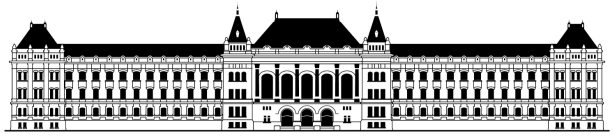

Budapest University of Technology and Economics
Department of Telecommunications

Privacy enhancing protocols for wireless networks

Collection of Ph.D. Theses
of
Tamás Holczer

Supervisor:
Levente Buttyán, Ph.D.

Budapest, Hungary
2012

1 Introduction

In this thesis some privacy aspects of different wireless networks are investigated. These networks are radio frequency identification systems, vehicular ad hoc networks, and wireless sensor networks.

At first, private authentication methods are proposed and analyzed for low cost identification systems. A typical example for such an application is a Radio Frequency Identification System (RFID) system, where the provers are low-cost RFID tags, and the number of the tags can potentially be very large. I study the problem of private authentication in RFID systems. More specifically I propose two methods, that are the privacy efficient key-tree based authentication, and the group based authentication.

The first key-tree based private authentication protocol has been proposed by Molnar and Wagner as a neat way to efficiently solve the problem of privacy preserving authentication based on symmetric key cryptography. However, in the key-tree based approach, the level of privacy provided by the system to its members may decrease considerably if some members are compromised. In this thesis, I analyze this problem, and show that careful design of the tree can help to minimize this loss of privacy. First, a benchmark metric is introduced for measuring the resistance of the system to a single compromised member. This metric is based on the well-known concept of anonymity sets. Then, it is shown how the parameters of the key-tree should be chosen in order to maximize the system's resistance to single member compromise under some constraints on the authentication delay. In the general case, when any member can be compromised, a lower bound is given on the level of privacy provided by the system. Some simulation results are also presented that show that this lower bound is quite sharp.

After that, a novel group based authentication scheme is proposed, similar to the key-tree based method. This scheme is also based on symmetric-key cryptography, and therefore, it is well-suited to resource constrained applications in large scale environments. The proposed scheme is analyzed and shown that it is superior to the previous key-tree based approach for private authentication both in terms of privacy and efficiency.

In the second part of the thesis, I analyze the privacy consequences of inter vehicular communication. The promise of vehicular communications is to make road traffic safer and more efficient. However, besides the expected benefits, vehicular communications also introduce some privacy risk by making it easier to track the physical location of vehicles. One approach to solve this problem is that the vehicles use pseudonyms that they change with some frequency. In this part, the effectiveness of this approach is studied. A model based on the concept of mix zone is defined, the tracking strategy of the adversary is characterized in this model, and a metric is introduced to quantify the level of privacy enjoyed by the vehicles. I also report on the results of an extensive simulation where my model is used to determine the level of privacy achieved in realistic scenarios. In particular, in my simulation, I used a rather complex road map, generated traffic with realistic parameters, and varied the strength of the adversary by varying the number of her monitoring points. My simulation results provide information about the relationship between the strength of the adversary and the level of privacy achieved by changing pseudonyms.

From the first results, it can be seen that untraceability of vehicles is an important requirement in future vehicle communications systems. Unfortunately, heartbeat messages used by many safety applications provide a constant stream of location data, and without any protection measures, they make tracking of vehicles easy even for a passive eavesdropper. One commonly known solution is to transmit heartbeats under pseudonyms that are changed regularly in order to obfuscate the trajectory of vehicles. However, considering a global attacker, this approach is effective only if some silent period is kept during the pseudonym change and several vehicles change their pseudonyms nearly at the same time and at the same location. Unlike other works that proposed explicit synchronization between a group of vehicles and/or

required pseudonym change in a designated physical area (i.e., a static mix zone), a much simpler approach is proposed that does not need any explicit cooperation between vehicles and any infrastructure support. My basic idea is that vehicles should not transmit heartbeat messages when their speed drops below a given threshold, and they should change pseudonym during each such silent period. This ensures that vehicles stopping at traffic lights or moving slowly in a traffic jam will all refrain from transmitting heartbeats and change their pseudonyms nearly at the same time and location. Thus, my scheme ensures both silent periods and synchronized pseudonym change in time and space, but it does so in an implicit way. I also argue that the risk of a fatal accident at a slow speed is low, and therefore, my scheme does not seriously impact safety-of-life. In addition, refraining from sending heartbeat messages when moving at low speed also relieves vehicles of the burden of verifying a potentially large amount of digital signatures, and thus, makes it possible to implement vehicle communications with less expensive equipments.

In the last part, I propose protocols that increase the dependability of wireless sensor networks, which are potentially useful building blocks in cyber-physical systems. Wireless sensor networks can be used in many critical applications such as martial or critical infrastructure protection scenarios. In such a critical scenario, the dependability of the monitoring sensor network can be crucial. One interesting part of the dependability of a network, is how the network can hide its nodes with specific roles from an eavesdropping or active attacker.

In this problem field, I propose protocols which can hide some important nodes of the network. More specifically, I propose two privacy preserving aggregator node election protocols, a privacy preserving data aggregation protocol, and a corresponding privacy preserving query protocol for sensor networks that allow for secure in-network data aggregation by making it difficult for an adversary to identify and then physically disable the designated aggregator nodes. The basic protocol can withstand a passive attacker, while my advanced protocols resist strong adversaries that can physically compromise some nodes. The privacy preserving aggregator protocol allows electing aggregator nodes within the network without leaking any information about the identity of the elected node. The privacy preserving aggregation protocol helps collecting data by the elected aggregator nodes without leaking the information, who is actually collecting the data. The privacy preserving query protocol enables an operator to collect the aggregated data from the unknown and anonymous aggregators without leaking the identity of the aggregating nodes.

2 Research Objective

In wireless networks, it is essential to protect the privacy of the users. In this thesis, some privacy enhancing protocols are given for RFID systems and vehicular ad hoc networks. Some protocols are also proposed to hide some special nodes in wireless sensor networks.

In RFID systems, it can be essential that the user's identity being authenticated remains hidden from an external eavesdropper. There are some solutions to this problem, but they are inefficient in terms of backend load or provides very little privacy if some of the users are compromised. I propose two private authentication schemes for RFID systems which solve this problem. These schemes are the key-tree based and the group based authentication. These schemes can efficiently authenticate a user to the system without revealing the user's identity to an eavesdropper, and loose only a small portion of privacy in case of compromise.

In vehicular ad hoc networks, many safety related applications rely on the periodically broadcast beacons (heartbeat messages). However these beacons pose some privacy threats to the user. The main problem is that the vehicles can be traced based on the beacon messages. This problem can be partially solved by regularly changed pseudonyms. In my thesis, I investigate how

efficient can be any pseudonym change algorithm, and give an efficient algorithm, which changes the pseudonym in silent periods, without decreasing the safety of the drivers.

Wireless sensor networks can be used in critical scenarios, such as critical infrastructure protection, where an attacker might want to disturb the normal operation of the network. Such an attack can be the physical destruction of the aggregator nodes. The first step towards the destruction is the identification of the aggregator nodes. In this thesis, I propose two protocols, which hides the identity of the aggregators in the aggregator election phase, and one of the protocols can also hide the identity of the aggregator while the network is aggregating data.

3 Methodology

In each thesis group, I applied analytical methods or simulations in order to validate that the proposed solutions work as expected in the considered environment. The models are mainly based on probability theory.

In general my methodology was the following. First I defined the metric to be used to measure the efficiency of any solution. After that an attacker model was defined. In general I used rather strong attacker models, which is a good practice in designing security protocols, because if a system is secure and does not leak private information with a strong attacker, then it will act better in real life with weaker attackers. After defining the metric and the attacker model, the third step was the design of the solution. After that the solution was analyzed using the metric and the attacker model using analytical methods or simulation.

The simulations were realized in Matlab and Perl, while the required movement patterns for the vehicular ad hoc networks were generated by SUMO [KHRW02].

4 New Results

4.1 Private Authentication in Resource Constrained Environments

THESES 1: *I present an optimized key-tree based and a group based private authentication scheme for resource constrained environments, which is superior to previous key-tree based approaches in terms of privacy.*

Key-tree based private authentication has been proposed by Molnar and Wagner [MW04] as a neat way to efficiently solve the problem of privacy preserving authentication based on symmetric key cryptography. However, in the key-tree based approach, the level of privacy provided by the system to its members may decrease considerably if some members are compromised. In this thesis, this problem is analyzed, and it is shown that careful design of the tree can help to minimize this loss of privacy. First, a benchmark metric is introduced for measuring the resistance of the system to a single compromised member. This metric is based on the well-known concept of anonymity sets. Then, it is shown how the parameters of the key-tree should be chosen in order to maximize the system's resistance to single member compromise under some constraints on the authentication delay. In the general case, when any member can be compromised, a lower bound is given on the level of privacy provided by the system. Some simulation results are also presented that show that this lower bound is quite sharp.

After the analysis of the key-tree based scheme, a novel private authentication scheme based on groups of users is proposed. The proposed scheme is analyzed and shown that it is superior to the key-tree based approach for private authentication both in terms of privacy and efficiency.

The operation of key-trees is illustrated with Figure 1. The user authenticates itself by a sequence of keys, which defines a leaf on the tree. This approach is useful as the task of the verifier is logarithmic in the number of all possible users. The drawback of this solution is that if a user is compromised, then some other users are partially compromised as well. This effect can be measured by the metric defined as follows:

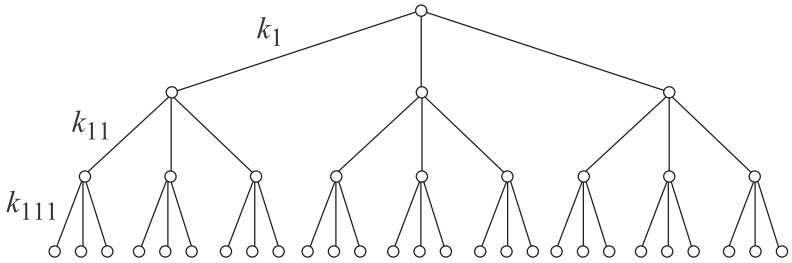

Figure 1: Illustration of a key-tree. There is a unique key assigned to each edge. Each leaf represents a member of the system that possesses the keys assigned to the edges of the path starting from the root and ending in the given leaf. For instance, the member that belongs to the leftmost leaf in the figure possesses the keys k_1, k_{11}, and k_{111}.

THESIS 1.1: *I define a metric (R) used for comparing private authentication methods for resource constrained environments as the normalized expected anonymity set size in case of a single member compromise. I prove that this metric is complement to the one tag tampering based metric (M) defined in [ADO05].*

The normalized expected anonymity set size in case of a single member compromise, is the anonymity set size of a randomly chosen user if one user is compromised normalized with its maximum value.

One can calculate (R) for key-trees using the following equation:

$$\begin{aligned} R &= \frac{\bar{S}}{N} = \sum_{i=0}^{\ell} \frac{|P_i|^2}{N^2} \\ &= \frac{1}{N^2}\left(1 + (b_\ell - 1)^2 + ((b_{\ell-1} - 1)b_\ell)^2 + \ldots + ((b_1 - 1)b_2 b_3 \ldots b_\ell)^2\right) \\ &= \frac{1}{N^2}\left(1 + (b_\ell - 1)^2 + \sum_{i=1}^{\ell-1}(b_i - 1)^2 \prod_{j=i+1}^{\ell} b_j^2\right), \end{aligned} \tag{1}$$

where P_i is the size of the ith subset after on member is compromised, N is the number of all nodes, and b_i is the branching factor on the ith level. The metric can be easily understood by observing Figure 2. In the remaining part, this metric will be used to compare different trees and the group based solution.

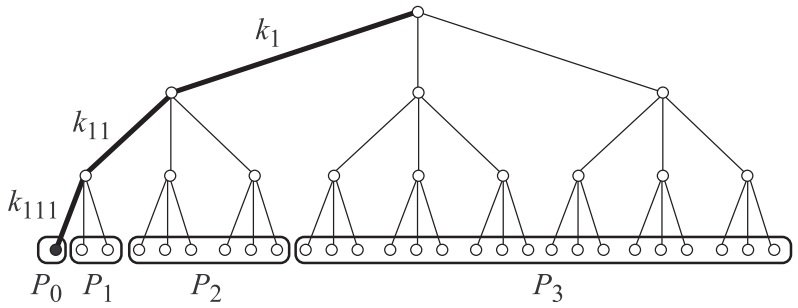

Figure 2: Illustration of a single member compromise in the key-tree. Without loss of generality, it is assumed that the member corresponding to the leftmost leaf in the figure is compromised. This means that the keys k_1, k_{11}, and k_{111} become known to the adversary. This knowledge of the adversary partitions the set of members into anonymity sets P_0, P_1, ... of different sizes. Members that belong to the same subset are indistinguishable to the adversary, while it can distinguish between members that belong to different subsets. For instance, the adversary can recognize a member in subset P_1 by observing the usage of k_1 and k_{11} but not that of k_{111}, where each of these keys are known to the adversary. Members in P_3 are recognized by not being able to observe the usage of any of the keys known to the adversary.

The metric M used in [ADO05] is defined in that paper as:

1. The attacker has one tag T_0 (e.g., her own) she can tamper with and thus obtains its complete secret. For the sake of calculation simplicity, we assume that T_0 is put back into circulation. When the number of tags in the system is large, this does not significantly affect the results.

2. The attacker then chooses a target tag T. She can query it as much as she wants but she cannot tamper with it.

3. Given two tags T_1 and T_2 such that $T \in \{T_1, T_2\}$, we say that the attacker succeeds if she definitely knows which of T_1 and T_2 is T. We define the probability to trace T as being the probability that the attacker succeeds. To do that, the attacker can query T_1 and T_2 as many times as she wants but, obviously, cannot tamper with them.

The proof of that M and R are complement ($M + R = 1$) is based on their definition:

In the following $P_1 \dots P_k$ are the subsets of the tags after the compromise of some tags ($\sum_{i=1}^{k} P_i = N$).

In the third step of calculating M, the attacker can be successful if (and only if) T_1 and T_2 belongs to different subsets.

The probability of the attacker's success is the probability that two randomly chosen tags belongs to two different subsets. This probability can be calculated as follows:

$$M = 1 - \Pr(T_1, T_2 \text{ are in } P_1) - \ldots - \Pr(T_1, T_2 \text{ are in } P_k) = 1 - \sum_{i=1}^{k} \left(\frac{P_i}{N}\right)^2$$

This is the complement of the metric R ($M + R = 1$).

In the following a method is given, how the optimal branching factor can be found:

THESIS 1.2: *I give a recursive greedy algorithm that finds the optimal branching factor vector using metric R for a given number of users (N) and a maximal delay D. I prove the optimality of the resulting branching factor vector.*

The problem of finding the optimal branching factor vector can be described as an optimization problem as follows: *Given the total number N of members and the upper bound D_{max} on the maximum authentication delay, find a branching factor vector $B = (b_1, b_2, \dots b_\ell)$ such that $R(B)$ is maximal subject to the following constraints:*

$$\prod_{i=1}^{\ell} b_i = N \tag{2}$$

$$\sum_{i=1}^{\ell} b_i \leq D_{max} \tag{3}$$

The following algorithm finds the optimal branching factor vector:

The illustration of the operation of the algorithm can be found in Table 1.

Table 1: Illustration of the operation of the recursive function f when called with $B = (5, 5, 5, 3, 3, 3, 2, 2, 2)$ and $d = 90$. The rows of the table correspond to the levels of the recursion during the execution.

recursion level	B	d	B'	$\prod(B')$
1	$(5, 5, 5, 3, 3, 3, 2, 2, 2)$	90	$(3, 3, 2, 2, 2)$	72
2	$(5, 5, 5, 3)$	18	(5)	5
3	$(5, 5, 3)$	13	(5)	5
4	$(5, 3)$	8	(5)	5
5	(3)	3	(3)	3

Algorithm 1 Optimal branching factor generating algorithm

$f(B, d)$
if $\sum(B) > d$ **then**
  exit (no solution exists)
else
  find $B' \subseteq B$ such that
  $\prod(B') + \sum(B \setminus B') \leq d$ and $\prod(B')$ is maximal
end if
if $B' = B$ **then**
  return $(\prod(B'))$
else
  return $\prod(B')|f(B \setminus B', d - \prod(B'))$
end if

The proof of optimality is based on a set of lemmas, and can be found on the dissertation. The lemmas are the following:

Lemma 1. Let N and D_{max} be the total number of members and the upper bound on the maximum authentication delay, respectively. Moreover, let B be a branching factor vector and let B^* be the vector that consists of the sorted permutation of the elements of B in decreasing order. If B satisfies the constraints of the optimization problem defined above, then B^* also satisfies them, and $R(B^*) \geq R(B)$.

Lemma 2. Let $B = (b_1, b_2, \ldots b_\ell)$ be a sorted branching factor vector (i.e., $b_1 \geq b_2 \geq \ldots \geq b_\ell$). The following lower and upper bounds on $R(B)$ can be given:

$$\left(1 - \frac{1}{b_1}\right)^2 \leq R(B) \leq \left(1 - \frac{1}{b_1}\right)^2 + \frac{4}{3b_1^2} \tag{4}$$

Lemma 3. Let N and D_{max} be the total number of members and the upper bound on the maximum authentication delay, respectively. Moreover, let $B = (b_1, b_2, \ldots, b_\ell)$ and $B' = (b'_1, b'_2, \ldots, b'_{\ell'})$ be two sorted branching factor vectors that satisfy the constraints of the optimization problem defined above. Then, $b_1 > b'_1$ implies $R(B) \geq R(B')$.

Lemma 4. Let N and D_{max} be the total number of members and the upper bound on the maximum authentication delay, respectively. Moreover, let $B = (b_1, b_2, \ldots, b_\ell)$ and $B' = (b'_1, b'_2, \ldots, b'_{\ell'})$ be two sorted branching factor vectors such that $b_i = b'_i$ for all $1 \leq i \leq j$ for some $j < \min(\ell, \ell')$, and both B and B' satisfy the constraints of the optimization problem defined above. Then, $b_{j+1} > b'_{j+1}$ implies $R(B) \geq R(B')$.

Theorem 1. Let N and D_{max} be the total number of members and the upper bound on the maximum authentication delay, respectively. Moreover, let B be a vector that contains the prime factors of N. Then, $f(B, D_{max})$ is an optimal branching factor vector for N and D_{max}.

Proof. I will give a sketch of the proof. Let $B^* = f(B, D_{max})$, and let us assume that there is another branching factor vector $B' \neq B^*$ that also satisfies the constraints of the optimization problem and $R(B') > R(B^*)$. I will show that this leads to a contradiction, hence B^* should be optimal.

Let $B^* = (b^*_1, b^*_2, \ldots, b^*_{\ell^*})$ and $B' = (b'_1, b'_2, \ldots, b'_{\ell'})$. Recall that B^* is obtained by first maximizing the first element in the vector, therefore, $b^*_1 \geq b'_1$ must hold. If $b^*_1 > b'_1$, then $R(B^*) \geq R(B')$ by Lemma 3, and thus, B' cannot be a better vector than B^*. This means that $b^*_1 = b'_1$ must hold.

We know that once b_1^* is determined, my algorithm continues by maximizing the next element of B^*. Hence, $b_2^* \geq b_2'$ must hold. If $b_2^* > b_2'$, then $R(B^*) \geq R(B')$ by Lemma 4, and thus, B' cannot be a better vector than B^*. This means that $b_2^* = b_2'$ must hold too.

By repeating this argument, finally, we arrive to the conclusion that $B^* = B'$ must hold, which is a contradiction. $\diamond$ $\square$

Until now, it was assumed that none or only one user is compromised. In the following it is analyzed what happens if more than one user is compromised. The problem is visualized in Figure 3.

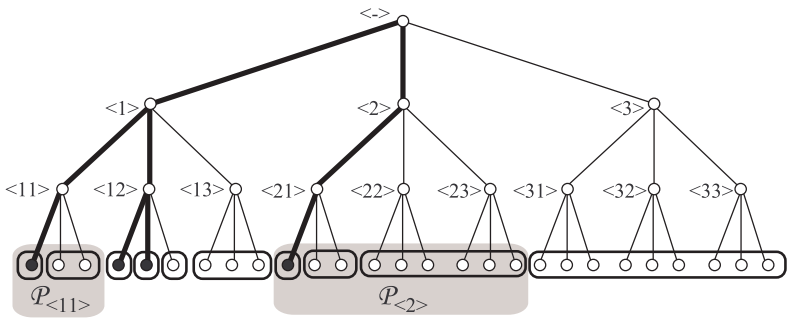

Figure 3: Illustration of what happens when several members are compromised. Just as in the case of a single compromised member, the members are partitioned into anonymity sets, but now the resulting subsets depend on the number of the compromised members, as well as on their positions in the tree. Nevertheless, the expected size of the anonymity set of a randomly selected member is still a good metric for the level of privacy provided by the system, although, in this general case, it is more difficult to compute.

THESIS 1.3: *I give a lower bound on the privacy of the system when any number of members could be compromised.*

In the following, a leaf is called compromised if it belongs to a compromised member, and a non-leaf vertex is called compromised if it lies on a path that leads to a compromised leaf in the tree. If vertex v is compromised, then

- K_v denotes the set of the compromised children of v, and $k_v = |K_v|$;
- $\mathcal{P}_v$ denotes the set of subsets (anonymity sets) that belong to the subtree rooted at v (see Figure 3 for illustration); and
- $\bar{S}_v$ denotes the average size of the subsets in $\mathcal{P}_v$.

We are interested in computing $\bar{S}_{\langle - \rangle}$. We can do that as follows:

$$\begin{aligned}
\bar{S}_{\langle - \rangle} &= \sum_{P \in \mathcal{P}_{\langle - \rangle}} \frac{|P|^2}{b_1 b_2 \dots b_\ell} \\
&= \frac{((b_1 - k_{\langle - \rangle}) b_2 \dots b_\ell)^2}{b_1 b_2 \dots b_\ell} + \sum_{v \in K_{\langle - \rangle}} \sum_{P \in \mathcal{P}_v} \frac{|P|^2}{b_1 b_2 \dots b_\ell} \\
&= \frac{((b_1 - k_{\langle - \rangle}) b_2 \dots b_\ell)^2}{b_1 b_2 \dots b_\ell} + \frac{1}{b_1} \sum_{v \in K_{\langle - \rangle}} \bar{S}_v \qquad (5)
\end{aligned}$$

In general, for any vertex $\langle i_1, \ldots, i_j \rangle$ such that $1 \leq j < \ell - 1$:

$$\bar{S}_{\langle i_1,\ldots,i_j \rangle} = \frac{((b_{j+1} - k_{\langle i_1,\ldots,i_j \rangle})b_{j+2} \ldots b_\ell)^2}{b_{j+1} \ldots b_\ell} + \frac{1}{b_{j+1}} \sum_{v \in K_{\langle i_1,\ldots,i_j \rangle}} \bar{S}_v \tag{6}$$

Finally, for vertices $\langle i_1, \ldots, i_{\ell-1} \rangle$ just above the leaves, we get:

$$\bar{S}_{\langle i_1,\ldots,i_{\ell-1} \rangle} = \frac{(b_\ell - k_{\langle i_1,\ldots,i_{\ell-1} \rangle})^2}{b_\ell} + \frac{k_{\langle i_1,\ldots,i_{\ell-1} \rangle}}{b_\ell} \tag{7}$$

In the following, we can derive a formula which is independent of the position of the compromised members, based on the assumption that the compromised members are distributed uniformly at random over the leaves of the key-tree.

We can estimate $k_{\langle - \rangle}$ for the root as follows:

$$k_{\langle - \rangle} \approx \min(c, b_1) = k_0 \tag{8}$$

If a vertex $\langle i \rangle$ at the first level of the tree is compromised, then the number of compromised leaves in the subtree rooted at $\langle i \rangle$ is approximately $c/k_0 = c_1$. Then, we can estimate $k_{\langle i \rangle}$ as follows:

$$k_{\langle i \rangle} \approx \min(c_1, b_2) = k_1 \tag{9}$$

In general, if vertex $\langle i_1, \ldots, i_j \rangle$ at the j-th level of the tree is compromised, then the number of compromised leaves in the subtree rooted at $\langle i_1, \ldots, i_j \rangle$ is approximately $c_{j-1}/k_{j-1} = c_j$, and we can use this to approximate $k_{\langle i_1,\ldots,i_j \rangle}$ as follows:

$$k_{\langle i_1,\ldots,i_j \rangle} \approx \min(c_j, b_{j+1}) = k_j \tag{10}$$

Using these approximations in expressions (5 – 7), we can derive an approximation for $\bar{S}_{\langle - \rangle}$, which is denoted by $\bar{S}_0$, in the following way:

$$\begin{aligned}
\bar{S}_{\ell-1} &= \frac{(b_\ell - k_{\ell-1})^2}{b_\ell} + \frac{k_{\ell-1}}{b_\ell} && (11)\\
\ldots &\quad \ldots \\
\bar{S}_j &= \frac{((b_{j+1} - k_j)b_{j+2} \ldots b_\ell)^2}{b_{j+1} \ldots b_\ell} + \frac{k_j}{b_{j+1}} \bar{S}_{j+1} && (12)\\
\ldots &\quad \ldots \\
\bar{S}_0 &= \frac{((b_1 - k_0)b_2 \ldots b_\ell)^2}{b_1 \ldots b_\ell} + \frac{k_0}{b_1} \bar{S}_1 && (13)
\end{aligned}$$

An illustration of this approximation is presented on Figure 4.

In the following I introduce a group based solution, where the members are put into groups instead of trees. The approach is displayed on Figure 5 and Figure 6. Each member posses a group key and an own key. During authentication, first the group is identified by the usage of the group key. After that, the member proves its identity by the usage of the own key.

THESIS 1.4: *I propose a group based scheme for private authentication, which is superior to the key-tree based approach both in terms of privacy and storage space.*

The storage space required by the group based scheme is always two keys (one group key and one own key) on the prover side, while in most of the cases it is at least two in case of the tree based scheme.

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.

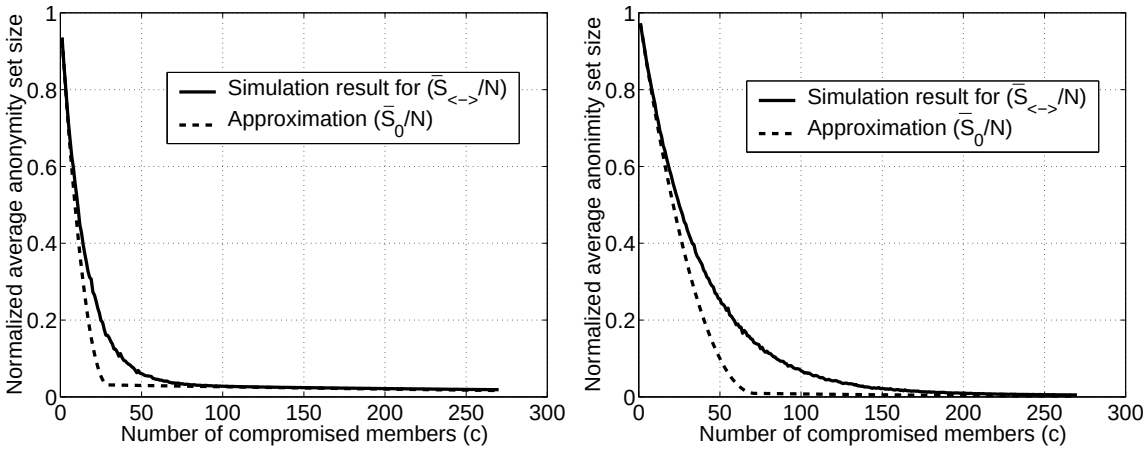

Figure 4: Simulation results for branching factor vectors $(30, 30, 30)$ (left hand side) and $(72, 5, 5, 5, 3)$ (right hand side). As we can see, $\bar{S}_0/N$ approximates $\bar{S}_{\langle - \rangle}/N$ quite well, and in general it provides a lower bound on it.

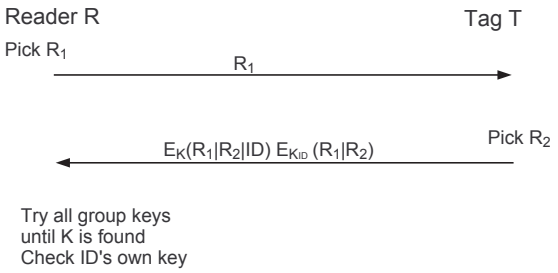

Figure 5: Operation of the group-based private authentication scheme. K is the group key stored by the tag,K_{ID} is the tag's own secret key, ID is the identifier of the tag, R_1 and R_2 are random values generated by the reader and the tag, respectively, | denotes concatenation, and $E_K()$ denotes symmetric-key encryption with K.

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 with constant branching factor, 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.

The result of a compromise of a member (R) can be calculated as follows:

$$R = \frac{1}{N^2} \left(nC + (n(\gamma - C))^2 \right) \tag{14}$$

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.

A comparison is given between the tree and the group based scheme on Figure 7 using the expected value of R. 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.

The related publications are [C5, C6, C1].

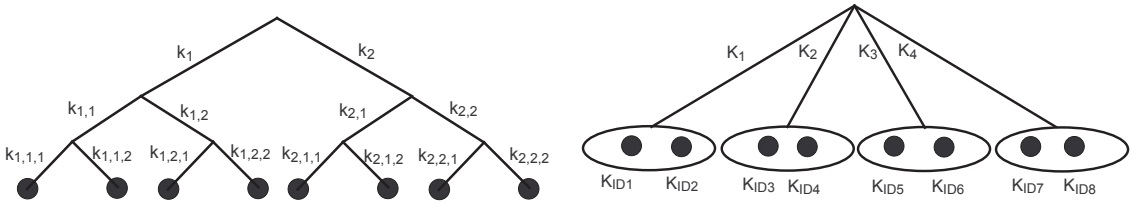

Figure 6: **On the left hand side**: The tree-based authentication protocol uses a tree, where the tags correspond to the leaves of the tree. Each tag stores the keys along the path from the root to the leaf corresponding to the given tag. When authenticating itself, a tag uses all of its keys. The reader 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.

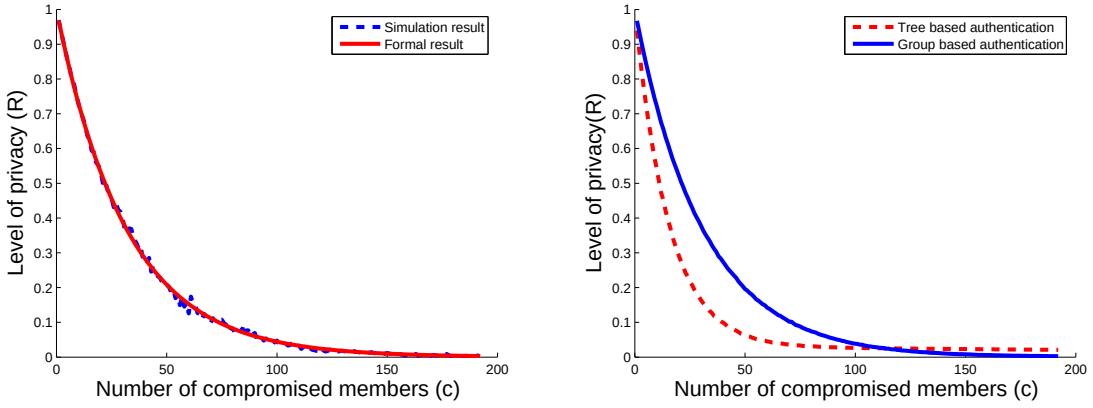

Figure 7: **On the left hand side**: The analytical results obtained for the expected value of R match the averaged results of ten simulations. The parameters are: $N = 2^{14}$ and $\gamma = 64$. **On the right hand side**: Results of the simulation aiming at comparing the key-tree based scheme and the 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.

4.2 Location Privacy in Vehicular Ad Hoc Networks

THESES 2: *For VANETS, I define a model for analyzing the location privacy of the vehicles based on the concept of the mix zone, characterize the tracking strategy of the adversary in this model, and introduce a metric to quantify the level of privacy enjoyed by the vehicles. I also propose an approach called SLOW for implementing mix zones that does neither require extensive RSU support nor complex communication between vehicles, and that does not endanger safety-of-life to any significant extent, while providing both syntactic mixing and semantic mixing. I investigate the properties of the solution through simulation.*

Vehicles using VANETs broadcast beacon (heartbeat) messages frequently to inform the nearby vehicles about their actual positions and speed. This information can be used to avoid accidents, and help the traffic to go as smoothly as possible.

An attacker can eavesdrop these beacon messages in order to track the route of some vehicles. If the attacker can only eavesdrop on some certain points of the city, then the city is split into an observed zone, and a remaining zone. The remaining zone acts as a mix for the vehicles, thus it will be called mix zone in the following. This partitioning is displayed on Figure 8.

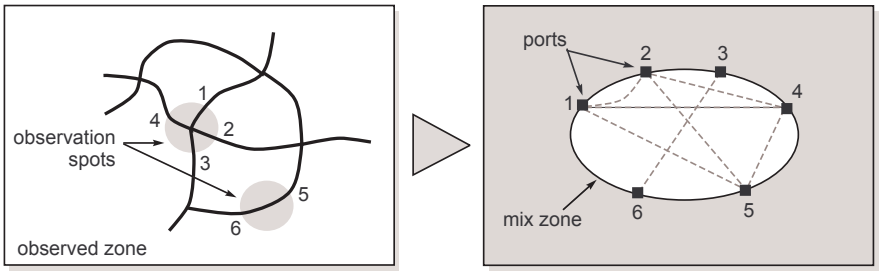

Figure 8: **On the left hand side:** The figure illustrates how a road network is divided into an observed and an unobserved zone in my model. In the figure, the observed zone is grey, and the unobserved zone is white. The unobserved zone functions as a *mix zone*, because the vehicles change pseudonyms and mix within this zone making it difficult for the adversary to track them. **On the right hand side:** The figure illustrates how the road network on the left can be abstracted as single mix zone with six ports.

THESIS 2.1: *I introduce a metric to quantify the level of privacy enjoyed by the vehicles in the mix zone model, and show a tracking strategy which is proven to be optimal in this model.*

The attacker can have some statistical knowledge about the mix zone. This knowledge is subsumed in a model that consists of a matrix $Q = [q_{ij}]$ of size $M \times M$, where M is the number of ports of the mix zone, and M^2 discrete probability density functions $f_{ij}(t)$ $(1 \leq i, j \leq M)$. q_{ij} is the conditional probability of exiting the mix zone at port j given that the entry point was port i. $f_{ij}(t)$ describes the probability distribution of the delay when traversing the mix zone between port i and port j.

The general objective of the adversary is to relate exiting events to entering events. More specifically, in my model, the adversary picks a vehicle v in the observed zone and tracks its movement until it enters the mix zone. In the following, the port at which v entered the mix zone is denoted by s. Then, the adversary observes the exiting events for a time T such that the probability that v leaves the mix zone before T is close to 1 (i.e., $\Pr\{t_{out} < T\} = 1 - \epsilon$, where ϵ is a small number, typically, in the range of $0.005 - 0.01$, and t_{out} is the random variable denoting the time at which the selected vehicle v exits the mix zone). For each exiting vehicle v', the adversary determines the probability that v' is the same as v. For this purpose, she uses her observations and the model of the mix zone. Finally, she decides which exiting vehicle corresponds to the selected vehicle v.

The decision algorithm used by the adversary is intuitive and straightforward: The adversary knows that the selected vehicle v entered the mix zone at port s and in timeslot 0. For each exiting event $k = (j, t)$ that the adversary observes afterwards, she can compute the probability p_{jt} that k corresponds to the selected vehicle as $p_{jt} = q_{sj} f_{sj}(t)$ (i.e., the probability that v chooses port j as its exit port given that it entered the mix zone at port s multiplied by the probability that it covers the distance between ports s and j in time t). The adversary decides

for the vehicle for which p_{jt} is maximal. The adversary is successful if the decided vehicle is indeed v.

Theorem 2. The above described decision algorithm realizes the Bayesian decision.

The importance of the theorem is that the Bayesian decision minimizes the error probability, thus, it is in some sense the ideal decision algorithm for the adversary. The proof that the algorithm realizes a Bayesian decision can be found in the dissertation.

The following simulations were made with SUMO [KHRW02], which is an open source micro-traffic simulator, developed by the Center for Applied Informatics (ZAIK) and the Institute of Transport Research at the German Aerospace Center. The above characterized attacker was realized by a Perl script.

THESIS 2.2: *I show through simulation, that the success probability of the attacker depends on the density of the traffic and the number of eavesdropping points owned by the attacker. I also show, that the success probability of the attacker saturates over a given number of eavesdropping points.*

Figure 9 contains the resulting success probabilities of the adversary as a function of her strength. The different curves belong to different traffic intensities. The results are quite intuitive: we can conclude that the stronger the adversary, the higher her success probability. Note, however, that from above a given strength, the success probability saturates at about 60 %. Higher success probabilities can not be achieved, because the order of the vehicles may change between junctions without the adversary being capable of tracking that. Note also that the saturation point is reached with the control of only the half of the junctions. The intensity of the traffic is much less important parameter, than the strength of the attacker. The success probability of the attacker is nearly independent from the intensity of the traffic above a given attacker strength.

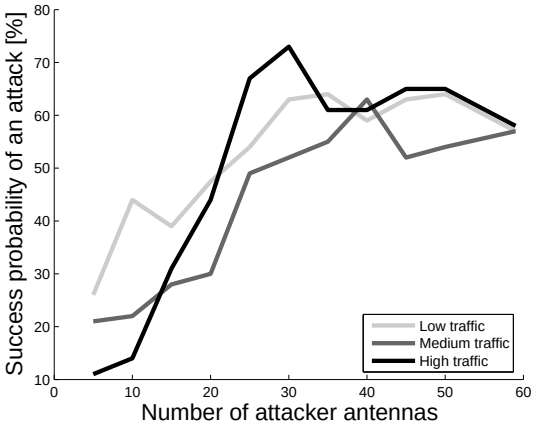

Figure 9: Success probabilities of the adversary as a function of her strength. The three curves represent three different scenarios (the darker the line, the more intensive the traffic).

Some more insight on the attacker can be found on Figure 10. This shows that while the size of the anonymity set (V) seems to be large (which seemingly makes the adversary's decision difficult), the distribution of the members is highly non-uniform.

In the following, a global eavesdropping attacker is assumed, and a pseudonym changing algorrithm is proposed. The proposed solution is called SLOW (Silence at LOW speeds). It

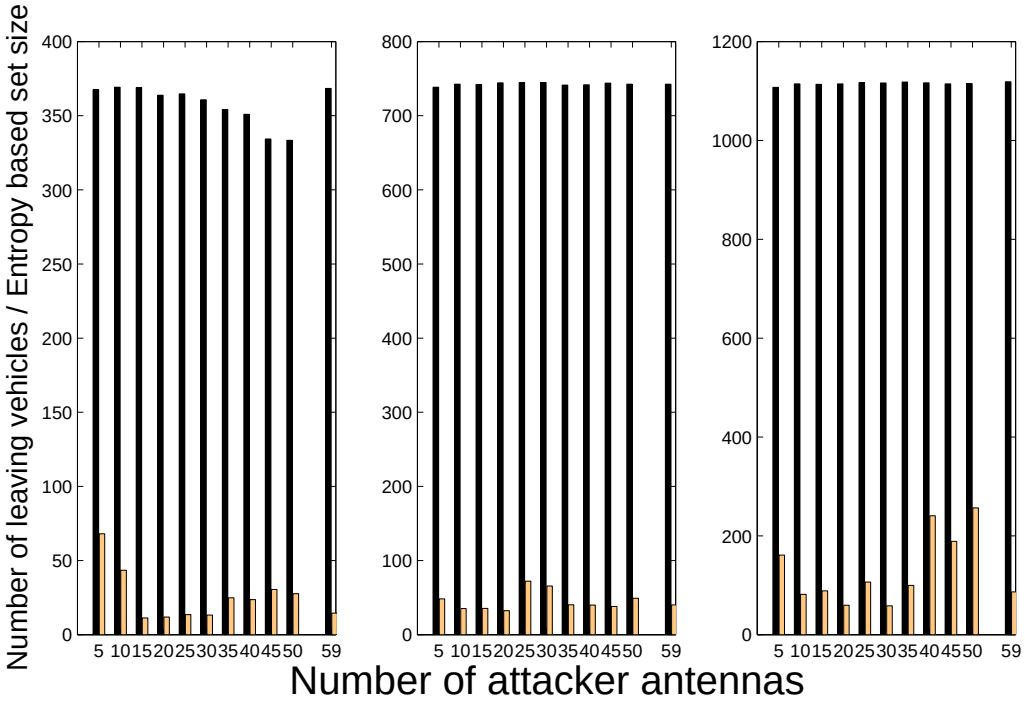

Figure 10: The dark bars show how the size of the set V of the vehicles that exit the mix zone during the observation period varies with the strength of the adversary (y axis: number of attacker antennas). The three sub-figures are related to the three different traffic situations (low traffic – left, medium traffic – middle, high traffic – right). The light bars illustrate the effective size of V. As we can see, the effective size is much smaller than the real size, which means that distribution corresponding to the members of V is highly non-uniform.

works as follows. A threshold speed is chosen v_T, say $v_T = 30$ km/h. A vehicle will *not* broadcast any heartbeat message, or any other message containing location or trajectory data in the clear, if it is traveling below speed v_T, unless this is necessary for safety- of-life reasons. If the vehicle has not sent a message for a certain period of time, then it changes pseudonyms (identifiers at all layer of the network stack and related certificates) before the next transmission. Traffic signals in a crowded urban area seem like an ideal location for such a pseudonym change: whenever a crowd of vehicles stop at a traffic signal, they may go into one of several lanes, they may choose to turn or not to turn, and so on.

THESIS 2.3: *I propose an algorithm for pseudonym change called SLOW, and I show that it provides high level of privacy to the drivers, while it reduces the maximal communication overhead.*

Without any protection, one can imagine that an eavesdropping attacker can easily track vehicles periodically sending beacon messages. Some simulation results made in SUMO is presented on Figure 11 to confirm this statement.

When there is some kind of defense, then an attacker strategy must be defined. In the following, my attacker model essentially assumes that traffic at an intersection follows the FIFO (First In First Out) principle. While this is clearly not the case in practice, my attacker still achieves a reasonable success rate in a single intersection as shown in Figure 12 (more details about the operation of the attacker can be found in the dissertation). One can see, for instance,

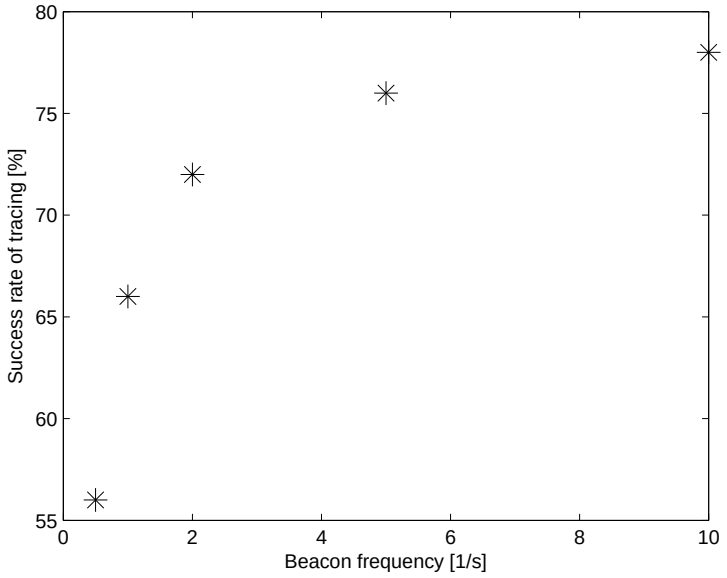

Figure 11: Success rate of an attacker performing vehicle tracking by semantic linking of heartbeat messages when no defense mechanisms are in use.

that when the total number of vehicles is 100, the attacker can still track a target vehicle through a single intersection with probability around $\frac{1}{2}$.

Figure 13 shows the success rate of the attacker in the general case, when the target traverses multiple intersections between its starting and destination points. As expected, the tracking capabilities of the attacker in this case are worse than in the single intersection case. The quantitative results of my simulation experiments suggest that only around 10% of the vehicles can be tracked fully by the attacker when the threshold speed is larger than 22 km/h (approximately 6 m/s).

The effectiveness of the attacker depends on the v_T threshold speed and the density of the vehicles. In general the higher the threshold speed at which vehicles stop sending heartbeats, the higher the chance that the attacker loses the target (i.e., the lower the chance of successful tracking). Moreover, in a dense network, it is more difficult to track vehicles. Note, however, that there is an important difference in practice between the traffic density and the threshold speed, namely, that the threshold speed can be influenced by the owner of the vehicle, while the traffic density cannot be.

In Figure 14, the results of some simple calculations can be seen showing the number of signature verifications performed as a function of the average speed. In this calculation, it is assumed that vehicles follow each other within 2 seconds. The communication range is assumed to be 100 m and the heartbeat frequency is 10 Hz. It can be seen in the figure that, in a traffic jam on an 8-lane road, each vehicle must verify as many as approximately 8,000 signatures per second. If SLOW is used with a threshold speed of around 30 km/h (approximately 8 m/s), then the vehicles never need to verify more than 1,000 signatures per second (assuming all other parameters are the same as before). This approach also works well in combination with congestion control where the transmission power is reduced in high density traffic scenarios. My approach therefore makes the hardware requirements of the OBU much lower and enables the use of less expensive devices.

The related publications are [C7, J2, O1, C8].

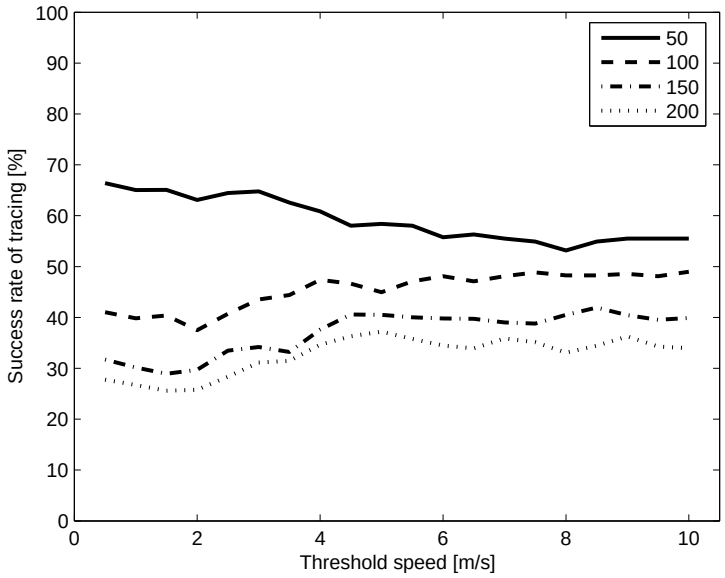

Figure 12: Success rate of the simple attacker in a single intersection. Different curves belong to different experiments with the total number of vehicles given in the legend.

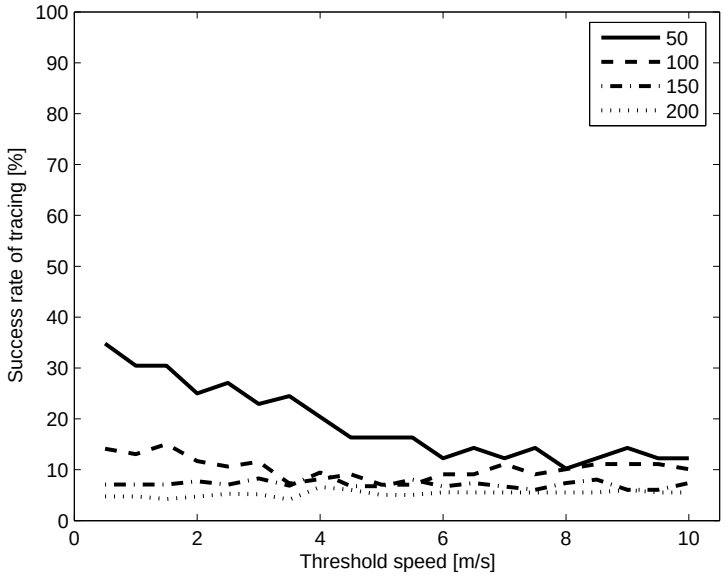

Figure 13: Success rate of the simple attacker in the general case, when the target traverses multiple intersections between its starting and destination points. Different curves belong to different experiments with the total number of vehicles given in the legend.

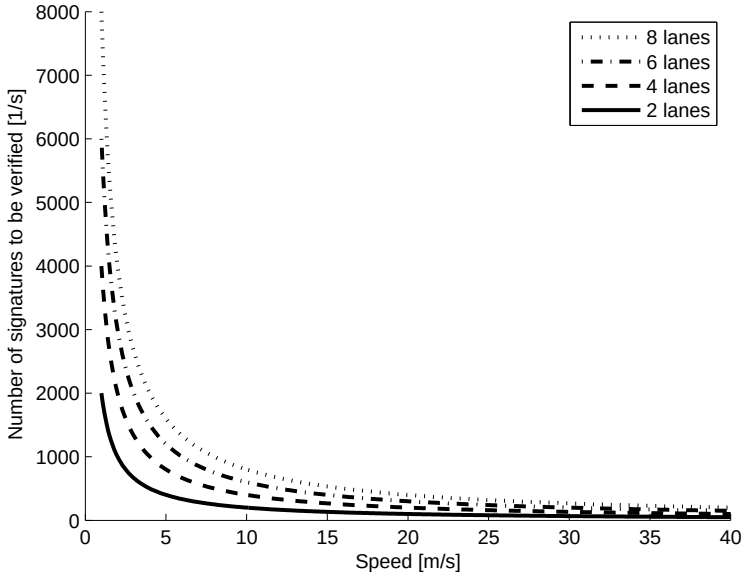

Figure 14: Number of signatures to be verified as a function of the average speed. The communication range is 100 m, and the heartbeat frequency is 10 Hz. Safety distance between the vehicles depends on their speed.

4.3 Anonymous Aggregator Election and Data Aggregation in Wireless Sensor Networks

THESES 3: *I present two anonym aggregator election protocols for wireless sensor networks, which can hide the identity of the elected aggregators from attackers.*

Wireless sensor and actuator networks are potentially useful building blocks for cyber-physical systems. Those systems must typically guarantee high-confidence operation, which induces strong requirements on the dependability of their building blocks, including the wireless sensor and actuator network. Dependability means resistance against both accidental failures and intentional attacks, and it should be addressed at all layers of the network architecture, including the networking protocols and the distributed services built on top of them, as well as the hardware and software architecture of the sensor and actuator nodes themselves. Within this context, in this part, the focus is on the security aspects of aggregator node election and data aggregation protocols in wireless sensor networks, especially on hiding the identity of the elected aggregator nodes.

THESIS 3.1: *I propose a simple anonym aggregator election protocol, which can withstand passive and active non-compromising attacks.*

The pseudo-code of the protocol is given in Algorithm 2, and a more detailed explanation of the protocol's operation is presented below. The protocol consists of two rounds, where the length of each round is τ. The nodes are synchronized, they all know when the first round begins, and what the value of τ is. At the beginning, each node starts two random timers, T1 and T2, where T1 expires in the first round (uniformly at random) and T2 expires in the second round (uniformly at random). Each node also initializes at random a binary variable, called announFirst, that determines in which round the node would like to send a cluster aggregator

Algorithm 2 Basic privacy preserving cluster aggregator election algorithm

```
start T1, expires in rand(0,τ) //timer, expires in round 1
start T2, expires in rand(τ,2τ) //timer, expires in round 2
announFirst = (rand(0,1) ≤ γ)
CAID = -1 // ID of the cluster aggregator of the node
while T1 NOT expired do
  if receive ENC(announcement) AND (CAID = -1) then
    CAID = ID of sender of announcement
  end if
end while
// T1 expired
if announFirst AND (CAID = -1) then
  broadcast ENC(announcement);
  CAID = ID of node itself;
else
  broadcast ENC(dummy);
end if
while T2 NOT expired do
  if receive ENC(announcement) AND (CAID = -1) then
    CAID = ID of sender of announcement
  end if
end while
// T2 expired
if (NOT announFirst) AND (CAID = -1) then
  broadcast ENC(announcement);
  CAID = ID of node itself;
else
  broadcast ENC(dummy);
end if
```

announcement. The probability that announFirst is set to the first round is γ, which is a system parameter.

In the first round, every node S waits for its first timer T1 to expire. If S receives an announcement before T1 expires, then the sender of the announcement will be the cluster aggregator of S. When T1 expires, S broadcasts a message as follows: If announFirst is set to the first round and S has not received any announcement yet, then S sends an announcement, in which it announces itself as a cluster aggregator. Otherwise, S sends a dummy message. In both cases, the message is encrypted (denoted by ENC() in the algorithm) such that only the cluster peers of S can decrypt it.

The second round is similar to the first round. When T2 expires S broadcasts a message as follows: If announFirst is set to the second round and S has not received any announcement yet, then S sends an announcement, otherwise, S sends a dummy message. In both cases, the message is encrypted.

It is easy to see that at the end of the second round each node is either a cluster aggregator or it is associated with a cluster aggregator whose ID is stored in variable CAID. Without the second round, a node can remain unassociated, if it sends and receives only dummy messages in the first round. In addition, a passive observer only sees that every node sends two encrypted messages, one in each round. An active attacker cannot do anything either without the knowledge of the secret keys. The keys can only be revealed by the compromise of some nodes, which leads us to the next protocol.

THESIS 3.2: *I propose a complex anonym aggregator election protocol, which can withstand passive and active non-compromising and compromising attacks as well.*

The definition of the election process is the following. The election process consists of two main steps: (i) Every node decides, whether it wants to be an aggregator, based on some random values. This step does not need any communication, the nodes compute the results locally. (ii) In the second step, an anonymous veto protocol is run, which reveals only the information that at least one node elected itself to be aggregator node. If no aggregator is elected, it will be clear for every participant, and every participant can run the election protocol again.

Step (i) can be implemented easily. Every node elects itself aggregator with a given probability p. The result of the election is kept secret, the participants only want to know that the number c of aggregators is not zero, without revealing the identity of the cluster aggregators. This is advantageous, because in case of node compromise, the attacker learns only whether the compromised node is an aggregator, but nothing about the identity or the number of the other aggregators. Let us denote the random variable representing the number of elected aggregators with C. It is easy to see that the distribution of C is binomial (N is the total number of nodes in one cluster):

$$\Pr(C = c) = \left(\begin{array}{c} N \\ c \end{array} \right) p^c \, (1-p)^{N-c}$$

The expected number of aggregators after the first step is: $c_E = Np$. So if on average $\hat{c}$ cluster aggregator is needed, then p should be $\frac{\hat{c}}{N}$ (this formula will be slightly modified after considering the results of the second step).

The probability that no cluster aggregator is elected is: $(1-p)^N$. To avoid this anarchical situation when no node is elected, the nodes must run step (ii) which proves that at least one node is elected as aggregator node, but the identity of the aggregator remains secret. This problem can be solved by an anonymous veto protocol. Such a protocol is suggested by Hao and Zieliński in [HZ06].

The operation of the anonym veto protocol consists of two consecutive rounds (G is a publicly agreed group with order q and generator g):

1. First, every participant i selects a secret random value: $x_i \in \mathbb{Z}_q$. Then g_i^x is broadcast with a knowledge proof. The knowledge proof is needed to ensure that the participant knows x_i without revealing the value of x_i. Without the knowledge proof, the node could choose g_i^x in a way to influence the result of the protocol (it is widely believed that for a given $g_i^x (\mathrm{mod}\ p)$ it is hard to find $x_i (\mathrm{mod}\ p)$, this problem is known as the discrete logarithm problem). Then every participant checks the knowledge proofs, and computes a special product of the received values:

$$g^{y_i} = \prod_{j=1}^{i-1} g^{x_j} \Big/ \prod_{j=i+1}^{N} g^{x_j}$$

2. $g^{y_i c_i}$ is broadcast with a knowledge proof (the knowledge proof is needed to ensure that the node cannot influence the election maliciously afterwards). c_i is set to x_i for non aggregators, while a random r_i value for aggregators.

The product $P = \prod_{i=1}^{N} g^{c_i y_i}$ equals to 1 if and only if no cluster aggregator is elected (none vetoed the question: Is the number of cluster aggregators elected zero?). If no aggregator is elected, then it will be clear for all participants, and the election can be done again. If P differs from 1, then some nodes are announced themselves to be cluster aggregators, and this is known by all the nodes.

If we consider the effect of the second step (new election is run if no aggregator is elected), the expected number of aggregators is slightly higher than in the case of binomial distributions. The expected number of aggregators (c_E) are:

$$c_E = \frac{Np}{1-(1-p)^N}$$

The anonymity of the election subprotocol depends on the parts of the protocol. Obviously, the random number generation does not leak any information about the identity of the aggregator nodes, if the random number generator is secure. A cryptographically secure random number generator, called TinyRNG, is proposed in [FC07] for wireless sensor networks. Using a secure random number generator, it is unpredictable, who elects itself to be aggregator node.

The anonymity analysis of the anonym veto protocol can be found in [HZ06]. The anonymity is based on the decisional Diffie-Hellman assumption, which is considered to be a hard problem.

The message complexity of the election is $O(N^2)$, which is acceptable as the election is run infrequently (N is the number of nodes in the cluster).

As a summary, after the election subprotocol every node is equiprobably aggregator node. The election subprotocol ensures that at least one aggregator is elected and this node(s) is aware of its status. An outside attacker does not know the identity of the aggregators or even the actual number of the elected aggregator nodes. An attacker, who compromised one or more nodes, can decide whether the compromised nodes are aggregators, but cannot be certain about the other nodes.

The special usage of these anonym aggregators is described in the following:

THESIS 3.3: *I propose an anonym aggregation and query scheme, which can be used with the complex anonym election scheme in the presence of a compromising attacker.*

The data aggregation and storage procedure use the broadcast channel. If the covered area is so small or the radio range is so large that every node can reach each other directly, then the aggregation can be implemented simply. Every node broadcasts their measurement to the common channel, and the cluster aggregator(s) can aggregate and store the measurements. If the covered area is bigger (which is the more realistic case), a connected dominating set based solution is proposed.

In each timeslot, each ordinary node (not member of the CDS) sends its measurement to one neighboring CDS member (to the parent) by unicast communication. When the epoch is elapsed and all the measurements from the nodes are received, the CDS nodes aggregate the measurements and use a modification of the Echo algorithm [Cha06] on the given spanning tree to compute the gross aggregated measurement in the following way: Each CDS member waits until all but one CDS neighbor sends its subaggregate to it, and after some random delay it sends the aggregate to the remaining neighbor. This means that the leaf nodes of the tree start the communication, and then the communication wave is propagated towards the root of the spanning tree. This behavior is the same as the second phase of the Echo algorithm. When one node receives the subaggregates from all of its neighbors, thus cannot send it to anyone, it can compute the gross aggregated value of the network. Then, this value is distributed between the cluster members by broadcasting it every CDS member.

This second phase is needed, so that every member of the cluster can be aware of the gross aggregated value, and the anonymous aggregators can store it, while the others can simply discard it. The stored data includes the timeslot in which the aggregate was computed, and the environmental variables if more than one variable (e.g. temperature and humidity) are recorded besides the value itself.

The aggregation function can be any statistical function of the measured data. Some easily implementable and widely used functions are the minimum, maximum, sum or average. In Figure 15, the aggregation protocol is visualized with five nodes and two aggregators using the average as an aggregation function.

The anonymity analysis of the aggregation subprotocol is quite simple. After the aggregation, every node possesses the same information as an external attacker can get. This information is the aggregated data itself, without knowing anything about the identity of the aggregators.

The message complexity of the aggregation is $O(N)$, where N is the number of nodes in the cluster. This is the best complexity achievable, because to store all the measurements by a single aggregator, all nodes must send the measurements towards the aggregator, which leads to $O(N)$ message complexity. In terms of latency, the advanced protocol doubles the time the aggregated measurement arrives to the aggregator compared to a naive system, where the identity of the aggregators are known to every participant. This latency is acceptable as in most WSN applications the time between the measurements is much longer than the time required to aggregate the data.

As a first step, the operator authenticates itself to the selected node O using the key k_O. After that, node O starts the query protocol by sending out a query, obtains the response to the query from the cluster, and makes the response available to the operator. In the following, it is assumed that O is not a CDS node. (If it is indeed a CDS node, then the first and last transmission of the query protocol can be omitted.)

Node O broadcasts the query data Q with the help of the CDS nodes in the cluster. This is done by sending Q to the CDS parent, and then every CDS member rebroadcasts Q as it is received. The query Q describes what information the operator is interested in. It includes a variable name, a time interval, and a field for collecting the response to the query.

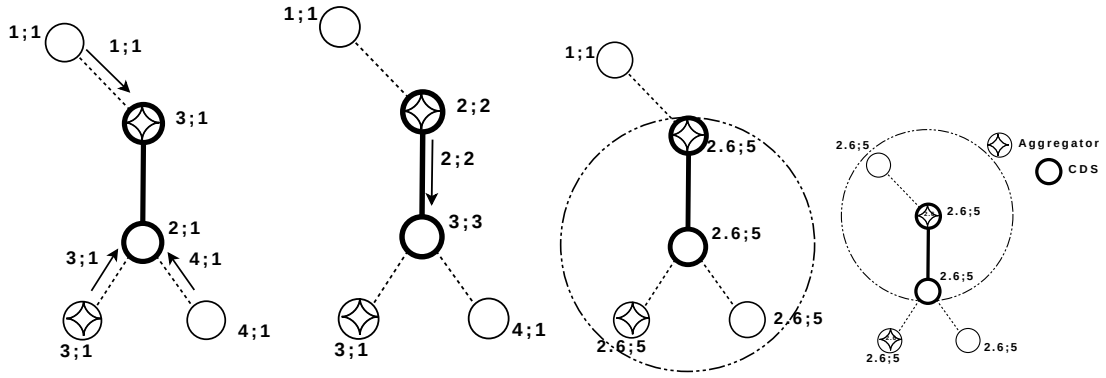

Figure 15: Aggregation example. The subfigures from left to right represents the consecutive steps of an average computation: (i) The measured data is ready to send. It is stored in a format of [actual average; number] of data. Non CDS nodes sends the average to their parents. (ii) The CDS nodes start to send the aggregated value to its parents. (iii) A CDS node receives an aggregate from all of its neighbors, and starts to broadcast the final aggregated value. Nodes willing to store the value can do so. (iv) Other CDS nodes receiving the final value rebroadcasts it. Nodes willing to store the value can do so.

The idea of the query protocol is that each node i in the cluster contributes to the response by a number R_i, which is computed as follows:

$$R_i = \begin{cases} h(Q|k_i), \text{ for non-aggregators} \\ h(Q|k_i) + M, \text{ for aggregators} \end{cases} \tag{15}$$

where M is the stored measurement (available only if the node is an aggregator), h is a cryptographic hash function, and k_i is the key shared by node i and the operator. Thus, non-aggregators contribute with a pseudo-random number $h(Q|k_i)$ computed from the query and the key k_i, which can later be also computed by the operator, while aggregator nodes contribute with the sum of a pseudo-random number and the requested measurement data. The sum is normal fix point addition, which can overflow if the hash is a large value.

The goal is that the querying node O receives back the sum of all these R_i values. For this reason, when the query Q is received by a non CDS node from its CDS parent, it computes its R_i value and sends it back to the CDS parent in the response field of the query token. When a CDS parent receives back the query tokens with the updated response field from its children, it computes the sum of the received R_i values and its own, and after inserting the identifiers of the nodes sends the result back to its parent. This is repeated until the query token reaches back to the CDS parent of node O, which can forward the response $R = \sum R_i$ and the list of responding nodes to node O, where the sum is computed by normal fix point addition. This operation is illustrated in Figure 16.

When receiving R from O, the operator can calculate the stored data as follows. First of all, the operator can regenerate each hash value $h(Q|k_i)$, because it stores (or can compute from a master key on-the-fly) each key k_i, and it knows the original query data Q. The operator can subtract the hash values from R (note that the responding nodes list is present in the response), and it gets a result $R' = cM$, where c is the actual number of aggregators in the cluster. Note that each aggregator contributed the measurement M to the response, that is why at the end, the response will be c times M, where c is the number of aggregators. Unfortunately, this number c is unknown to the operator, as it is unknown to everybody else. Nevertheless, if M is restricted to lie in an interval $[A, B]$ such that the intervals $[iA, iB]$ for $i = 1, 2, \ldots, N$ are non-overlapping,

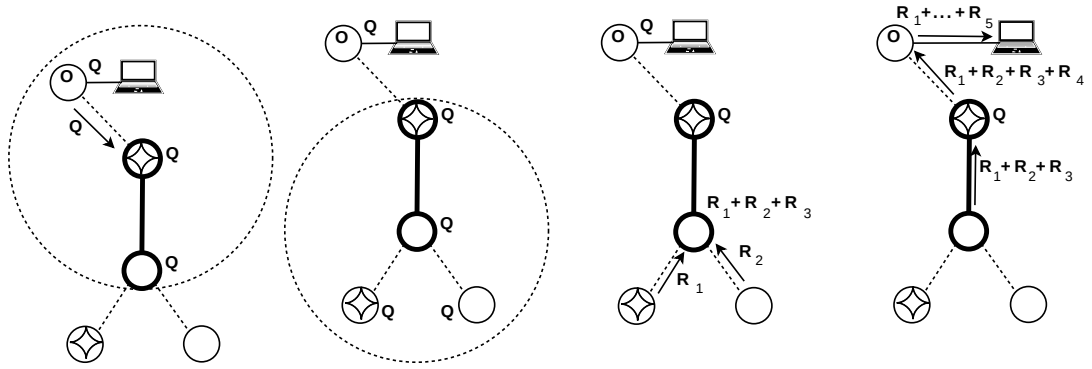

Figure 16: Query example. The subfigures from left to right represents the consecutive steps of a query: (i) The operator sends the Q query to node O. This node forwards it to its CDS parent. The CDS parent broadcasts the query. (ii) The CDS nodes broadcasts the query, so every node in the network is aware of Q. (iii) Every non CDS node (except O) sends it response to its parent. (iv) The sum of the responses is propagated back to the parent of O (including the list of responding nodes, not on the figure), who forwards it to the operator through O.

then cM can fall only into interval $[cA, cB]$, and hence, c can be uniquely determined by the operator by checking which interval R' belongs to. Then, dividing R' with c gives the requested data M. The detailed description how to find such an interval can be found in the dissertation.

The proposed protocol has many advantageous properties. First, the network can respond to a query if at least one aggregator can successfully participate in the subprotocol. Second, the operator does not need to know the identity of the aggregators, thus even the operator cannot leak that information accidentally (although, after receiving the response, the operator learns the actual number of the aggregator nodes). Third, the protocol does not leak any information about the identity of the aggregators: an attacker can eavesdrop the query information Q, and the R_i pseudo random numbers, but cannot deduce from them the identity of the aggregators. Finally, the message complexity of the query is $O(N)$, where N is the number of nodes in the cluster. This is the best complexity achievable, when the originator of the query does not know the identity of the aggregator(s). The latency of the query protocol depends on the longest path of the network rooted at node O.

The related publications are [C2, C3, J1, J3].

5 Application of New Results

In this theses three different wireless network based systems are considered: Radio Frequency Identification Systems, Vehicular Ad Hoc Networks, and Wireless Sensor Networks. In this chapter, the application of new results is described.

5.1 Radio Frequency Identification Systems

The application of RFID is very widespread, some application areas and many currently used and foreseen applications can be found in [WNYD09, RFI12], like: payment by mobile phones, inventory systems, access control.

Any usage of RFID systems, where the holder of the tag is a human being might breach the privacy of the holder. The solutions proposed can be used in such situations. An example application is the automated fare collection systems, where the pass for the mass transportation system can contain a RFID tag. In such a system, the system designer might consider the usage of key-trees or group based private authentication, in particular if the legal environment requires the usage of some kind of privacy enhancing technology. Some of the results regarding private authentication were delivered to joint projects with Mobile Innovation Center, Hungary[1].

5.2 Vehicular Ad Hoc Networks

The application of Vehicular Ad Hoc Networks is very widespread, but can be categorized into three main categories: safety related applications, transport efficiency, and information/entertainment applications [HL08, WTM09].

Most of the safety and traffic efficiency related applications are based on the beacon messages, which are frequent messages containing the location, heading, identifier, and some other attributes of the vehicle. These messages can enable the tracking of individual vehicles, which is a undesirable side effect of the usage of VANETs. This side effect is analyzed in the dissertation, and a countermeasure is proposed as well. This countermeasure algorithm is compatible with the framework proposed by the Car 2 Car Communication Consortium [Con12].

Most of VANET related results of my dissertation were parts of the result of the SeVeCom[2] European Commission funded project. The results were delivered to and accepted by the European Commission.

5.3 Wireless Sensor Networks

Wireless sensor networks can be used in many scenarios. In the following, some applications are given based on [ASSC02] with a special attention on the possible need of hiding some special nodes: military applications, critical infrastructure protection.

In the above mentioned applications, aggregation might be needed, and the loss of the aggregator might have undesirable consequences. Hence in these applications, the anonym aggregator election, aggregation, and query schemes proposed in my dissertation can be used. Such an application is the critical infrastructure protection with wireless sensor nodes. This was the goal of the Wireless Sensor and Actuator Networks for Critical Infrastructure Protection project(WSAN4CIP[3]) funded by the Eropean Comission, where the results were delivered.

[1] `http://www.mik.bme.hu//`
[2] `http://www.sevecom.org/`
[3] `http://www.wsan4cip.eu`

References

[ADO05] G. Avoine, E. Dysli, and P. Oechslin. Reducing time complexity in rfid systems. In *Proceedings of the 12th Annual Workshop on Selected Areas in Cryptography (SAC'05)*, pages 291–306. Springer, 2005.

[ASSC02] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[Cha06] E.J.H. Chang. Echo algorithms: Depth parallel operations on general graphs. *Software Engineering, IEEE Transactions on*, (4):391–401, 2006.

[Con12] Car 2 Car Communication Consortium. "`http://www.car-to-car.org`", 2012.

[FC07] Aurélien Francillon and Claude Castelluccia. TinyRNG: A cryptographic random number generator for wireless sensors network nodes. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, pages 1–7, April 2007.

[HL08] H. Hartenstein and K.P. Laberteaux. A tutorial survey on vehicular ad hoc networks. *Communications Magazine, IEEE*, 46(6):164 –171, June 2008.

[HZ06] F. Hao and P. Zielinski. A 2-round anonymous veto protocol. In *Proceedings of the 14th International Workshop on Security Protocols, Cambridge, UK*, 2006.

[KHRW02] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. Sumo (simulation of urban mobility); an open-source traffic simulation. In A Al-Akaidi, editor, *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM2002)*, pages 183–187, Sharjah, United Arab Emirates, September 2002. SCS European Publishing House.

[MW04] D. Molnar and D. Wagner. Privacy and security in library rfid: Issues, practices, and architectures. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 210–219. ACM, 2004.

[RFI12] Wikipedia RFID. Radio-frequency identification. "`http://en.wikipedia.org/wiki/Radio-frequency_identification`", 2012.

[WNYD09] D.L. Wu, W.W.Y. Ng, D.S. Yeung, and H.L. Ding. A brief survey on current rfid applications. In *Machine Learning and Cybernetics, 2009 International Conference on*, volume 4, pages 2330–2335. IEEE, 2009.

[WTM09] T.L. Willke, P. Tientrakool, and N.F. Maxemchuk. A survey of inter-vehicle communication protocols and their applications. *Communications Surveys Tutorials, IEEE*, 11(2):3 –20, quarter 2009.

6 Publication of New Results

International Journal Papers

[J1] T. Holczer and L. Buttyán. Anonymous aggregator election and data aggregation in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 18 pages, 2011. Article ID 828414.

[J2] P. Papadimitratos, A. Kung, F. Kargl, Z. Ma, M. Raya, J. Freudiger, E. Schoch, T. Holczer, L. Buttyán, and J-P. Hubaux. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):100–109, 2008.

109 independent citations

[J3] P. Schaffer, K. Farkas, Á. Horváth, T. Holczer, and L. Buttyán. Secure and reliable clustering in wireless sensor networks: A critical survey. *Computer Networks*, 2012.

International Conference and Workshop Papers

[C1] G. Avoine, L. Buttyan, T. Holczer, and I. Vajda. Group-based private authentication. In *Proceedings of the International Workshop on Trust, Security, and Privacy for Ubiquitous Computing (TSPUC 2007)*. IEEE, 2007.

16 independent citations

[C2] L. Buttyán and T. Holczer. Private cluster head election in wireless sensor networks. In *Proceedings of the Fifth IEEE International Workshop on Wireless and Sensor Networks Security (WSNS 2009)*, pages 1048–1053. IEEE, IEEE, 2009.

[C3] L. Buttyán and T. Holczer. Perfectly anonymous data aggregation in wireless sensor networks. In *Proceedings of The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (WSNS 2010)*, San Francisco, November 2010. IEEE.

[C4] L. Buttyan, T. Holczer, and P. Schaffer. Spontaneous cooperation in multi-domain sensor networks. In *Proceedings of the 2nd European Workshop on Security and Privacy in Ad-hoc and Sensor Networks (ESAS)*, Visegrád, Hungary, July 2005. Springer.

8 independent citations

[C5] L. Buttyan, T. Holczer, and I. Vajda. Optimal key-trees for tree-based private authentication. In *Proceedings of the International Workshop on Privacy Enhancing Technologies (PET)*, June 2006. Springer.

39 independent citations

[C6] L. Buttyan, T. Holczer, and I. Vajda. Providing location privacy in automated fare collection systems. In *Proceedings of the 15th IST Mobile and Wireless Communication Summit, Mykonos, Greece*, June 2006.

[C7] L. Buttyan, T. Holczer, and I. Vajda. On the effectiveness of changing pseudonyms to provide location privacy in vanets. In *Proceedings of the Fourth European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS2007)*. Springer, 2007.

71 independent citations

[C8] L. Buttyan, T. Holczer, A. Weimerskirch, and W. Whyte. Slow: A practical pseudonym changing scheme for location privacy in vanets. In *Proceedings of the IEEE Vehicular Networking Conference*, pages 1–8. IEEE, IEEE, October 2009.

6 independent citations

[C9] L. Dora and T. Holczer. Hide-and-lie: Enhancing application-level privacy in opportunistic networks. In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking ACM/SIGMOBILE MobiOpp 2010*, Pisa, Italy, February 22-23 2010.

1 independent citations

[C10] A. Dvir, T. Holczer, and L. Buttyán. VeRA - version number and rank authentication in rpl. In *Proceedings of the 7th IEEE International Workshop on Wireless and Sensor Networks Security (WSNS 2011)*. IEEE, 2011.

National Journal Papers

[N1] L. Buttyan, T. Holczer, and P. Schaffer. Incentives for cooperation in multi-hop wireless networks. *Híradástechnika*, LIX(3):30–34, March 2004. (in Hungarian).

Other

[O1] T. Holczer, P. Ardelean, N. Asaj, S. Cosenza, M. Müter, A. Held, B. Wiedersheim, P. Papadimitratos, F. Kargl, and D. D. Cock. Secure vehicle communication (sevecom). Demonstration. Mobisys, June 2009.