P. Golle (Eds.), *Privacy Enhancing Technologies*, Springer, LNCS 4258, pp. 332–350, 2006.

[3] D. Chaum. The Dining Cryptographers Problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[4] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In R. Dingledine and P. Syverson (Eds.), *Designing Privacy Enhancing Technologies*, Springer LNCS 2482, pp. 54–68, 2002.

[5] D. Molnar and D. Wagner. Privacy and security in library RFID: issues, practices, and architectures. In *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 210–219, 2004.

[6] K. Nohl and D. Evans. Quantifying Information Leakage in Tree-Based Hash Protocols. In *Proceedings of the Conference on Information and Communications Security*, Springer LNCS 4307, pp. 228–237, 2006.

[7] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of the Privacy Enhancing Technologies (PET) Workshop*, Springer LNCS 2482, pp. 41–53, 2002.