A Security–Energy Trade-Off for Authentic Aggregation in Sensor Networks

Erik-Oliver Blaß Joachim Wilke Martina Zitterbart
Institute of Telematics, University of Karlsruhe, Germany
Email: {blass,wilke,zit}@tm.uka.de

***Abstract*— To reduce energy consumption, aggregation takes place in a wireless sensor network. All measured data is collected and preprocessed multiple times on its way towards a data sink, e.g., a base station. However, aggregation implies new challenges to security: as the sink finally receives aggregated data, it is difficult to verify not only the aggregate's correctness, but also the origin of the data the aggregate was computed from. In the presence of an attacker in the network, data transmissions and aggregation could have maliciously been modified. Yet, it turns out that in-network aggregation and data authenticity are contradictory communication properties. This research examines the possibility of finding a *trade-off* between security (authenticity) and energy-savings (aggregation). If the user is willing to accept data's authenticity with $p \leq 100\%$ probability, he can still save large amounts of energy compared to authentic communication without aggregation.**

I. Introduction

Data transport in wireless sensor networks follows a new communication paradigm: aggregation. Sensors report their measurements, e.g., temperature, towards a data sink. Typically, on the way to the sink, data is aggregated by aggregation nodes. More precisely, aggregation nodes collect measurements from other nodes and preprocess them before sending the resulting *aggregate* further towards the data sink. An example is shown in Figure 1. Sensor nodes a and b measure the room temperature of room 1, e.g., at the ceiling and at the floor. The sink, represented as a laptop in Figure 1, is, however, only interested in the mean temperature of the whole building. Therefore, a and b send their measured temperatures to aggregation node x, which can compute the mean temperature of room 1 and sends this aggregate to aggregation node z. Nodes c and d do the same for room 2: they send their data to node y for aggregation and forwarding to node z. Finally, node z computes the mean temperature of the whole building and sends it to the sink.

Instead of sending all measured values to the sink without any in-network preprocessing and merging, aggregation greatly reduces the volume of transported data, the number of data transmissions and therefore saves valuable energy.

Yet, if the measured data is very sensitive and important, data transport has to be secured. Otherwise, an attacker could illegally read secret measurements or modify them. It might be even possible for an attacker to compromise nodes, i.e., to read out all their secrets and to completely take control of them. These compromised nodes might behave like regular nodes, but maliciously modify important data of other nodes. For example, a compromised aggregation node x might intentionally compute a false aggregate and forward it to z. If both nodes a and b would report a very high room temperature of 100°C, e.g., because of a fire in room 1, compromised node x could ignore this information and compute a save mean temperature of 20°C. As a result, aggregation node z and finally the sink would not be aware of a dangerously risen temperature in the building. In addition, even without node compromise, an attacker could try to masquerade as node x and inject wrong aggregates into the network, for example to z.

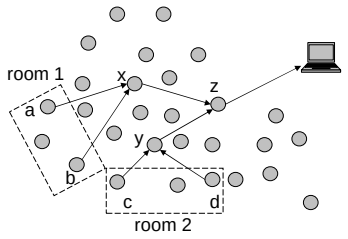

Fig. 1. Data aggregation example

Hence, security in an aggregating sensor network does not only imply confidentiality of transported data, but also the correctness of aggregation as well as the authenticity or originality of participating (aggregation) nodes. Not only direct node-to-node authenticity, e.g., between nodes a and z, is of importance, but also *end-to-end* authenticity, e.g., between nodes a and z or a and the sink. The sink and all *intermediate* aggregation nodes like z have to verify their received aggregates for correctness and authenticity in the presence of one or more malicious nodes.

This research examines the implications of aggregation on (end-to-end) authenticity in the presence of multiple malicious nodes. First of all, aggregation turns out to be contradictory to authenticity. Therefore, a security-energy trade-off called *ESAWN*, *Extended Secure Aggregation for Wireless sensor Networks*, is proposed. On the one hand, the user can choose to save more energy by aggregation, but has to accept "weaker" authenticity for his data. On the other hand, if the user wishes more secure data, he has to spend more energy. Weaker authenticity means that the user can not always expect data to be 100% authentic, but gradually less authentic, i.e., only $p \leq 100\%$ authentic. ESAWN copes with multiple malicious nodes, supports arbitrary aggregation functions, and does not rely on any central infrastructure.

The rest of this proposal is structured as follows: After a brief overview of related work in Section II, Section III

describes the implications of aggregating data transport on authenticity in a sensor network. Section IV introduces ESAWN, a protocol for a Security-Energy trade-off, giving an user the ability to parameterize security and energy based on his demands. Also, first results of ESAWN's implementation are presented. Section V concludes this proposal.

II. Related Work

Compared to simple node-to-node authenticity, which can be provided by any key establishment scheme like [1], real end-to-end authenticity is a new field of research in aggregating sensor networks. Only a few papers have been published which suffer from a lot of drawbacks: for example, schemes based on *privacy-homomorphism* as [2] limit aggregation functions to only trivial mathematical computations. More complex aggregation is impossible with this approach, it makes frequent use of computationally, energy expensive public-key cryptography and has been proven to be insecure. The same applies for [3]: only rudimentary aggregation functions, e.g., the computation of a median, are supported. Among other things, comparison of data, as part of a more complex aggregation function, is impossible. As another disadvantage, this scheme requires a secure broadcast protocol.

III. Aggregation vs. Authenticity

Interestingly, authenticity and aggregation are contradictory communication paradigms: an aggregation node x takes inputs from multiple sensor nodes, e.g., measurement A from a and B from b, and computes an aggregate $f(A, B)$. This aggregate is further forwarded towards the sink. Receiving node z can verify the (node-to-node) authenticity of aggregate $f(A, B)$ as coming from x easily, because it might know a pairwise secret key shared between x and z. However, verifying the correctness of $f(A, B)$ and its authenticity regarding nodes a and b is difficult. As aggregation functions f can be of arbitrary complexity, $f(A, B)$ might reveal absolutely no information regarding the involved data from a and b – imagine f being a one-way hash function. Thus, it is *impossible* for a node z or the sink, only receiving an aggregate, to verify whether $f(A, B)$ has been computed in a correct way and whether a and b have been the responsible contributing nodes for $f(A, B)$.

To be able to verify $f(A, B)$'s correctness and authenticity, z would need knowledge about measurements A and B as well as about the originating nodes a and b. Obviously, if a and b would forward their measurements to node z, z could easily recompute $f(A, B)$. However, this foils the general idea of aggregation. Directly sending all measurements to z would be same as no aggregation. The total amount of data to be transported and the number of packets would not be reduced, no energy would be saved due to the aggregation.

If the user wants authenticity for his transported data, he has to spend energy for it, which in return means (at least partial) cancellation of possible energy savings due to aggregation. Still, there is the chance of balancing aggregation and authenticity by finding a security-energy trade-off. This is the general principle behind the proposed protocol ESAWN.

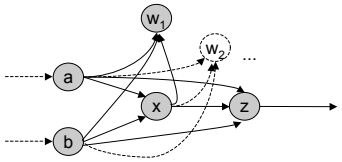

Fig. 2. Using witnesses in ESAWN

IV. ESAWN

For ESAWN, the following assumptions are made:

1.) ESAWN assumes a-priori distributed pairwise secret keys between all nodes who are going to communicate which each other. For example, nodes a and x or a and z share pairwise keys with each other. This can be implemented efficiently by a key distribution protocol, e.g., [1].

2.) The attacker is assumed to be in the position to compromise b-percent of the whole network. For instance, if the network consists of $n = 5.000$ nodes and $b = 10\%$, the attacker has compromised 500 nodes. Furthermore, in every current verification step of the ESAWN-protocol, not all n nodes, but only a subset of $n' \leq n$ nodes are involved. As it is assumed that the attacker compromises the nodes *uniformly distributed*, b-percent of the protocols participating n' nodes are compromised. For example, if $b = 10\%$ and a total of $n' = 10$ nodes involved in ESAWN's *current* verification, there would be $k = 1$ compromised node ESAWN would have to deal with.

3.) ESAWN does not provide authenticity at the same level as one is typically used to, i.e, data is authentic or data is *not* authentic, but ESAWN only assures *gradual* data authenticity. Gradual authenticity means that data is authentic only within a certain probability $p \leq 100\%$ in the presence of b-percent compromised nodes or k compromised nodes taking part in the current verification step, respectively.

4.) Verification works inductively. According to Figure 1, aggregations of aggregation nodes x and y are verified first. After their successful verification, node z's aggregation is verified. So, the induction hypothesis for a current verification is that all contributing aggregations for this aggregation have been successfully verified and are secure in the presence of k compromised nodes.

A. Protocol Description

To verify aggregations, ESAWN uses *witnesses*. Given a situation somewhere in a sensor network, aggregation node x aggregates measurements A from a and B from b and sends the resulting aggregate $f(A, B)$ to z. This is shown in Figure 2. Note that in this case, nodes a and b could also be aggregation nodes.

To verify x's possibly malicious aggregation as well as a's and b's authenticity in the presence of k malicious nodes, a total of k witnesses $w_1, \ldots, w_k$ have to be utilized. The witnesses are randomly chosen nodes in the direct physical neighborhood of x. Nodes a and b send their measurements A, B, or aggregates if they are aggregation nodes, not only

to x but also to nodes z and $w_1, \ldots, w_k$. The measurements are encrypted with pairwise keys shared between a or b and z, $w_1, \ldots, w_k$. Furthermore, aggregation node x sends its encrypted aggregate $f(A, B)$ not only to z, but also to all k witnesses $w_1, \ldots, w_k$ (cf. Figure 2).

Now, all witnesses w_i and node z can verify the *correctness* of received $f(A, B)$ by re-computing f using A and B from nodes a and b. As all data is encrypted using pairwise keys, data is authentically transported between any two nodes, i.e., node-to-node authentically. In addition, using the induction hypothesis, data has been transported end-to-end authentically between nodes a, b and node z, if no more than k nodes are compromised out of $\{x, z, w_1, \ldots, w_k\}$. No data forgery can take place unnoticeably. Using this scheme, all aggregations in the sensor network are verified. Finally, after verifying the "last" aggregate, the sink can be sure about its correctness and end-to-end authenticity in the presence of up to b-percent of compromised nodes in the network.

It is quite obvious that these replication of a's and b's data to z and all witnesses w_i requires a lot of additional data transmission, wastes energy and therefore partially levels out energy savings due to aggregation. Therefore, the user can selectively choose not to verify every aggregation in the network all the time, but to probabilistically verify aggregations. The user can adjust the verification rate to be $p \leq 100\%$. Thus, every time an aggregate is computed, ESAWN verifies its correctness and authenticity only with a probability of p-percent. This greatly reduces verification energy costs by $1 - p$ percent, but, of course, also reduces the security: If the aggregation node, which will not be verified, is incidentally malicious, it might unnoticeably forge its aggregate. Data is correct and authentic only with a probability of p percent.

B. Performance Evaluation

To measure the performance and impact on energy consumption of ESAWN, a first implementation has been programmed using the simulator GloMoSim.

Preliminary results can be seen in Figure 3. Varying the total number of nodes in the network between $n = 1000$ and $n = 10000$ nodes, random sensor networks have been created. The logarithmically scaled y-axis shows the number of packets, i.e., data transmissions, necessary for different ESAWN-configurations to transport all measurements starting from the measuring sensor nodes towards the sink. However, all curves in Figure 3 are printed in relation to a baseline, namely the top most, horizontal curve at 100%. This curve represents the number of packets required for a secure data transport *not* using ESAWN and any aggregation. This would be data transport, where the aggregation nodes would not do any aggregation but only forward received data further towards the sink. All aggregation would take place at the sink. Implicit for the sink, this would be correct "aggregation". Also, as all measurements would be encrypted for the sink from the measuring sensor, this would be authentic data transport. All other curves in Figure 3 are shown in relation to this baseline. For example, an ESAWN curve with a certain y-value of 60% at an x-value of 5.000 nodes would indicate that only 60% of the packets are necessary compared to the no-aggregation approach. The lowermost curve represent aggregation without any security, i.e., $k = 0, p = 0$. Basically, this is the energy savings possible using aggregation. Neither ESAWN nor any other security protocol, can be below this curve and save more energy. All curves in Figure 3 fall, because the relative impact of aggregation on energy savings slowly grows with the total number of nodes in the network. As you can see, an ESAWN-configuration guaranteeing $p = 100\%$ authenticity and correctness while protecting against $k = 3$ compromised nodes taking part in each verification is more expensive than, e.g., $k = 2, p = 100\%$.

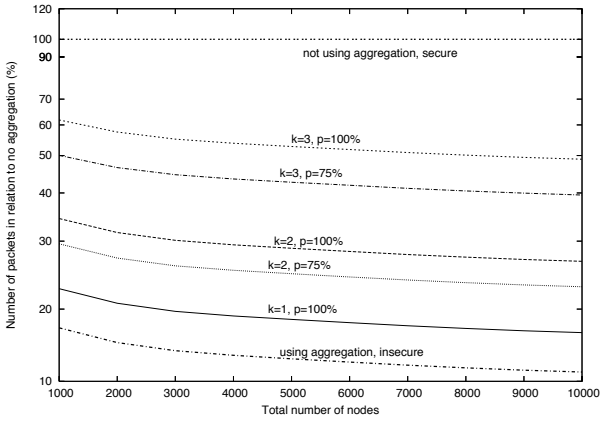

Fig. 3. Security-Energy trade-off

Typically, the user can estimate b in advance and therefore judge the value of k for every verification. On the one hand, it is now possible for the user to calculate the energy costs according to his security demand p and decide whether he can afford these costs. On the other hand, the user can see the security p he will get for spending a certain amount of energy for ESAWN and decide whether this is sufficient.

V. Conclusion

This research discusses a contradiction between end-to-end authenticity, data correctness and aggregation. A first protocol, ESAWN, is proposed, which implements a *security–energy trade-off*. The user can find a balance of spending a certain amount of energy to get a certain level of security in return. ESAWN is able to cope with multiple compromised nodes, forgoes central infrastructures and supports arbitrary aggregation functions. Current work investigates the relation between security of individual aggregations as described herein and the resulting combined security of multiple concatenated aggregations, e.g., the security of measuring nodes and the sink using multiple aggregations in between.

References

[1] L. Eschenauer and V. Gligor, "A key management scheme for distributed sensor networks," in *ACM Computer and Communications Security*, 2002.

[2] E. Mykletun and J. Girao, "Public key based cryptoschemes for data concealment in wireless sensor network," in *IEEE International Conference on Communications ICC*, 2006.

[3] B. Przydatek, D. Song, and A. Perrig, "Sia: secure information aggregation in sensor networks," in *ACM International conference on Embedded networked sensor systems SenSys*, 2003.