Analysis and Extension of Benenson's Robust User Authentication Scheme

Wolfgang Meyer zu Bergsten, Rudolf Mathar

Institute for Theoretical Information Technology
RWTH Aachen University
52062 Aachen, Germany
Email: {wmzb,mathar}@ti.rwth-aachen.de

***Abstract*—In [1], Benenson, Gedicke and Raivio propose a user authentication scheme for sensor networks. The scheme considers how to organize access control to a wireless sensor network for authorized users and rejection of unauthorized adversaries.**

In the first part of this paper we analyze and optimize the proposed authentication scheme.

The second part deals with distribution of public keys to sensor nodes. Benenson's protocol requires periodic distribution of a certification authority's public key to sensor nodes. Considering wireless sensor network applications, these updates have to be accomplished over the air on a deployed network. We propose a wireless, online reconfiguration scheme implementing the key distribution.

Keywords: wireless sensor networks, robust authentication, key distribution, security

I. Introduction

Wireless sensor networks (WSN) are deployed in various scenarios where measurement of distributed data is necessary. These cases include agricultural usage (temperature, humidity), fire detection or military applications. The data is given to users for analysis.

Deployment of a sensor network is associated with investments. A company that deploys a WSN consisting of sensor nodes wants to have a return on investments by selling access to sensor data. Therefore, restricted access to the sensors such that only legitimate users can read data is essential.

A. Requirements and Goals

Sensor nodes in a WSN should be cheap. Therefore, sensors must use minimal resources, especially a small battery, low complexity integrated circuits and communication modules. These necessities result in the following requirements for protocols and algorithms in sensor networks:

- short transmitted messages,
- fast algorithms,
- low power algorithms.

Respecting these requirements is the basis for a sensor network fulfilling a given purpose.

Additionally, security properties have to be achieved. The most important security aspects to be considered are

- to inhibit false authentication,
- to protect against tampering with data.

B. Contributions

The original article [1] leaves a couple of questions unanswered. In this paper we consider the following two:

- Sec. III: *Detailed analysis and improvements of the protocol steps and parameters*. In some cases, the selection criteria of parameters or values for cryptographic primitives are not obvious. If the components are required, we will explain the necessity of the parameters and steps. In other cases we enhance the protocol steps. In particular, we reduce the charge and time effort payed by the sensors for an authentication of a user, while keeping an identical security level. The reduction of energy usage by the sensors is especially noteworthy as their energy is a scarce resource, whereas the user's energy typically can be recharged easily.
  A second enhancement from the protocol modification is the possibility to improve resistance against power-drain denial-of-service attacks (Sec. III-A).
- Sec. IV: *New protocol for secure distribution of the Certification Authority's (CA) public keys.* The CA's public key has to be updated regularly in order to make up for the unavailability of a trustworthy clock source in the sensor nodes. The distribution of new keys is not considered in the original paper. We propose a feasible protocol for distributing these keys which is based on the existing authorization protocol. By reusing the cryptosystems already used in the original proposal, we achieve minimal additional resource consumption.

II. Description of the Robust User Authentication Scheme of Benenson et al.

This section describes the user authentication scheme conceived by Benenson et al. as given in the original work [1]. Firstly, the system and adversary model as well as the problem statement are presented. The proposed protocol for solving the problem is specified afterwards. Finally, the topics considered in this paper are explained.

A. System and Threat Model

Benenson et al. consider a large static sensor network. Several mobile users are authorized to access the sensor network using mobile devices. During a single query from a

user to the WSN, the network topology is fixed, i.e. the users and sensors have to stay in place. A user can communicate directly with m sensors that are in range of its query device. The goal of an adversary is to get access to the sensor network without valid credentials.

B. Problem Description

The problem addressed in [1] is the authentication of a user to sensors. The critical properties of the authentication is to fulfill the *safety* and *liveness* requirements.

- If the authentication ensures that only legitimate users are granted access to sensor data the scheme is *safe*.
- If a legitimate user eventually receives the data it queries the scheme is *live*.

C. Proposed Solution

The cryptographic primitives used in the original protocol are an Elliptic Curve Cryptosystem (ECC) [2] and a hash function. Both systems are efficiently implementable on sensor nodes [3]. Additionally a signature scheme providing message recovery upon verification is employed. Specifically, the scheme uses a 163-bit ECC system for generating Nyberg-Rueppel signatures [4]. The SHA1 cryptographic hash function [5] is used to calculate hash values.

During system setup and before deployment of the sensor network, a certification authority CA is created by the deploying entity. The CA is assumed to have computation resources comparable to a personal computer. The CA owns a private/public key pair $(K_{CA,priv}, K_{CA,pub})$. Each legitimate user receives a private key $K_{U,priv}$ and a certificate containing the user's public key $K_{U,pub}$, signed by the CA:

$$cert_U = sign_{K_{CA,priv}}(K_{U,pub})$$

. Every sensor contains a copy of the certificate authority's public key, which allows the sensors to verify the user's certificate.

The protocol steps are as follows:

1)
$$U \rightarrow WSN : U, cert_U \tag{1}$$
The user U sends its identity and its certificate $cert_U$ to the wireless sensor network.

2)
$$s_i \rightarrow U : (s_i, n_i) \tag{2}$$
Each sensor node $s_i, i \in \{1, \ldots, m\}$ sends its nonce n_i (a random number used once, used to prevent replay attacks [6]) to the user U.

3)
$$U \rightarrow s_i : S_i = sign_{K_{U,priv}}(h(U, s_i, n_i)) \tag{3}$$
The user U consecutively responds to each sensor node s_i with a signature S_i on the hash value of U, s_i and n_i concatenated.

TABLE I
TIME AND CHARGE CONSUMPTION FOR USERS (U) AND SENSORS (S) FOR A SINGLE AUTHENTICATION [1].

	No. of	Time[s]		Charge[mC]	
Step	Packets	U	s	U	s
1	9	0.011	0.011	0.24	0.24
2	1	0.001	0.001	0.03	0.03
3	3	65	0.004	117	3.6
4	1	0.001	375	20	675

4)
$$s_i : \quad K_{U,pub} = \quad verify_{K_{CA,pub}}(cert_U) \tag{4}$$
$$s_i : \quad verify_{K_{U,pub}}(S_i) \tag{5}$$
The sensor node verifies and decodes the user's certificate $cert_U$ with the public key of the CA, resulting in the user's public key $K_{U,pub}$. If the verification is successful, the node verifies the correctness of the signature S_i with $K_{U,pub}$.

In contrast to regular certificates [6], the users' certificates do not contain expiration dates. This is due to the fact that time synchronization in WSNs is not feasible (no realtime clocks in the nodes due to power constraints; wireless time synchronization expensive in terms of transmit/receive power, security problems). To counter this restriction, key renewal is facilitated using periodic updates of the CA's public key in the nodes.

D. Resource Consumption

The consumption of charge and time during authentication of a single sensor node to a user is shown in Tab. I. We see that transmitting/receiving a single message takes $T_t = 0.001$s and $C_t = 0.03$mC. Creating a signature takes $T_s = 65$s and $C_s = 117$mC. In step 4 two signatures are processed, so verifying and decoding a single signature takes half the time and energy, i.e. $T_v = 187$s and $C_v = 336$mC.

E. Open Questions

The original paper leaves two questions unanswerd:

- What is the cryptographic role of the protocol steps and parameters?
- How are updated public keys of the CA distributed to the sensors?

We will answer them in the next sections.

III. Protocol Analysis and Enhancements

In this section, we analyze the protocol described in the original paper and propose enhancements regarding electric charge consumption and time. The steps correspond to the steps originally proposed. First, we describe the function of the protocol step and its parameters regarding its security role and necessity. If possible, an enhancing modification of the protocol step is proposed. The modifications reduce power consumption and execution time, while keeping an identical security level.

1) In the first step (Eqn. 1), the user's identity and certificate is transmitted to all sensors in the network. If more than one concurrent user is allowed in the WSN, the certificate and corresponding identity are required by the sensors to allow for encryption to the user in step 4. Thus both are required and the original step is used unmodified.
2) The sensor transmits a challenge n_i to the user in the second step (Eqn. 2) as part of a challenge-response protocol [6]. This protects the nodes against replay attacks. The identity s_i is transmitted as well to identify the source of the n_i. We see no possibilities to enhance this step while keeping an identical security level. The original step is used unmodified.
3) In the third step (Eqn. 3), the user authenticates to the sensor node using the challenge n_i, thereby achieving the second step in the challenge-response protocol. We can minimize this step to the necessary components by omitting the user's identity from the signed hash (it is not used in any step). Furthermore, removing the sensor's identity from the hashed values and instead prepending it to the message allows the sensor to just process messages in which it is addressed. Therefore, this step can be changed to the following step:

$$U \rightarrow s_i : S_i = (s_i, sign_{K_{U,priv}}(h(n_i))) \quad (6)$$

The attentive reader will note that this modification increases the length of the transmitted message by the length of the nonce (when keeping the identical security levels). However, for networks with more than one sensor, the combined energy consumption of all nodes is lower with the new approach than with the original one. This property is further explained in the next step.
4) In the forth step (Eqn. 4, 5), the signatures are verified and decoded. We see no possibility to reduce the operations in any way while keeping the same security level. However, with the addressing introduced by Eqn. 6, the sensor just has to decode packages targeted at it. For both protocols, the sensor has to verify and decode $K_{U,pub} = verify_{K_{CA,pub}}(cert_U)$.
For each sensor's challenge, the user sends one message $S_j, j \in \{1, \ldots, m\}$. In the original paper, every node s_i receives and verifies all messages S_j. Then, the node checks if it contains its identity and the corresponding node. If the node is not targeted by a message, the verfication is refused and the next message is processed. Once the user is authenticated, the node can ignore further messages S_j. The average number of messages that have to be processed by a sensor node is $\frac{m}{2}$ (assuming the sensor quits listening and verifying data after successful authentication until the user authenticated to all remaining sensors). Therefore, the average power and time spent for authorization on the sensor side is

$$C = \frac{m}{2} \cdot (C_t + C_v) \text{ and}$$

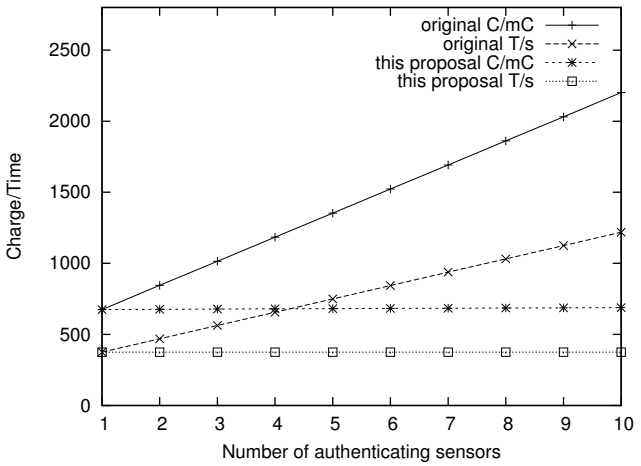

Fig. 1. Charge consumption and time spent by a single node during authentication of a user to m sensors.

$$T = \frac{m}{2} \cdot (T_t + T_v).$$

Charge and time spent on the user's side is

$$C = \frac{m}{2} \cdot (C_t + C_s) \text{ and}$$
$$T = \frac{m}{2} \cdot (T_t + T_s).$$

With the new proposal, the sensor has to receive $\frac{m}{2}$ times the s_i part of the messages S_i and one signature $sign_{K_{U,priv}}(h(n_i))$ (taking the same assumptions as before that signatures are only verified until authentication is established). Then, just one signature verfication $verify_{K_{U,pub}}(sign_{K_{U,priv}}(h(n_i)))$ is exectued. Therefore, the average power and time spent for authorization by each sensor is

$$C = \frac{m}{2} \cdot C_t + C_v \text{ and}$$
$$T = \frac{m}{2} \cdot T_t + T_v.$$

As depicted in Fig. 1, this results in a large improvement compared to the original protocol for a network with more than one sensor. In the unlikely case of a single sensor, the degradation is negligible.
Charge and time consumed by the user with the new protocol is

$$C = \frac{m}{2} \cdot (C_t + C_s) \text{ and}$$
$$T = \frac{m}{2} \cdot (T_t + T_s).$$

For sensor networks below a certain size, the cost of one message is negligible in the complete electric charge and time requirements. In sensor networks with about 50 sensors the charge consumption increases by about 10 percent of the single sensor amount. However, a sensor network using the original proposal would have an increase of more than 1,000 percent.
Therefore, the addressing introduced by Eqn. 6 enhances the lifetime of a sensor with a battery significantly.

A. Power-Drain Attack Resistance

The new scheme improves resistance against power-drain denial-of-service attacks. During such an attack, an adversary sends bogus packets to sensors with the goal of depleting their power source. When a node receives such a packet, it is processed. An attacker would gain the biggest benefit if it executes steps 1 to 3 with faked packages. In step 4, the sensors try to verify the packet sent in step 3 and spend a lot of time and energy on that task.

The new protocol reduces the power spent for decoding such bogus communication. If the sensor is not the addressed one, it just ignores the packet and does not execute the expensive verification, thereby saving power. Additional methods, like putting the sensor node to sleep for a certain time if a packet is not verified successfully, can be employed with both protocols alike. As a result, the WSN will withstand this specific attack for a longer time if the modified protocol is employed.

IV. Key Distribution Scheme

In this section we solve the Certification Authority key distribution problem raised in the original paper [1]. Key updates have to be feasible wirelessly. The basic idea of our method is to let the nodes authenticate a user and then receive the updated public key of the CA, enveloped in a Nyberg-Rueppel signature. Key renewal is necessary because a sensor network's deploying entity might sell access to the sensor network to customers for a specific time interval. If the key always stays the same, a non-subscribed user who once had a subscription is possibly still in possession of its signed key. This allows for continued querying of data, which is an illegal access in the considered use case.

A. Key Update Protocol

There are two possibilities for key distribution to the sensor nodes. Firstly, we can physically access the sensor nodes and reprogram them using a wired connection. The second possibility is to update nodes remotely over the air.

1) Physical Access Update: This approach ensures that the communication is hard to eavesdrop and hard to manipulate, thus implementing the most important security requirements for the update. Furthermore, this method allows to verify the successful update immediately and, as we are physically at the node, we can also inspect it for malfunction, damage and signs of attacks. The big problem with this approach: It is contradicting most use cases for wireless sensor networks. WSN installations are targeting cheap deployment, cheap maintenance and large numbers of nodes. Having to physically access each node for updating the certificate is expensive, depending on the frequency of such updates and the number of nodes. In special cases, like monitoring conditions in hazardous areas (monitoring temperature inside a vulcano) or inaccessible areas (dangerous terrain, combat zones), access might even be impossible. In order to provide reasonable security, updates have to be performed lots of times during the lifetime of a sensor node, each time creating the cost.

As a consequence, key update schemes requiring physical access to the nodes are not feasible.

2) Wireless Update: The second possibility is to execute the update wirelessly. The problem in such a scheme is the possibility of unreliable communications. This may result in sensor nodes that do not perform the key update. Reasons for this can be natural communication problems, i.e. weather or seasonal conditions or an adversary disrupting communications. If a key update is inhibited for some nodes, the keys contained in the WSN are not consistent, communication from users to parts of the network is not functional and the performance of the network degrades.

If wireless updates are employed, an adversary might attack the WSN by replaying outdated, formerly valid messages, or by tampering with transmitted data. Additionally, the protocol should be resistant against power-drain attacks.

During system setup, a Lamport one-time password [7] is stored in the sensors. The rest of the setup actions stay the same. Lamport's one-time-password authentication protocol, also known as S/KEY [8], is a fast way to create one-time passwords. The security of Lamport's authentication protocol is based on the computational infeasability of reversing a cryptographic hash function. A Lamport one-time password chain to create k passwords is created by applying a hash function k times to its own output (Eqn. 7). The key generation is seeded with a random 160-bit value p_0. For this protocol, we choose the SHA1 hash function as it is already required for the authentication protocol.

$$p_l = \text{SHA1}^l(p_0), l = 1, \ldots, k \tag{7}$$

The resulting last password p_k is stored in all sensor nodes, and all password $p_{(0..l)}$ are stored by the CA. They are used for authentication in reverse order.

Generating Lamport authentication pads is cheap. We created a billion (10^9) passwords using a computer with a Pentium M CPU in less than 15 minutes. A billion passwords would last for more than 30 years if the key is updated every minute or nearly 2 000 years if updated once an hour.

The update procedure consists of the following steps:

1) A valid user authenticates to sensor nodes using the protocol from Sec. III.
2) $$U \rightarrow WSN : (p_{l-1}, sign_{K_{CA,priv}}(N_{CA,pub}))$$
   The update user sends a message containing the next password and the new CA key $N_{CA,pub}$, encrypted in a Nyberg-Rueppel signature using the CA's current private key, to the WSN.
3) $$s_i : \quad check(SHA1(p_{l-1}) = p_l)$$
   $$s_i : \quad K_{CA,pub} = verify(sign_{K_{CA,priv}}(N_{CA,pub}))$$
   The sensor first checks if the provided password is the next one-time password. If the check is successful, the signed part of the message is verified and the resulting

key is stored as the new public key of the CA. The password p_{l-1} replaces the previous password p_l in the sensor.

To allow for recovery from temporary communication failures, such a packet is sent again after a specific amount of time. E.g. it would be reasonable to resend the update two times, each after one tenth of the public key update time interval. In this case, if an adversary is just temporarily disrupting the communication to the nodes, the affected nodes will be synchronized to the network again after a certain time.

V. Analysis of the Key Distribution Scheme

The signing of the new public key of the CA with its old private key ensures resistance against acceptance of invalid update messages by the sensors. The one-time password is not necessary to secure the password update, but it is significant in resisting a denial of service attack.

A. Impersonation Attack

An attacker is not able to fake a new public key, because the sensors verify the signature of the message and accept the new key only if the signature is valid. The new key can only be successfully signed if the creator of the message is in possession of the CA's private key. It is assumed that the CA's private key is well protected and inaccessible to an adversary.

B. Replay Attacks

A replayed message will not be accepted by the sensor node because the private key used to sign the replayed message does not match the public key currently used to verify the signature in the sensor. Additionally, the verification of the one-time password would fail.

C. Power-Drain Denial of Service

A power-drain denial of service attack is still possible, however the addition of the one-time password reduces the resources spent if an invalid message is sent. This is due to the fact that the verification of the one-time password requires just the hashing of the previously stored password, and if the password is not valid, the expensive Nyberg-Rueppel signature verification is not performed.

D. Complex Attack

Despite the protection against the previous attacks, a more sophisticated scheme can be executed by an adversary to mount a power-drain attack. This can be achieved if the adversary performs the following steps.

1) Disrupt communication from the key update user to sensors for the time of the key update,
2) Eavesdrop the key renewal message and extract the Lamport password,
3) Send messages starting with the one-time Lamport password, but fill the signed key part with random data.

During such an attack, affected sensor nodes will execute expensive verification of the faked Nyberg-Rueppel signature, thereby depleting the energy source. The Lamport password does not provide protection against this attack.

E. Resource Efficiency

The proposed key update protocol reuses code which is present in the sensor nodes anyway: The ECC Nyberg-Rueppel signature and the SHA1 hash function. Additional code and memory required for this update is minimal, increasing the ROM an RAM requirements insignificantly.

F. Adding Nodes

Over time, it might be necessary to add nodes to the sensor network, either to extend the network or to replace failing nodes. When a new node is added to the network, we have to initialize it with the current CA's public key and the current Lamport password.

VI. Conclusions

In the first part of this paper, an analysis of Benenson's robust user authentication scheme [1] is given. We conclude that the scheme fulfills its promise of user authentication. During the analysis, several optimization possibilities have been discovered. A new protocol based on the original one was proposed and benefits analyzed. It was shown that the optimized protocol provides enhancements in terms of charge consumption and execution time, while maintaining the same security level.

The second part addressed the key distribution required for the authentication scheme. We proposed a lightweight protocol reusing primitives already existing in the sensors due to the usage of Benenson's authentication scheme. This protocol protects the wireless sensor network from impersonation and replay attacks and also provides some resistance against denial of service attacks.

References

[1] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," in *Workshop on Real-World Wireless Sensor Networks (REALWSN), 2005*.

[2] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. New York: Springer, 2004.

[3] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," 2004.

[4] K. Nyberg and R. A. Rueppel, "A new signature scheme based on the dsa giving message recovery," in *Proceedings of the 1st ACM CCCS*. Fairfax: ACM, 1993.

[5] *Secure hash standard*. Washington: National Institute of Standards and Technology, 2002, uRL: `http://csrc.nist.gov/publications/fips/`. Note: Federal Information Processing Standard 180-2.

[6] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 2001. [Online]. Available: http://www.cacr.math.uwaterloo.ca/hac/

[7] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[8] N. Haller, "The S/KEY one-time password system," in *In Proceedings of the Internet Society Symposium on Network and Distributed Systems*, 1994, pp. 151–157.