

Protecting Small Keys in Authentication Protocols for Wireless Sensor Networks

Kalvinder Singh

Australia Development Laboratory, IBM
and School of Information and Communication Technology, Griffith University
Gold Coast, Queensland, Australia, kalsingh@au.ibm.com

Kartikey Bhatt, Vallipuram Muthukumarasamy

School of Information and Communication Technology, Griffith University
Gold Coast, Queensland, Australia
{kartikey.bhatt@student., v.muthu@}griffith.edu.au

Abstract

Wireless sensor networks provide solutions to a range of monitoring problems. However, they also introduce a new set of problems mainly due to small memories, weak processors, limited energy and small packet size. Thus only a very few conventional protocols can readily be used in sensor networks. This paper introduces efficient protocols to distribute keys in wireless sensor networks. We also show how to harden the protocol against brute force attacks on small security keys. This is achieved without the necessity of using traditional encryption. The proposed protocols guarantee that the new key is fresh and that the communicating nodes use the same key. The protocols were implemented in TinyOS and simulated using TOSSIM. Energy consumption and memory requirements are analysed in detail.

1 Introduction

Wireless sensors and actuators have the potential to drastically change the way people live as they permeate the environment. Sensors can be used to monitor objects, space and the interaction of objects within a space. Sensors can monitor a wide range of diverse phenomena by collecting information such as vibrations, temperature, sound, and light. Different sensors have different costs. For example, a sensor detecting light will have different costs to a sensor recording sound. However, the less costly sensors can be used to detect a phenomena before alerting the more costly sensors to start their monitoring. As the number of heterogeneous sensors increases, so will the amount of interactions between the sensors.

Key establishment protocols are used to set up shared secrets between sensor nodes. Sensor nodes suffer from lim-

ited computational capabilities, battery energy, and available memory. Asymmetric cryptography is unsuitable for most sensor architectures due to higher computational overhead, and energy and memory consumption. Random key predistribution schemes [4],[7],[14] are a major class of key establishment protocols for sensor networks. One of the issues with random key predistribution schemes is that if a certain number of sensor nodes become compromised, then the entire sensor network can become compromised. We propose a protocol that establishes a new pair-wise key between two sensor nodes. If the underlying random key predistribution scheme becomes compromised in this new protocol, it will not affect the entire sensor network.

The PIKE [3] and Singh [18] schemes showed that there are a class of problems where one or more sensor nodes act as a trusted intermediary to facilitate key establishment. When using symmetric key cryptography, if two entities sharing no previous secret want to communicate securely with each other, they generally do so with the assistance of a third party. Typically, the trusted intermediary provides an authentication service that distributes a secure session key to the sensor nodes. The issues with using a sensor node as the trusted third party are: the trusted intermediary can become compromised; key sizes in sensors nodes are not large; sensor networks may only require authenticated messages, without any need for encryption.

Another class of problems occur when human interaction with sensor networks is required [5]. People authenticating with a sensor network (using a PDA or mobile phone) will prefer using short passwords over long binary keys.

In this paper we propose a number of protocols to address these problems. The proposed protocols combine traits from a two party password-authenticated key exchange protocol [1] and a symmetric key server-based key establishment protocol [11]. The proposed protocols do not

require traditional encryption to transport the new session key. We will show the sensor nodes can prove that the new key is fresh. We will also demonstrate how key confirmation ensures that the nodes are guaranteed to be using the same key. The proposed protocols can be extended to use human readable passwords instead of binary keys.

2 Sensor Networks

We refer to a sensor network as a heterogeneous system combining small, smart, cheap, and sensing devices with general-purpose computing elements. Sensor network applications [2] include tracking bushfires, monitoring wildlife, conducting military surveillance, and monitoring public exposure to contaminants.

The sensor nodes are resource constrained. A typical sensor node is the Mica mote [22]. The Mica motes contain a 4 MHz processor with 512 KB flash memory and 4 KB of data memory. A Mica mote also has a separate 512 KB flash memory unit accessed through a low-speed serial peripheral interface. The RF communication data transfer rate is approximately 40 kbps. The maximum transmission range is approximately 100 metres in open space. One of the major issues in sensor networks is to reduce the communication costs. Communication is the most energy intensive operation, and many protocols are designed to reduce as much communication overhead as possible.

Mica sensor nodes run TinyOS [20], an event-driven operating system specifically designed for wireless sensor environments. The memory footprint for TinyOS is small: a minimum installation (the core components) uses 400 bytes of data and instruction memory. TinyOS is developed in nesc and supports other hardware platforms. The TinyOS network packet wraps the payload during sensor node communication. To save space, it does not transmit the source address of the sender.

Network security in sensor environments differ in many ways from other distributed systems. Sensor nodes have little computational power, thus even efficient cryptographic ciphers must be used with care. Security protocols should use a minimal amount of RAM. Communication is extremely expensive, so any increase in message size caused by security mechanisms comes at significant cost. Energy is the most important resource, so each additional instruction or bit transmitted means the sensor node is a little bit closer to death. Nearly every aspect of sensor networks is designed with extreme power conservation.

Perrig et al. proposed a cryptography library using symmetric keys [17]. However, recent work has shown that asymmetric keys can also be used [16]. Until recently it was believed that asymmetric key algorithms were too heavy-weight. Thus much of the work on authentication and key establishment protocols have used a symmetric key cryp-

tography library, A survey of current authentication mechanisms in wireless sensor networks is given in [21].

One of the major issues with some of the key establishment protocols, is that there are situations where if a sufficient number of sensor nodes become compromised, the entire sensor network may become compromised. The other major issue is that the existing key establishment protocols do not allow for human readable passwords to be used to authenticate devices.

3 Notation and assumptions

This paper will use the following notation to describe security protocols and cryptographic operations:

A, B	The two nodes who wish to share a new session key.
S	A trusted server.
N_A, N_B	Nonces generated by nodes A and B respectively.
$[[M]]_K$	Encryption of message M with key K to provide confidentiality.
$[M]_K$	One-way transformation of message M with key K to provide integrity.
K_{AB}, K'_{AB}	The long-term key initially shared by A and B and the new session key.
K_{AS}, K_{BS}	Long-term keys initially shared by A and S , and B and S respectively.
X, Y	The concatenation of data strings X and Y .
$A \rightarrow B : m$	A sends a message m to B .
\xrightarrow{m}	Another way to define sending of message m .

4 Limitations and Problems

This section discusses limitations and problems with existing network protocols when applied to wireless sensor networks.

4.1 Two party password-based protocols

We first investigate a method to create a new key between two sensor nodes, such that even if the old key is compromised, the new key will not be compromised. Password-based protocols are a good source of protocols that provide this feature. Another useful feature is that they do not rely on the fact that original key is a strong key.

The PPK protocol [1] is an example of a password-based protocol. It only needs two messages to complete the key exchange. However, the protocol was defined using the RSA algorithm. But the RSA has been shown not to be feasible in sensor networks [16].

4.2 A three party protocol

We now examine a three party key establishment protocol. The Janson–Tsudik 3PKDP protocol [11] is a server-based protocol, and relies upon long term keys held between the client and the server. The following constructs are used when defining the protocol:

$$AUTH_A = [N_A, K_{AB}, A]_{K_{AS}}$$

$$MASK_A = [[AUTH_A]]_{K_{AS}}$$

$$AUTH_B = [N_B, K_{AB}, B]_{K_{BS}}$$

$$MASK_B = [[AUTH_B]]_{K_{BS}}$$

Janson–Tsudik protocol (shown in *Protocol 1*) uses a novel approach when encrypting the key. This allows the *MASK* values to be either generated via an encryption algorithm or a MAC algorithm.

Protocol 1 Janson–Tsudik 3PKDP protocol

M1	$A \rightarrow B :$	A, N_A, N'_A
M2	$B \rightarrow S :$	A, B, N_A, N_B
M3	$S \rightarrow B :$	$AUTH_A, MASK_A \oplus K_{AB},$ $AUTH_B, MASK_B \oplus K_{AB}$
M4	$B \rightarrow A :$	$AUTH_A, MASK_A \oplus K_{AB},$ $[N'_A, N'_B, B]_{K_{AB}}, N'_B$
M5	$A \rightarrow B :$	$[N'_A, N'_B, A]_{K_{AB}}$

In wireless sensor networks, constraints may lead to the requirement to have some sensor nodes as a trusted third party. A problem with sensor nodes is that they can be easily compromised, and any keys that the nodes contain can usually be obtained if someone has physical access to the node.

A problem with *Protocol 1* is that if an adversary recorded the protocol between A , B and S , then the adversary could obtain S and have the keys K_{AS} and K_{BS} . The adversary can then calculate the *AUTH*, *MASK* values and finally calculate the value for K_{AB} .

Also, depending on the size of the keys K_{AS} and K_{BS} , a brute force attack is also possible. If we assume that A is malicious or compromised, then A can perform the following calculation to compromise the key between B and S . We first specify $K_i \in X$, where X is the set of all possible keys between B and S . The adversary (in this case A) can perform the following calculation for each K_i as shown in Equation (1).

$$AUTH_B = [N_B, K_{AB}, B]_{K_i} \quad (1)$$

If there is more than one K_i then for each of the values the following calculation, as shown in Equation(2), can be performed.

$$MASK_B = [AUTH_B]_{K_i} \quad (2)$$

In *Protocol 1*, an adversary can also perform an offline attack, without any of the sensor nodes being compromised.

The adversary can perform the following calculations for each K_i as shown in Equation (3) and Equation (4). If the Equation (4) returns true, then the adversary has discovered both K_{AB} and K_{BS} . They are the values K and K_i respectively from Equation (3).

$$K = [AUTH_B]_{K_i} \oplus (MASK_B \oplus K_{AB}) \quad (3)$$

$$[N'_A, N'_B, B]_{K_{AB}} = [N'_A, N'_B, B]_K \quad (4)$$

It should be noted that this attack is only possible if the keys between the sensor nodes and the intermediary node are small enough to perform a brute force attack.

4.3 Three Party Password–Based Protocols

There are several three party protocols, which assume that the long term keys held by the parties and the trusted server are small. Gong, Lomas, Needham and Saltzer (GLNS) [9] published the first three party password–based protocol. The protocol was later revised [8]. However, the protocol relies upon public key cryptography, which may not be applicable to a sensor node.

Steiner *et al.* [19] created a novel three party protocol, where the server cannot obtain the session key. However, flaws in the protocol were later found [6], [12].

The Lin *et al.* [13] protocol has no known attack, however, it does need seven separate messages. Each message that a sensor node sends or receives requires extra energy, which the sensor may not have.

5 Proposed Protocols

5.1 Modified PPK Protocol

We have developed a *Modified Protocol 1*, where we use elliptic curves instead of RSA. We have extended some of the techniques found in [15] to modify the PPK protocol to use elliptic curves.

The most difficult part of this protocol is mapping A, B, K_{AB} to a random point on the elliptic curve. The function f_1 is used to generate a point. A general procedure for this can be found in IEEE 1363 standard [10] (Appendix A.11.1). The function f_2 is used to generate the new key K_{AB} .

The procedure involves calculating a hash of the key. The hash value is equal to x in the elliptic curve, and thus y can be calculated. The original key chosen between A and B should be generated such that a point can be easily found in the elliptic curve. Otherwise, according to [10] the procedure becomes non–deterministic. We assume that K_{AB} was originally chosen so that finding a point on an elliptic is deterministic. However, if K_{AB} is a password rather than

Modified Protocol 1 PPK protocol with elliptic curve

Shared Information: Generator g of G where $y^2 = x^3 + ax + b$

A	B
$P_1 = r(f_1(A, B, K_{AB}))$ $x \in_R \mathbb{Z}_q$ $t_A = xg$ $m = t_A + P_1 \quad \xrightarrow{m} \quad t_A = m - P_1$	
$P_2 = r(f_1(B, A, K_{AB}))$ $y \in_R \mathbb{Z}_q$ $t_B = yg$ $t_B = m' - P_2 \quad \xleftarrow{m'} \quad m' = t_B + P_2$ $Z_{AB} = xt_B \quad \quad \quad Z_{AB} = yt_A$	
$K'_{AB} = f_2(A, B, m, m', Z_{AB}, P_1)$	

a binary key, then we are not guaranteed that the first hash value can be directly mapped to an elliptic curve.

5.2 Proposed three party protocol

When creating our new protocol we assume m_{AS} and m'_{AS} are messages involved in *Modified Protocol 1*, where the two participants are A and S . There are similar constructs m_{BS} and m'_{BS} for messages between B and S .

Proposed Protocol 1 Janson–Tsudik 3PKDP protocol

$M1$	$A \rightarrow B :$	A, N_A, N'_A, m_{AS}
$M2$	$B \rightarrow S :$	$A, B, N_A, N_B, m_{AS}, m_{BS}$
$M3$	$S \rightarrow B :$	$AUTH_A, MASK_A \oplus K_{AB},$ $AUTH_B, MASK_B \oplus K_{AB}, m'_{AS},$ m'_{BS}
$M4$	$B \rightarrow A :$	$AUTH_A, MASK_A \oplus K_{AB},$ $[N'_A, N'_B, B]_{K_{AB}}, N'_B, m'_{AS}$
$M5$	$A \rightarrow B :$	$[N'_A, N'_B, A]_{K_{AB}}$

When the node S obtains $M2$, it will be able to calculate new values for K_{AS} and K_{BS} . It can use these new values to create the $AUTH$ and $MASK$ constructs. Once the protocol is finished, these new keys can be discarded. Thus, if ever S was compromised, than the value K_{AB} will not be compromised.

6 Analysis and Comparison

We first carry out a performance comparison between the Lin protocol and our *Proposed Protocol 1*. The values for the Lin protocol were obtained directly from their publication [13], except that we have also included the number of messages sent by each sensor node. If we converted the Lin protocol to use elliptic curves, there would still be one

more ECC multiplication performed on both sensor nodes A and B , as well as extra random number calculations (for the mapping of the key to an elliptic curve point). As seen from Table 1, our proposed protocol has less number of messages and less MAC calculations. However, the proposed protocol does require more random number calculations. The number of encryption and decryption operations are the same for both protocols. The flexibility of our protocol allows us to convert the encryption and decryption operations into generating MACs. In this case the number of MACs generated will be the same between the two protocols, and the proposed protocol will not need any encryption or decryption operations.

Table 1. Performance Comparison Lin vs. Proposed

	Lin			Proposed		
	A	B	S	A	B	S
messages sent	3	2	2	2	2	1
modular exponentiation	3	3	4	0	0	0
ECC multiplication	0	0	0	2	2	4
en(de)cryption	1	1	2	1	1	2
MACs	3	3	4	2	2	2
random numbers	1	1	2	3	3	2

The *Modified Protocol 1* and *Proposed Protocol 1* were implemented in TinyOS [20] and simulation was run using TOSSIM. David Malan’s [16] implementation of elliptic curve for sensor networks was used. We used the curve $y^2 + xy = x^3 + x^2 + 1$ and reduced polynomial $f(x) = x^{163} + x^7 + x^6 + x^3 + 1$. The simulation time for the two party was 13.38 seconds. The simulation time for the three party protocol was 17.92 seconds. The simulation results for the two protocols are shown in Table 2, where energy comparisons are made between the sensor nodes and the protocols. All the values are given in microjoules.

Table 2. Simulation results

ID	ECC			ECC & SKIPJACK		
	radio	cpu	total	radio	cpu	total
A	78.56	31.54	110.10	131.68	52.89	184.57
B	80.10	32.16	112.26	117.72	47.27	164.99

A memory distribution comparison of our implementation of *Proposed Protocol 1* is shown in Table 3. The .bss and .data segments use SRAM, while the .text segment uses ROM. When we used elliptic curve cryptography the size

of the application for both ROM and RAM has doubled in size.

Table 3. Memory Overhead In Bytes On MICA2 Platform

Memory	ECC	No ECC
ROM	40130	25470
RAM	1478	761
.data	264	264
.bss	1214	497
.text	39866	25206

7 Conclusions

We have proposed novel key establishment protocols for both two party and three party scenarios. The proposed protocols combine traits from a two party password-authenticated key exchange protocol and a symmetric key server-based key establishment protocol. The proposed protocols do not require traditional encryption to transport the new session key. The proposed protocols can be extended to use human readable passwords instead of binary keys. A comparison study was performed, showing the frequencies of different cryptographic operations. We implemented the protocols and compared the energy consumption of the nodes for both the two party and three party scenarios. The impact on memory by adding elliptic curves to a sensor application was analyzed.

References

- [1] V. Boyko, P. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In B. Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 156–171. Springer-Verlag, 2000.
- [2] N. Bulusu. Introduction to wireless sensor networks. In N. Bulusu and S. Jha, editors, *Wireless Sensor Networks: A Systems Perspective*. Artech House, 2005.
- [3] H. Chan and A. Perrig. PIKE: Peer intermediaries for key establishment in sensor networks. In *Proceedings of IEEE Infocom*. IEEE Computer Society Press, Mar. 2005.
- [4] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, page 197, Washington, DC, USA, 2003. IEEE Computer Society.
- [5] W. S. Conner, J. Heidemann, L. Krishnamurthy, X. Wang, and M. Yarvis. Workplace applications for sensor networks. In N. Bulusu and S. Jha, editors, *Wireless Sensor Networks: A Systems Perspective*. Artech House, 2005.
- [6] Y. Ding and P. Horster. Undetectable on-line password guessing attacks. *SIGOPS Oper. Syst. Rev.*, 29(4):77–86, 1995.
- [7] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 42–51, New York, NY, USA, 2003. ACM Press.
- [8] L. Gong. Optimal authentication protocols resistant to password guessing attacks. In *8th IEEE Computer Security Foundations Workshop*, pages 24–29, June 1995.
- [9] L. Gong, M. A. Lomas, R. M. Needham, and J. H. Saltzer. Protecting poorly chosen secrets from guessing attacks. *IEEE Journal on Selected Areas in Communications*, 11(5):648–656, 1993.
- [10] IEEE. Standard specifications for public key cryptography. Technical Report IEEE 1363-2000, IEEE, 2000. <http://grouper.ieee.org/groups/1363/P1363>.
- [11] P. Janson and G. Tsudik. Secure and minimal protocols for authenticated key distribution. *Computer Communications*, 18(9):645–653, September 1995.
- [12] C.-L. Lin, H.-M. Sun, and T. Hwang. Three-party encrypted key exchange: attacks and a solution. *SIGOPS Oper. Syst. Rev.*, 34(4):12–20, 2000.
- [13] C.-L. Lin, H.-M. Sun, M. Steiner, and T. Hwan. Three-party encrypted key exchange without server public-keys. *IEEE Communications Letters*, 5(12):497–499, 2001.
- [14] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 52–61, New York, NY, USA, 2003. ACM Press.
- [15] P. MacKenzie. More efficient password-authenticated key exchange. In D. Naccache, editor, *Topics in Cryptology — CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 361–377. Springer-Verlag, 2001.
- [16] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *Proc. 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pages 71–80, Santa Clara, CA, USA, October 2004. IEEE Computer Society Press.
- [17] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Seventh Annual International Conference on Mobile Computing and Networks (MobiCOM 2001)*, Rome, Italy, July 2001.
- [18] K. Singh and V. Muthukumarasamy. A minimal protocol for authenticated key distribution in wireless sensor networks. In *ICISIP '06: Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing*, Bangalore, India, December 2006. To be published.
- [19] M. Steiner, G. Tsudik, and M. Waidner. Refinement and extension of encrypted key exchange. *SIGOPS Oper. Syst. Rev.*, 29(3):22–30, 1995.
- [20] TinyOS. <http://www.tinyos.net/>, 2006.
- [21] H. Vogt. Exploring message authentication in sensor networks. In *ESAS*, pages 19–30, Heidelberg, Germany, August 2004.
- [22] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, San Francisco, CA, USA, 2004.