Authenticated Key Establishment Protocols for a Home Health Care System

Authenticated Key Establishment Protocols for a Home Health Care System

# Kalvinder Singh [1,2], Vallipuram Muthukkumarasamy [2]
[1] *Australia Development Lab, IBM*
kalsingh@au.ibm.com
[2] *School of Information and Communication Technology, Griffith University*
Gold Coast, 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. Sensor networks can exist in many different environments, and each environment has its own unique characteristics and requirements. As an example application, a home health care system is proposed and examined in detail in this paper. We show how cryptographically weak physiological data can be used to establish keys between body sensors, where the sensors have no other prior secret. This paper also proposes a protocol where a hand held device, such as a PDA, can establish a key with the majority of sensors found in our home health care system. This is achieved without the necessity of using traditional encryption. Detailed analysis of each of the protocols is provided. The protocols were implemented in TinyOS and simulated using TOSSIM and ATEMU. Energy consumption and memory requirements are analysed and it was found that an RSA implementation of our protocols has some advantages over an ECC implementation.

1. Introduction

Wireless sensors and actuators have the potential to significantly change the way people live and interact. As the sensors permeate the environment they can 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 associated costs. For example, a sensor simply detecting light will have different costs to a sensor recording sound. However, less costly sensors can be used to detect a phenomenon before alerting the more expensive sensors to start their monitoring. As the number of heterogeneous sensors increases, so will the amount of interactions between the sensors. We propose a number of key establishment and authentication protocols that can be successfully used in a Wireless Sensor Network (WSN).

There are many different types of sensor environments, ranging from large areas covered by sensors, to many sensors in a small area [1]. Different environments have a wide range of varying characteristics. For instance, sensors placed in large open area are not as physically secure as sensors implanted in an individual's body. The protocols proposed in this paper are mainly designed for our home health care system, although these protocols can also be applied to other environments that have similar security characteristics.

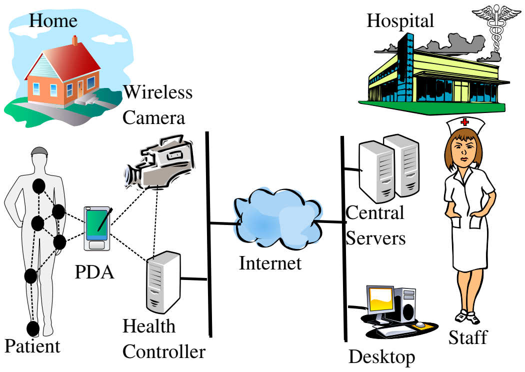

Fig. 1: Proposed Home Health Care System

Context awareness is an important aspect of body sensor networks [2]. For instance, blood pressure increasing due to exercise is normal. But if the blood pressure increases while at rest then that could mean a serious medical condition. Sensors may not just measure the physiological values, but also the body motions, and can lead to a number of different sensors needing to communicate with each other.

Figure 1 gives a diagrammatic representation of our proposed home health care system. The diagram shows a patient at home with a number of body sensors that can communicate with a camera sensor, the health controller, and a PDA. The cameras may only start recording if the body sensors detect that there may be a medical emergency, such as the patient lying horizontal in the kitchen. Surveillance software, such as S3 [3], can be used to detect if the patient is cleaning the kitchen, or getting something from the ground, or there actually is an emergency. If the software does detect an emergency, the hospital staff are notified, examine the information, and decide on the best course of action. The PDA is used to give feedback to the patient about the condition of their body, as well as the status of the sensors. The PDA can notify the patient of any detected emergency, allowing the patient

1-4244-1502-0/07/$25.00 © 2007 IEEE

to report back a false alarm if one has occurred. The PDA can be replaced with a mobile phone or any other hand held communication devices.

Some of the data recorded from body sensors include the heart rate, blood pressure, temperature, and blood oxygen level. They require a data rate of around 2 bits per second [4], [5]. However, other information sent with the message, such as the location of the sender node (8 bits), a MAC (we have specified the size to be the same as the size of the physiological data), and a counter to stop replay attacks (32 bits) raises the data rate to around 10 bits per second. This paper, therefore, assumes the data rate of 10 bits per second for body sensors. Another type of sensor is a surveillance camera, and the data rate requirement for video streams [6] is much greater than that of body sensors. A single camera normally requires 1–4 Mbits per second bandwidth. Providing secure data transfer between sensors is a requirement for our home health care system.

A number of researchers have used environmental data as the only source of secret information to establish keys between body sensors [7], [8], [9], [10], [11]. The major benefit of using environmental data is that body sensors can use this information to authenticate that the other sensor is also on the same person and not of another individual. However, these researchers have cited a number of problems with that approach. The problems include the following:

- only cryptographically strong environmental data can be used.
- the environmental values can become compromised, in which case the new session key is also compromised.

These problems limit the use of environmental data for establishing keys to only a few cases.

The other difficulty our system encounters is in the key establishment scheme of the PDA with sensors at home and in the body. It is envisaged that the patient will simply be users of the system, and will not be able to set up security certificates or keys.

In this paper we propose and develop a number of protocols to address these problems. We show that password protocols can be used to establish keys between body sensors, if passwords are replaced by physiological data. A new protocol is developed to allow a patient to connect a PDA to the home health care system thus able to view information about each of the sensors (ranging from cameras to body sensors). The proposed protocol does not require traditional encryption to transport the new session key. We show that the sensor nodes can establish keys even if no previous shared keys exist between them.

2. Notation

This paper will use the notations shown in Table 1 to describe security protocols and cryptographic operations.

Table 1: Notations

Notation	Description
A, B	The two nodes who wish to share a new session key
S	A trusted server
N_A, N_B	Random numbers 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 respectively
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
$\oplus$	Exclusive–or function

3. Problems and Limitations

A sensor network can consist of many different computing devices. Some have more computational power (and memory) than others. A Body Sensor Network (BSN) is a network of wearable heterogeneous sensors [12]. The sensors are spread over the entire body, and monitor and communicate a range of health related data. BSNs are used in the health industry to monitor a patient's physical and biochemical parameters continuously, in different environments and locations where ever the patient needs to go. BSNs can also be used by athletes to measure their performance. Another use for BSNs is controlling characters in video games [12]. Health information collected from sensors needs to be secured and in some countries, for example the USA, security is mandated [13].

Key establishment protocols are used to set up shared secrets between sensor nodes, especially between neighbouring nodes. When using symmetric keys, we can classify the key establishment protocols in WSNs into three main categories: Pair–wise schemes; Random key predistribution schemes; Key Distribution Center (KDC). The Pair–wise schemes and Random key predistribution schemes are designed for open environments, where there can be many individual sensors [14]. A difficulty with the above schemes is that updating the keys between the nodes is still an unsolved problem. Another drawback is that, when using the random key predistribution schemes, the shared keys cannot be used for entity authentication, since the same keys can be shared by more than a single pair of nodes [15]. The KDC mechanisms by themselves are not suitable for large scale WSN environments, although combinations of a KDC mechanism and the previously mentioned schemes have created hybrid protocols [16]. Some of the limitations with using a sensor node as a KDC mechanism are:

- The KDC scheme relies upon other schemes to create the trusted intermediary.
- The key sizes in sensor nodes are not large enough, so over a period the key between the sensor and the trusted intermediary may become compromised. If the KDC protocol messages were captured and saved by an adversary, then the adversary may calculate the new keys created.
- Some sensor networks may not need an encryption algorithm, although KDC protocols require an encryption

algorithm to encrypt the new key.

A password has been proposed as a way to initiate key establishment [17]. However, the use of a PIN code or a password is not applicable to BSNs since many of the sensors do not have a user-interface. Sensors also may be placed in hard–to–reach places, with some of the sensors implanted into the body. To complicate matters, the sensors may harvest energy directly from the body [18], thus allowing the sensors to exist for long periods of time. Updating keys is therefore an important function.

This paper uses the generic name *Secure Environmental Value* (SEV) referring to sensed data that can be obtained by sensors in an environment. The SEV is usually hard to obtain through other means. Examples of an environment where SEVs may be found include:

- Human body, where it is difficult to attach a device on the body without the knowledge of the person.
- A secured location, for instance a military base or unmanned vehicle, such as UAVs, or a secure home environment.
- Hard to reach places, for instance a satellite in orbit.

The example environment used in this paper is the human body, where BSNs have been developed to measure the physiological values found in individuals [12]. Health sensors can use Inter–Pulse–Interval (IPI) [8] or Heart Rate Variance (HRV) [9] as good sources for cryptographically random numbers and the physiological values can be used as a one–time pad. Protocols [11], [10] have been developed that used these physiological values to encrypt a new key between a sensor pair. For instance, Venkatasubramanian and Gupta[11] used a single message to send a new key to the neighbouring sensor node, as shown in *Protocol 1*.

Protocol 1 Venkatasubramanian BSN protocol

$$A \rightarrow B : N_A, [N_A]_{RANDKEY}, RANDKEY \oplus SEV$$

The new key $RANDKEY$ is encrypted with the physiological value SEV, which is only known to sensors on a particular person. Sensor node B validates that $RANDKEY$ is correct by verifying the MAC of N_A.

Venkatasubramanian and Gupta noted that finding additional cryptographically sound physiological values is still an open research problem. Only cryptographically strong physiological values, such as IPI and HRV, can be used. Also, modern wireless technology (ultra wideband – UWB, radar [19]) may be used to remotely capture the heart rate. It may encounter security risks when only using IPI and HRV to secure the communication. Other cryptographically weaker physiological values, such as blood pressure, and iron count, are less susceptible to those remote attacks.

4. Establishing Keys between Body Sensors

Even though PINs and passwords may not be used in body sensors, we show that password protocols can be used. Passwords have low randomness, and therefore have similar characteristics to many SEVs. A four digit PIN contains less than 14 bits of randomness and can be used in a password protocol. A typical password length of eight characters has less than 48 bit of randomness, if we randomly choose upper and lower case letters as well as the digits 0 to 9. We investigate the suitability of password protocols for the sensor environment. Password protocols have the special property of allowing secrets with small entropy to be used for key establishment. Password protocols are designed so that both off–line and on–line attacks are not feasible. A feature or by–product of most password protocols is that if the password is compromised, then any keys created before the password was compromised will not be compromised.

Key sizes in sensor networks are small, normally 64 bits, so that the encryption or integrity tests do not consume a large amount of energy [20]. Small key sizes lead to the need to update keys on a regular basis.

Researchers have shown that password protocols can be implemented in sensor networks [17]. The password protocols were using either a human–entered password on the sensor, or an existing 64 bit key. Instead we propose that environmental data can be used in the absence of using small keys or passwords. Also, previous approaches for sensor environments only used elliptic curve cryptography. However, RSA password protocols that can be converted to ECC have large exponent such as 1024 bits. When the protocol is converted to use ECC, then 160 bit arithmetic is required.

We instead investigated the EKE protocol, which is an RSA based password protocol where the exponent only needs to be 160 bits [21]. The EKE protocol is chosen because other variants of password protocols require exponents of size 1024 bits. The EKE protocol is diagrammatically shown in *Protocol 2*. A drawback of the EKE protocol is that it cannot use ECC [22].

Protocol 2 Diffie–Hellman–based EKE protocol

Shared Information: Generator g of G where $p - 1 = qr$

$$\begin{array}{ccc}
A & & B \\
r_A \in_R \mathbb{Z}_p & & \\
t_A = g^{r_A} & \xrightarrow{A,[[t_A]]_{SEV_1}} & r_B \in_R \mathbb{Z}_p \\
 & & K_{AB} = t_A^{r_B} \\
 & & t_B = g^{r_B} \\
K_{AB} = t_B^{r_A} & \xleftarrow{[[t_B]]_{SEV_2},[[n_B]]_{K_{AB}}} & \\
 & \xrightarrow{[[n_A,n_B]]_{K_{AB}}} & \text{Verify } n_B \\
\text{Verify } n_A & \xleftarrow{[[n_A]]_{K_{AB}}} &
\end{array}$$

The EKE protocol contains four messages. Node A sends the first message to node B, the message contains the location of A (the location value is in the clear), and the first part of Diffie–Hellman, t_A, is encrypted by the weak key SEV_1. After the first message is sent, node B will calculate the second part of the Diffie–Hellman scheme and hence be able to calculate the session key K_{AB}. Node B then sends the second part of the Diffie–Hellman scheme encrypted by the weak key SEV_2 to node A. The nonce n_B is also sent, encrypted by the session key K_{AB}. The last two messages authenticate both A and B, as well as confirming that they have the session key K_{AB}. The encryption of t_A, t_B, n_A, and n_B can be implemented with an exclusive–or function, as originally described by [21].

Depending on which environmental value is measured, and how long the protocol will run, different SEVs may be used for the request and response. However, if the SEV stays constant throughout the running of the protocol, then both SEV_1 and SEV_2 will be the same. The EKE protocol is designed for a constant password throughout the running of the protocol, so similar or same data for both SEV_1 and SEV_2 will not adversely affect the protocol.

The EKE protocol was originally designed to handle small entropy secrets, so that off–line and on–line dictionary attacks are infeasible for an adversary. Another useful feature is that even if the secrets SEV_1 or SEV_2 are compromised or available freely after the running of the key establishment protocol, the session key K_{AB} will remain secure and safe.

Both nonces n_A and n_B are cryptographically strong random numbers, allowing the exclusive–or function to be used for encryption. If any nonce was not cryptographically strong then either $n_A \oplus K_{AB}$ or $n_B \oplus K_{AB}$ operation would allow an adversary to significantly reduce the number of valid K_{AB} values. A characteristic of the EKE protocol is that the nonces are never sent out in the clear, since the nonces are used to encrypt the new key K_{AB}.

The value of p should be chosen wisely [21].The value of p should be as close to $2^N - 1$ as possible for the best security.

Even though exclusive–or and block cipher symmetric cryptography is suitable in an RSA environment, it is not suitable when converting to elliptic curves [22].The EKE (RSA) protocol is compared with a ECC based password protocol [17].

Using the RSA implementation [23] from the Deluge system and porting it to the mica2 mote system, and only using 160 bit exponents, we found that the total number of cycles is 147879. Password protocols that can use an ECC implementation inherently require 1024 bit exponents when in RSA mode [22]. When measuring the number of cycles by using the ECC implementation for sensors [24], including an implementation of the square root function, we get a total number of 18790689. The key size was 160 bits, which is equivalent to 1024 bits in RSA. When moving to the ECC protocols, more secure keys are required. There is a significant number of extra cycles in a ECC implementation over the RSA implementation.

We also examine the memory of the application, as shown in Table 2. The combination of *.bss* and *.data* segments use SRAM, and the combination of *.text* and *.data* segments use ROM. The *.text* contains the machine instructions for the application. The *.bss* contains uninitialized global or static variables, and the *.data* section contains the initialized static variables.

We used the values provided by the TOSSIM simulator (a part of the TinyOS installation) to obtain an indication of the power consumption when sending a message. In our calculations we do not take into account any collision avoidance times. On the mica2 mote, the cost of sending an extra 20 bytes is 28.1 microjoules. There is a substantial startup cost for each message sent, and then there is an added cost for every bit that is sent.

TABLE 2: MEMORY OVERHEAD IN BYTES ON MICA2 PLATFORM

Memory	RSA	ECC
ROM	1942	9720
RAM	177	859
.data	60	8
.bss	117	851
.text	1882	9712

5. SECURELY CONNECTING THE PDA

When a patient starts up a PDA or obtains a new PDA. Keys need to be created with the PDA and the sensors (both body and camera sensors). The mechanism we use is that the patient, when starting up the PDA, will need to enter either a password or PIN. The remainder of this section discusses a protocol that can be efficiently used to establish keys with all the sensor devices in the household and one of the sensors on the body (normally the body control sensor).

We assume that the body sensors, especially any body sensors that are leader nodes, have obtained the session keys from other sensors in the house. It is envisaged that when the controller sensor was added to the body, it had embedded a key with a central server. Then by using a KDC protocol, it can obtain keys with all the other sensors in the house. When the body sensors establish or update keys between themselves, they can use the password protocols (as described earlier in this paper). However, a hand held device, such as a PDA or mobile phone, may be purchased from a local store and will not have any keys. If patients are required to set up certificates or keys themselves, then the security system may be set up incorrectly. Also, if the device becomes lost or stolen, then an adversary is able to physically obtain any long–term keys held on the device. Our solution is to propose a multiple server protocol that can create session keys between the sensors in the house and the body controller sensor, with the PDA.

A multiple server protocol has previously been developed for normal sensor networks [25]. The main reason for a multiple server protocol is that if sensors exists in an open environment, then KDC nodes can be physically compromised. In our sensor environment, the sensors are less likely to be physically compromised (either they be sensors implanted in the body, or the cameras placed in the home). However, multiple server protocols are still important for a home health care system. The reasons for a multiple server protocol are:

- Increase the randomness of the new key, by having multiple parties adding randomness to the new key.
- A camera may break down, or run out of power. A multiple server protocol increases the availability of the key establishment service.

- The key between the camera and the sensor may become compromised. The keys used by the sensors are normally small in size, since cryptographic algorithms consume more energy when larger keys are used.

In our attempt to create an efficient multiple server protocol, we specified n servers where each server corresponds to a camera. The *Proposed Protocol 1* shown below, represents each of the cameras as S_i. The PDA device is labelled as A, and sends the first message, A, B, N_A, to each of the cameras. Each camera sends their message to back to the PDA. The PDA will calculate the keys K_{AS_i} and the cross checksums, and sends the cross checksums as well as parts of the messages from the cameras to the body sensor. The body sensor creates its own cross–checksums and compares them against the cross–checksums created by the PDA. At this stage, the keys K_S and K_{AB} are created by the body sensor. The body sensor sends N_B, the keying data, and the its newly created cross–checksums to the PDA. The PDA can now also create the keys K_S and K_{AB}. The final message completes the key confirmation between the PDA and the body sensor, as shown in *Proposed Protocol 1*. If key confirmation is not vital, then the final message can be removed.

The *Proposed Protocol 1* provides key authentication, key freshness and key confirmation, using multiple authentication servers. In our *Proposed Protocol 1*, the following constructs are used: π is the password or SEV, A is the PDA, B is a body sensor, S_i is a camera, t_{AS_i} and t_{S_iA} are the Diffie–Hellman values, $m_{AS_i} = [[t_{AS_i}]]_\pi$, $m'_{AS_i} = [[t_{S_iA}]]_\pi$, $AUTH_{Ai} = [A, B, K_i]_{KAS_i}$, $MASK_{Ai} = [[AUTH_{Ai}]]_{K_{AS_i}}$, $AUTH_{Bi} = [A, B, K_i]_{K_{BS_i}}$, and $MASK_{Bi} = [[AUTH_{Ai}]]_{K_{BS_i}}$

Proposed Protocol 1 A Preliminary Multiple Server Protocol

$M1$	$A \to S_i:$	m_{AS_i}, A, B
$M2$	$S_i \to A:$	$m'_{AS_i}, S_i, AUTH_{Ai}, MASK_{Ai} \oplus K_i,$ $MASK_{Bi}, AUTH_{Bi} \oplus K_i$
$M3$	$A \to B:$	$S_1, MASK_{B1}, AUTH_{B1} \oplus K_1, \ldots,$ $S_n, MASK_{Bn}, AUTH_{Bn} \oplus K_n,$ $cc_A(1), \ldots, cc_A(n), N_A, A$
$M4$	$B \to A:$	$cc_B(1), \ldots, cc_B(n), N_B, [N_A]_{K_{AB}}$
$M5$	$A \to B:$	$[N_B]_{K_{AB}}$

The PDA, the body sensor and the cameras contribute to the key value. The values N_A and N_B are generated by the PDA and the body sensor respectively as input to the MAC function, that determines the session key. The key used with the MAC function is generated by the servers. Both the PDA and body sensor compute the session key as $K_{AB} = [N_A, N_B]_{K_S}$. The keys K_{AS_i} are generated by computing the diffie–hellman part of the protocol. The PDA and body sensor should have a minimum number of cameras returning valid results before confirming that the key is valid. The PDA will calculate $cc_A(i)\ \forall i \in 1, \ldots, n$.

$$cc_A(i) = \begin{cases} [K_i]_{K_i} & \text{if valid,} \\ EM & \text{otherwise} \end{cases} \tag{1}$$

Where EM is an error message; an example will be the value zero. There is a remote chance a valid case may be zero. If the valid value is zero, the camera needs to be considered a compromised server (even though it is not a malicious server).

The body sensor will calculate $cc_B(i)$, and compare it with $cc_A(i)$. If they are the same, then the server S_i is valid. Below is a way the PDA and body sensor compare the cross checksum for $cc_A(i)$ and $cc_B(i)$.

$$cc_B(i) = \begin{cases} cc_A(i) = [K_i]_{K_i} & \text{if valid,} \\ EM & \text{otherwise} \end{cases} \tag{2}$$

After the comparison of the entire cross checksums, a set of valid keys $V_1, \ldots, V_m$ should remain. The creation of K_S is defined as follows.

$$K_S = V_1 \oplus \ldots \oplus V_m \tag{3}$$

Where V_i is the i^{th} valid key given by a server, and m is the total number of valid servers $t \leq m \leq n$, where t is the minimal number of trusted servers. Another advantage of the proposed protocol is that the cameras will not be able to calculate K_S. The calculated $cc_B(i)$ values are returned to the PDA, where the PDA performs similar checks as the body sensor and calculates K_S.

Once the PDA has established a key with one of the body sensors, then a KDC protocol can be used to establish keys with the other body sensors.

6. Analysis and Discussion

Our *Proposed Protocol 1* has a number of advantages, one of which is that the body sensor does not need good random number generators to create the nonces. The body sensor could even safely use a counter for their nonce values. Another advantage is that if a camera or a number of cameras are unavailable, the authentication service itself still exists through the working cameras. If one or more cameras become compromised, the authentication service or the security of the system is not compromised.

The proposed protocol only encrypts random information. If the encryption cipher uses an IV value (such as RC5 and SKIPJACK currently used in TinyOS [26]) then we can use a constant IV value. However, the constant IV value chosen for our protocol must only be used to encrypt the random data and should never be used to encrypt other information. Also, a wide variation of different ciphers can safely be used.

Some MACs have vulnerabilities when the message sizes are variable. All of our message sizes are of constant value, allowing us to safely use a wider range of MACs than previously available. The size of the MACs sent to the body sensor can be lower than that of conventional protocols. The integrity checking is performed by the body sensor. If x is the size of the MAC in bits, then an adversary has 1 in 2^x chance of blindly forging a valid MAC for a particular message. The adversary should be able to succeed in 2^{x-1} tries. Because of the low bandwidth of sensor nodes, a 4 byte MAC, requiring 2^{31} packets, will take years to complete. If an adversary did attempt this attack, the sensor node would be non–functional within that period. In addition, an adversary will need to forge $2t$ MACs; t MACs to A and t MACs to B, and stop traffic from the other base stations before they can determine the value of K_{AB}.

In the proposed protocol, the device that is most sensitive to energy restrictions is the body controller sensor. The message $M3$ is of the most concern, since it the largest message sent to the controller sensor. We calculate the size of the message as $M3 = (n+1)a_0 + a_1 + na_2 + na_3$ bytes. Where a_0 is the size of the location, a_1 is the nonce size, a_2 is the key size, a_3 is the MAC size, and n is the number of cameras. Assuming that the location is 1 byte in size (maximum 256 possible sensors), the nonce is 1 byte in size, the key is 8 bytes in size, and the MAC is 4 bytes in size, we get $M3 = 13n + 2$ bytes. If we assume that a packet size is 28 bytes, a configuration with more than two cameras will require multiple packets sent between the PDA and the body controller node. If there is no or little concern about whether the cameras or the camera keys are compromised, then the PDA can select two cameras to send to the body controller sensor.

The computational complexity for the body sensor depends on the number of valid servers the PDA forwards to the sensor, the number is defined as m. The computational cost of the MACs is $4m + 2$, and the cost of the encryption operations is m. The number of exclusive–or operations is $2m$.

7. Conclusions and Future Work

We have proposed a multi–server key establishment protocol that allows a PDA to obtain session keys with most of the sensors in our home health care system. We implemented salient features of the password protocols and compared the energy consumption of the nodes. The password protocols that could be converted to use ECC had a larger computational overhead than the EKE protocol, because of the stronger keys required by the ECC–based password protocols. Due to the EKE protocol only requiring 160 bit exponents, the message sizes of the EKE protocol were comparable to the ECC–based password protocols. The impact on memory by adding elliptic curves to a sensor application was analyzed, revealing that there is additional costs associated with an ECC solution over a RSA solution. Future work includes using cryptographic protocol verifiers to confirm that our protocols are secure.

References

[1] M. Kuorilehto, M. Hännikäinen, and T. D. Hämäläinen, "A survey of application distribution in wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 5, no. 5, pp. 774–788, 2005.

[2] S. Thiemjarus and G.-Z. Yang, "Context–aware sensing," in *Body Sensor Networks*, G.-Z. Yang, Ed. Springer–Verlag, 2006.

[3] A. Hampapur, L. Brown, J. Connell, N. Haas, M. Lu, H. Merkl, S. Pankanti, A. Senior, C.-F. Shu, and Y. Tian, "S3-r1: the ibm smart surveillance system-release 1," in *ETP '04: Proceedings of the 2004 ACM SIGMM workshop on Effective telepresence*. New York, NY, USA: ACM Press, 2004, pp. 59–62.

[4] T. Balomenos, "User requirements analysis and spcification of health status analysis and hazard avoidance artefacts," DC FET Project ORESTELA, Delieverable D02, Tech. Rep., 2001.

[5] E. Yeatman and P. Mitcheson, "Energy scavenging," in *Body Sensor Networks*, G.-Z. Yang, Ed. Springer–Verlag, 2006.

[6] Axis, "Setting up an ip–surveillance system using axis cameras and axis camera station software," http://www.axis.com/files/manuals/gd_ipsurv_design_en_070320.pdf, March 2007.

[7] S.-D. Bao, Y.-T. Zhang, and L.-F. Shen, "A design proposal of security architecture for medical body sensor networks," in *BSN '06: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 84–90.

[8] C. C. Y. Poon, Y.-T. Zhang, and S.-D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m–health," *IEEE Communications Magazine*, vol. 44, pp. 73–81, April 2006.

[9] S.-D. Bao, Y.-T. Zhang, and L.-F. Shen, "Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems," in *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005*. IEEE Press, 2005, pp. 2455–2458.

[10] S.-D. Bao and Y.-T. Zhang, "A new symmetric cryptosystem of body area sensor networks for telemedicine," in *6th Asian–Pacific Conference on Medical and Biological Engineering*, 2005. [Online]. Available: http://ifmbe-news.iee.org/ifmbe-news/july2005/shudibaopaper.html

[11] K. K. Venkatasubramanian and S. K. S. Gupta, "Security for pervasive health monitoring sensor applications," in *ICISIP '06: Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing*. Bangalore, India: IEEE Press, December 2006, pp. 197–202.

[12] O. Aziz, B. Lo, A. Darzi, and G.-Z. Yang, "Introduction," in *Body Sensor Networks*, G.-Z. Yang, Ed. Springer–Verlag, 2006.

[13] USA, "Summary of hipaa health insurance probability and accountability act," US Department of Health and Human Service, May 2003.

[14] D. Liu and P. Ning, *Security for Wireless Sensor Networks*, S. Jajodia, Ed. Springer Berlin / Heidelberg, 2007.

[15] P. Hämäläinen, M. Kuorilehto, T. Alho, M. Hännikäinen, and T. D. Hämäläinen, "Security in wireless sensor networks: Considerations and experiments." in *SAMOS*, 2006, pp. 167–177.

[16] 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.

[17] K. Singh, K. Bhatt, and V. Muthukkumarasamy, "Protecting small keys in authentication protocols for wireless sensor networks," in *Proceedings of the Australian Telecommunication Networks and Applications Conference*, Melbourne, Australia, December 2006, pp. 31–35.

[18] A. Kansal and M. Srivastava, "Energy–harvesting–aware power management," in *Wireless Sensor Networks: A Systems Perspective*, N. Bulusu and S. Jha, Eds. Artech House, 2005.

[19] E. M. Staderini, "Uwb radars in medicine," *IEEE Aerospace and Electronic Systems Magazine*, vol. 21, pp. 13–18, Janurary 2002.

[20] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 162–175.

[21] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society Press, 1992, pp. 72–84. [Online]. Available: citeseer.ist.psu.edu/bellovin92encrypted.html

[22] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, U. Maurer and R. Rivest, Eds. Springer Berlin / Heidelberg, 2003.

[23] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler, "Securing the deluge network programming system," *In the Fifth International Conference on Information Processing in Sensor Networks (IPSN'06)*, April 2006.

[24] A. Liu, P. Kampanakis, and P. Ning, "Tinyecc: Elliptic curve cryptography for sensor networks (version 0.3)," February 2007, http://discovery.csc.ncsu.edu/software/TinyECC/.

[25] K. Singh and V. Muthukkumarasamy, "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: IEEE Press, December 2006, pp. 78–83.

[26] TinyOS, "An operating system for sensor motes," http://www.tinyos.net/, 2007.