Analysis of the Authenticated Query Flooding Protocol by Probabilistic Means

Peter H. Schmitt, Frank Werner *
Universität Karlsruhe (TH)
Institut für Theoretische Informatik
{pschmitt,werner}@ira.uka.de

ABSTRACT

Secure multicast communication is a elementary mechanism in the field of wireless sensor networks, addressing a number of security means and algorithms. This paper analyses the energy consumption of the algorithm of authenticated query flooding as proposed by [1] but the applied technique can also be used to a more general probabilistic flooding paradigm. The verification results are obtained by means of the probabilistic model checking tool PRISM. We measure in our analysis the impact if only a portion of all packets stem from an adversary that needs to be hindered from reaching the whole network. What is new when choosing the concept of formal methods and what differs it from the simulation based analysis is that the results are precise and need no confidence level since all paths which lead to an observation are consequently expressed in an expectation value rather than an on-average value with confidence levels.

1. INTRODUCTION

Wireless sensor networks form a very active research topic in Computer Science with a variety of novel applications in a great numbers of diverse areas. For potential applications that transmit valuable or critical data security issues will play an important role. This has triggered investigations into how wireless sensor networks – which are particularly vulnerable to intrusion by an outside attacker – can be protected against malicious or accidental manipulations.

Communication in wireless sensor networks is characterised by the fact that no centralised knowledge about the identity, reachability, location or functionality of the individual nodes is available. Given these constraints *flooding* has become an accepted communication paradigm despite its high energy cost.

We will concentrate in this paper on the scenario that a base station, e.g., a laptop-like device spreads information by flooding authenticated queries to a wireless sensor network. An adversary may post illegitimate or fake queries disrupting or compromising the network. A probabilistic authentication protocol for *authenticated query flooding* has been proposed in [1] to limit the propagation of fake queries. The influence of various protocol parameters on the strictness of this limitation has been investigated by analytical computations and network simulations. In this paper we complement this analysis by estimations on average energy consumption under various parameter settings. This should provide us with valuable information in which situations an implementation of this algorithms is feasible in practice.

We use the PRISM tool – a probabilistic symbolic model checker [2, 5] – for our investigations. This might not be the tool that immediately suggests itself for the job at hand and we do believe that a comparable analysis could have been accomplished by a state-of-the-art simulation tool like ns2 [6]. At this point it is important to highlight, that we do not obtain the presented results by simulation. Rather the values are computed by verification and the inherent exploration of the complete state space, i.e. all possible interleavings of all runs.

What made PRISM attractive from our point of view is its solid foundations on the theory of Markov chains and its comfortable graphical user interface. There is no denying that the size of the Markov chain models accessible to analysis by PRISM is somewhat restrited in the number of module instances that can be carried out simultaneously.

This work is organised as follows. In section 2 we quickly review *authenticated query flooding* from [1], giving some idea of the algorithm and its main parameters. In section 3 we analyse the models with the PRISM tool [5] and explain the parameter influencing the model. Exemplary scenarios are considered and investigated i.e. probabilistic results like *"How much energy is used by reaching N nodes with a faked query?"*. This part also tries to bridge the gap to an practical energy critical example as quested by the ZeuS project where energy and gradual security is considered a central topic. The section is closed by giving an optimal security/energy tradeoff depending on an adversary, and the expected times until the network is flooded. In the end in section 5 the whole matter is wrapped up, giving an outlook for possible improvements and finally concluding with suggestions for further energy related exploration.

2. THE AUTHENTICATED QUERY FLOODING ALGORITHM

The authenticated query flooding (AQF) algorithm proposed in [1] assumes an ID-based key predistribution, see e.g.,[8]. Out of a pool of keys numbered from 1 to ℓ every node receives a ring of k randomly chosen keys. The way this predistribution may be organised is sketched in [1]. When the base station wants to flood a query q it first computes

*This research position is funded by the BW-FIT Project ZeuS (Zuverlssige Informationsbereitstellung in energiebewussten ubiquitren Systemen).

the value $x = h(q)$ of some given hash function h and then uses x as a seed to compute m pseudo random numbers $(kid_1, \ldots, kid_m)$. These numbers are interpreted as the numbers of keys $(k_{kid_1}, \ldots, k_{kid_m})$ from the pool. These keys are used to compute m message authentication codes (MACs). We stick to the design decision from [1] to use 1-bit MACs. Thus, using key k_{kid_i} on $x = h(q)$ the bit m_i is computed. The sequence $(m_1, \ldots, m_k) = macs(q)$ is called the *authenticator* for q. The base station then floods query q together with $macs(q)$ into the sensor network.

Upon receiving a query q and the authenticator $macs(q)$ a sensor node, which is assumed to share the hash function h and the pseudo random number generator with the base station, computes the indices $(kid_1, \ldots, kid_m)$ used to encode $macs(q)$. If for at least one of the keys k_{kid_i} that also belong to its own key ring it detects a mismatch between the computed and the received value m_i, it does not forward the query. In all other cases it sends it to all its neighbours. As in [1] we are only interested in the analysis of the flooding algorithm and ignore the situation when a sensor nodes considers a query as genuine and replies to it. Furthermore, the node memorises processed queries and immediately ignores them when receiving them for a second time. Obviously, a legitimate query q will be received by all reachable nodes in the network.

The adversary model adopted in [1] assumes that an attacker can feed messages plus authenticators into the network in the same way as the base station does. It is furthermore assumed that an adversary may *capture* sensor nodes and get hold of their keys. It thus may start flooding a query q using the correct MAC-bits for the keys it has captured, and randomly guesses the remaining authenticator bits. Assuming that an attacker has captured $\tilde{n}$ nodes using the theory of random sets the average number $\tilde{b}$ of keys known to the adversary and the expected value B of correct MAC-bits in a fake query authenticator can be computed. The probability p_f that a sensor forwards a query with a fake authenticator is according to [1] given by the formula:

$$p_f = \left(\frac{\ell - k}{\ell} + \frac{k}{\ell}\frac{B}{m}\right)^m = \left(\frac{\ell - k}{\ell} + \frac{k}{\ell}\frac{\tilde{b}}{m} + \frac{1}{2\ell}\right)^m$$

It is an essential contribution of [1] to suggest a criterion for choosing plausible values for p_f. Table 1 shows typical parameter values that we investigate with the corresponding probability p_f. In the experiments considered later in this paper we will use topologies with densities varying between 2.3 and 4.1 nodes and forwarding probabilities between 0% and 45% since the algorithm is working very efficient within this region.

Variable	Value range	Description
n	12 - 14	number of nodes in the sensor network
ℓ	1 000-10 000	number of keys in the key pool
k	50-250	number of keys in the key ring of a node
$keylen$	128	length of one key
m	100-500	size of the authenticator
MNKK	-	mean number of keys that a sensor has to validate per query
$data$	8	data bits
d	2-4	network density
$\tilde{n}$	≥ 1	Number of captured nodes
$\tilde{b}$	-	number of captured keys
$E_{\tilde{b}}$	-	number of keys in the authenticator known by the adversary
B	-	number of right bits in the fake authenticator
p_f	-	probability that the message will be forwarded

1: Annotation for the variable meanings and parameters for the AQF algorithm that contribute to the forwarding probability of p_f, and resulting energy. Blank fields depend on the setting and need individual computation.

3. ENERGY CONSUMPTION IN WSN

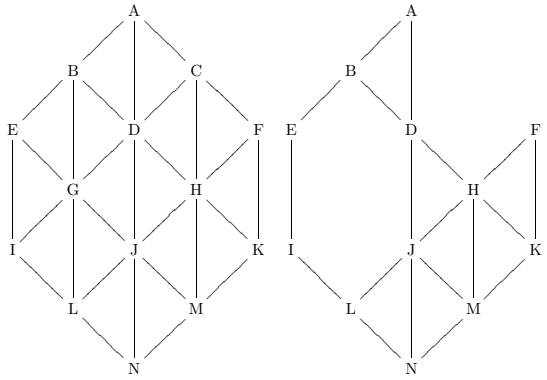
This section aims at providing scenarios and measures that can later on be beneficially used for building an actual wireless sensor network based on authenticated query flooding. Five different topologies of sensor networks (cf. Fig. 3) are introduced that will be analysed under the objective of power consumption until the request terminates. We assume legitimate and fake queries to be injected into the network at node A although any other point is feasible, and could be done in future work. Additionally an adversary might use 2 or more nodes to inject the faked query which is admissible but not covered here and kept as a possible future research topic.

In computing the energy balance all sensors from the network will be taken into account. The networks that we investigate are strongly *interlinked* and consist of up to 18 sensor nodes. The reason for picking symmetric topologies is that we think that results scale to even bigger network. In Figure 1a nodes have at most six neighbours. A variation of this network setup is topology 8 with two missing nodes (cf. Figure 1b). A *check-box-like topology* with 12 sensors is present in Figure 1c, where each sensor node has at most four neighbours. A *hexagon-like structure* presented in Figure 1d with each node having at most 3 proximate nodes. And finally the asymmetric topology from Figure 1e with a more realistic shape and 18 nodes will conclude the study.

By modelling these wireless sensor networks as a Discrete Time Markov Chain (DTMC) we do a reward analysis [4] within the probabilistic model checker PRISM. As such we use a sensor node specific energy consumption function to formulate the energy constraints within the model. The later analysis will reveal how parameters change when dealing with different topologies, and how the individual characteristics under the deployment of a probabilistic flooding manifest and can be compared.

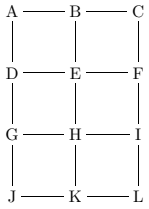
3.1 The Reward Model

The reward model is attaching costs to state transitions in a way that makes them computable for the prism tool. We consider a sensor receiving if it receives a packet by any of the gadgets in its vicinity. After reception it is computing and validating the 1-bit MACs for which it need energy

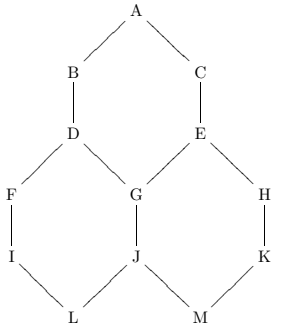


(a) Topology 7

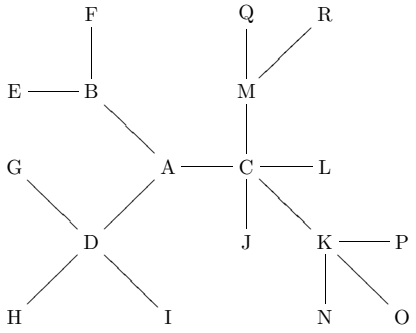
(b) Topology 8



(c) Topology 9



(d) Topology 10



(e) Topology 11

1: problem topologies

E(RC) including the powerrequirements for receiving. After the authenticator is computed the node does either attribute the query as non-fake, distributing it to vicinity nodes in which case a total energy draw of E(RCS) is needed. In case that the authenticator turns out to be faked, the packet is dropped without further transmission.

In the following analysis we use networks of TMote Sky sensors manufactures by Moteiv[7]. The gadgets will be arranged in topologies as illustrated in figure 3. This is rational since we expect to find an overview of how queries are processed using a probabilistic flooding mechanism, and to which extent the choice of the topology influences the security/energy ratio. The approach presented here tries to obtain a theoretical approximation of a real-world scenario by means of Markov chains with all the pros and limitations this involves.

As input parameters for our model (cf. Table 1) we use a 1 byte data packet. The key length is selected as 128 bits which seems to be a sound value considering an available memory of 8kB and the relatively high security level induced hereby. As varying parameters we choose the total number of keys in the keys pool ℓ between 1 000 and 10 000 appropriate for small networks, the size of the authenticator m between 100 and 500 and the number of keys with which the sensor nodes are preloaded, denoted as k to be within the range of 100 and 250. Underlying the model we assume that the adversary only knows the keys deployed on a single sensor node (k valid keys), since we have a small size network with devices more or less out of a burglar's reach. For approximating the average number of hashes a node has to validate, we use the figure *mean number of known keys* by a sensor (MNKK) which depends on the variable input parameters and is computed as:

$$MNKK = \frac{m \cdot k}{\ell}$$

As cryptographic hash function we use the MD2 (Message Digest Algorithm)[3] since it seems to be the best choice when dealing with 8-bit micro processors and a key length of 128 bit. Using these numbers we obtain a varying probability p_f (that a sensor accepts the query with a fake authenticator) which is used as an input parameter for our model. An example for varying the authenticator size m while ℓ, and k, are kept constant is illustrated in Table 3.1 below.

m	MNKK	pf	E(RC)	E(RCS)
-	-	-	0.1723	0.3355
100	1.6767	0.439	0.2574	0.4857
150	2.5	0.291	0.2999	0.5608
200	3.33	0.193	0.3425	0.6359
250	4.17	0.128	0.3850	0.7110
300	5	0.085	0.4276	0.7862
350	5.83	0.056	0.4701	0.8613
400	6.67	0.037	0.5127	0.9364
450	7.5	0.025	0.5552	1.0115
500	8.33	0.016	0.5978	1.0866

2: Power usage for the TMote Sky sensor node in mJ for receiving E(R), computing and comparing the hash values E(RC), and sending E(RCS) for a 8 bit data packet and a 128 bit keylength. The first line is without the AQF algorithm, for the remaining entries the total number of keys is fixed to $\ell = 6000$, the number of keys on each sensor is $k = 100$. Parameter MNKK is representing the *mean number of keys known* by a sensor node and the authenticator parameter m is varying.

For obtaining the expected energy use we finally question our model with the PCTL query as follows:

$$R =? \quad [\quad F \quad (\text{"}deadlock\text{"}) \quad]$$

Although the term "deadlock" might be misleading at this point, it is defining the appropriate state within our model, in which all queries are processed and no further step is possible. Whether is happens due to dropping of the queries

or due to the fact that the whole network is flooded needs no further specification here.

3.2 Energy Use

In the following we briefly sketch the energy requirements influencing the reward model of our sensor network. We expect the processor to run at 8MHz which corresponds to 6 million instructions per second. One byte data packet plus the size of the authenticator is assumed as being the payload.

The energy draw of the radio controller integrated in the board is assumed to be 5.9 mW for receiving, and 5.6 mW for sending which can be drawn out of the data sheets. The micro controller has a power need of 6 mW, resulting in 6 million operations per second (MIPS) total. For moving data from the radio controller to the CPU and vice versa an energy amount of 6.5 mW is needed, since transmission and CPU have to be switched on. All power needs relate to a packet of size 8 bit, a 128-bit key length, and need separate computation when increasing the authenticator size.

When looking at times that we need for computing the operation dependent power, we obtain for a simple reception of a packet of size 108 bit (100 bit authenticator and 8 bit data) the receiver needs 0.035 ms per byte, and in addition to 0.5 ms for initialisation, that sum up to 3.6475 ms per payload. Loading of the data from the radio controller to the CPU requires 0.05 ms per byte plus a constant time of 2 ms. For hashing the data 3 732 block instructions are required using the MD2 hash algorithm that runs in approximately 0.0075 ms. The hashes can be validated by the sensor node in 0.01859 ms. For transmission of the data, times equal to the times for receiving are needed.

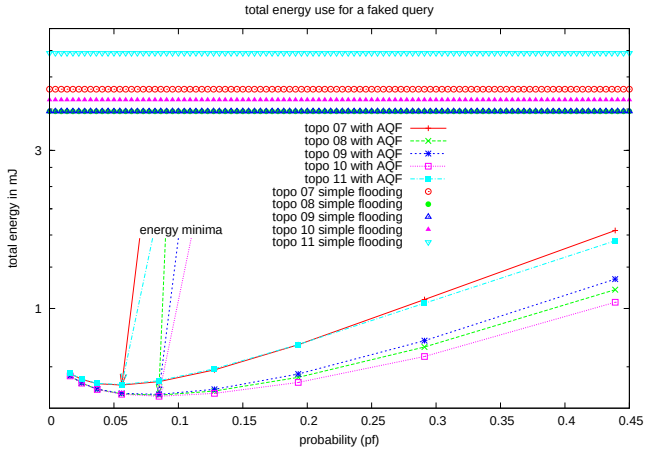
Some emerging energy figures that corelate with the increasing authenticator m are displayed in table 3.1 for the Tmote Sky board. All sensors are either in receive mode during the flooding procedure or try to authenticate the received query. Upon successful authentication, and in case of not being able to authenticate the data packet, they forward the query to neighbouring sensors. Since we want to limit the propagation of fake queries to a small part of the WSN, we consider a forwarding probability of faked packets p_f below 45%. For most topologies. In addition to figure p_f we compute the energy used without the AQF procedure, that is energy required for flooding a sound packet thru the network. During these experiments a query is received and spread to surrounding sensors without any involved computation.

4. ENERGY RESULTS

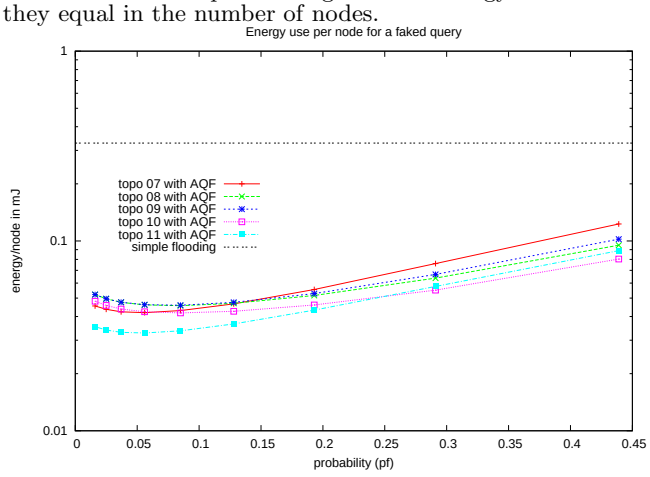
At this point it is necessary to point out, that when doing the reward rate computation the output for different settings are not easily to compare since the topologies vary e.g. in the network density, the total number of nodes in the network topology etc. To account for this, we compute in addition a correlating average rate per node, composed of the respective reward rate energy, and the number of nodes of the network. When doing so, we have in mind that sensor nodes do not power out evenly. Especially nodes close to the base station are expected to spend more power than the others, located further away.

p_f	m	TOP7	TOP8	TOP9	TOP10	TOP11
1	-	4.5924	3.9363	3.9363	4.2643	5.9045
0.439	100	1.7202	1.1398	1.2267	1.0440	1.6013
0.291	150	1.0641	0.7649	0.7997	0.7162	1.0370
0.193	200	0.7773	0.6196	0.6342	0.5981	0.7780
0.128	250	0.6527	0.5633	0.5700	0.5541	0.6575
0.085	300	0.6016	0.5474	0.5506	0.5434	0.6053
0.056	350	0.5871	0.5526	0.5541	0.5508	0.5894
0.037	400	0.5929	0.5701	0.5709	0.5693	0.5941
0.025	450	0.6109	0.5955	0.5958	0.5951	0.6115
0.016	500	0.6366	0.6261	0.6262	0.6259	0.6369

3: Energy in mJ for flooding different network topologies. The first line indicated the power use without AQF algorithm.



(a) Total energy use for a faked packet. Note that topology 8 and 9 have for simple flooding the same energy need since they equal in the number of nodes.



(b) Average energy use per node, which is equal for the simple non-authenticated version of the flooding algorithm.

2: Energy rewards for selected topologies for 100% fake queries.

Figure 4 illustrates different topologies with varying parameter p_f denoted on the x-axis and the involved energy requirements on a logarithmic scaling in mJ. Horizontal lines represent the power need of nodes without the securing mean, i.e. for simple flooding which is independent of the forwarding

probability p_f. The curves for the network topologies with varying energy are leftward curved with an upward slope and a global energy minimum. These minima depend on the topology and vary with the probability of forwarding a fake packet between 5% and 15%. Results are displayed in Table 3 with the pertinent authenticator size m and mean number of keys that a node knows.

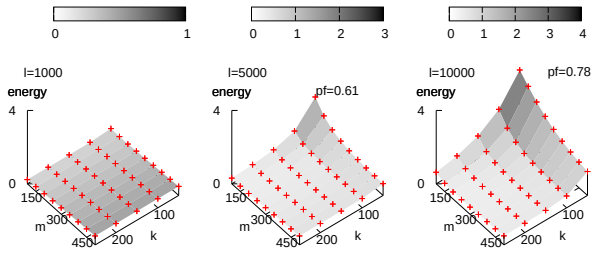
With increasing p_f the costs for security drop while more and more sensor nodes are affected by a faked query, and consequently the amount of energy used increases. Although the choice of topology 7 and 8 are similar and only distinct in 2 nodes, this has an enormous impact on the power needs. Especially the missing node C in topology 8 reduces the connectivity, thus creating a "bottleneck" and flattening the flooding depth. The energy curve of topology 9 almost equals topology 8 since right at the beginning the query is only spread twice. When looking at the energy per node of topology 10 and 11 it is obvious that they have the least energy need, due to the low interconnectivity between the individual nodes. According to this, their slope is comparatively flat, even up to a propagation probability of 40%. It is obvious that although the additional computation effort is made by using the AQF, the energy is far below the line of the unsecured network when assuming that 100% of the packets have a fake authenticator.

Finding the right mix of parameters that determine probability p_f and the way they relate to the energy is the key for successfully applying the AQF algorithm to a practical example. Due to this Figure 3 illustrates their correlation. Note that all three planes do intersect. Using a key pool with $\ell = 1000$ keys, the plane is relatively flat with an energy maximum at $k = 250, m = 500$ of $0.4970mJ$. The value for p_f is at this point 0%, meaning that fake query are dropped with 100% probability at the first sensor node. With increasing parameter ℓ the plane shifts to a new energy maximum at $m = 100, k = 50$ of 2.0382 mJ due to the fact that the parameter p_f here around 61%. Choosing a key pool of size 10 000 the situation becomes even more extreme, attaining power needs of 3.5078 mJ according to the high propagation of faked queries (p_f=78%).

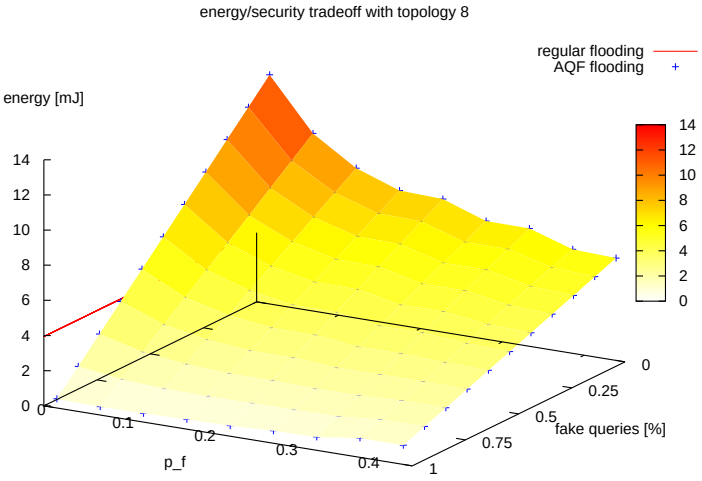
The explanation for this lies in the probability p_f for which we do only indirectly account thru the parameters of m, k, and l. In fact the evident increase in energy as shown in Figure 3 is due to the rapid increase of p_f which grows at many points beyond the admissible range of 50%, e.g. the energy maxima for $\ell = 5000$ and $\ell = 10000$ are $p_f = 0.61$ and $p_f = 0.78$. For this reason it is important to choose parameter ℓ according to the topology since a keypool of 10 000 keys does not achieve the desired effect when having only 15 sensor nodes in a network scenario.

4.1 Energy/Security tradeoff using topology 8

Since the energy wasted is dependent on the severity of the intrusion and the number of faked packets, the figures shown so far do not really account for this. Due to this reason, we now proceed with the problem as motivated earlier in the introduction, namely to find an adequate level of energy that secures the network from outside intrusion. That is in particular the solution of the energy/security tradeoff in relation to the number of packets sent by an adversary. Therefore a



3: The way the input parameters authenticator size m, number of keys per node k, and number of keys in the key pool ℓ correlate with the energy use.



4: Area representing the relation between the number of faked queries in percent, the probability of forwarding a fake packet p_f, and the corresponding energy required.

new variable is included in the model which represents the number of fake queries among the good ones, starting from 0 up to 100 percent. Since such a figure was missing is the section before we are now able to state quantitative predictions about how the energy and security level relate to the severeness of the intrusion.

For fixed parameters ℓ=4000, k=50, and m varying between 100 and 500 bits we compute the probability p_f and the hereby involved power. Denote at this point that there may be more than one parameter configuration that lead exactly to the same forwarding probability, i.e. when doubling ℓ and k the same probabilities p_f are obtained which follows due to obvious reasons. This analysis considers the total amount of energy which is used for a query to be flooded in topology 8, and can also be applied to the other scenarios. The corresponding figure 4 shows the result. The axes labels are: the *percentage of faked queries* on the axis of abscissae, the *security level* as explained by the formula above on the y axis, and the corresponding *power need* in mJ on the z-axis.

Note that 2 different data sources are contained in the illustration. The line on the left side at $p_f = 0$ shows the energy that would be used without the use of any securing

mean. That is packets are received by a node and transmitted again without the computation of hash values (cf. 3.1 variable MNKK), and validation of keys in between. In this case a constant amount of 3.94 mJ is needed for the network to be reached that is completely independent from the probability of forwarding a fake packet p_f.

The second point of interest is the plane showing the relation of faked queries, and p_f to the amount of energy hereby involved. This graph reads as follows: If we assume only sound packets to be send and only little security is in use – that is p_f is high – the AQF algorithm outperforms the regular flooding procedure with respect to energy. By increasing the portion of faked packets, the authenticationed flooding shows effect and the energy need starts to drop. On the other end of the scale ($p_f = 0.016$ and no fake queries) the situation is similar since the securing mean does not show any effect and reached a energy maximum of 13.0392 mJ. As the number of faked queries reaching the network is increasing, more and more packets get filtered out of the network this causing a rapid drop of the energy down to 0.5236 mJ if we assume 100% fake packets to be sent.

5. CONCLUSION

Due to the low-power nature of wireless sensor networks it is hard, choosing the right path between the different constraints such as security, energy, authenticity et cetera. The more it is key that appropriate measures are applied prior to the deployment of a network which shelter against intrusion from outside.
The analysis presented here reveals how the probabilistic model checking tool PRISM can successfully be applied to those challenging problems like the AQF algorithm. We proved, although commonly known problems like state space explosion prevail, there's an enormous wide range of problem instances allowing a deep investigation. The more, what differs them from common simulation tools is the high precision of the obtained results that go without the need of confidence levels.

By the use of reward functions quantitative assertions for a variety of properties can be verified. It turns out that due to the number of nodes in the range of hundreds as proposed in [1], we were not able to validate these results with the formal method approach. The more the here presented analysis should be understood as a complementary approach, that can be used to render simulation input parameters more precisely. As such we proved against our previous anticipation, that the choice of the topology has an tremendous impact on the network security for probabilistic algorithms.

Though our analysis is restricted in the number of nodes for which we give evidence, networks around the size of 15 nodes suffice most of the real-life applications. We further believe that most of the results presented here scale also well for even bigger networks due to symmetry reasons, which needs further proof in future work. So we intent to rerun the presented experiments within an simulation environment to make this work more compareable and strengthen the scalability assumption. Further work could also be spent on the model of the receiving process to include collisions, the initialisation phase, node breakdown failures etc., thus modelling a more realistic network.

6. REFERENCES

[1] Zinaida Benenson, Felix C. Freiling, Ernest Hammerschmidt, Stefan Lucks, and Lexi Pimenidis. Authenticated query flooding in sensor networks. *PerCom Workshops 2006*, 2006.

[2] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In H. Hermanns and J. Palsberg, editors, *Proc. 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

[3] B. Kaliski. The md2 message-digest algorithm [rfc1319]. Technical report, RSA Data Security, Inc., April 1992. [RFC1319] Network Working Group Request for Comments.

[4] M. Kwiatkowska, G. Norman, and A. Pacheco. Model checking expected time and expected reward formulae with random time bounds. *Computers & Mathematics with Applications*, 51(2):305–316, 2006.

[5] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In P. Kemper, editor, *Proc. Tools Session of Aachen 2001 International Multiconference on Measurement, Modelling and Evaluation of Computer-Communication Systems*, pages 7–12, September 2001. Available as Technical Report 760/2001, University of Dortmund.

[6] The Network Simulator – ns-2. `http://www.isi.edu/nsnam/ns/`.

[7] Datasheet: The TMote Sky Sensor from Moteiv. `http://www.tmotesky.com/products/docs/tmote-sky-datasheet.pdf`.

[8] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing pair-wise keys for secure communication in ad hoc networks: A probabilistic approach, 2003.