Key Establishment Using Group Information for Wireless Sensor Networks

William R. Claycomb[1], Rodrigo Lopes[1], Dongwan Shin[1], and Byunggi Kim[2]

[1] Secure Computing Laboratory, New Mexico Tech,
801 Leroy Place, Socorro, NM, USA, 87801
{billc,rodrigo,doshin}@nmt.edu

[2] College of Information Technology, Soongsil University,
511 Sangdo-dong, Dongjak-gu, Seoul 156-743, Korea
bgkim@comp.ssu.ac.kr

Abstract. Wireless sensor networks are commonly used for critical security tasks such as intrusion or tamper detection, and therefore must be protected. To date, security of these networks relies mostly on key establishment and routing protocols. We present a new approach to key establishment, which combines a group-based distribution model and identity-based cryptography. Using this solution enables sensor nodes to authenticate each other, and provides them with a structure to build secure communications between one another, and between various groups. Using our key establishment protocol, we show how to reduce or prevent significant attacks on wireless sensor networks.

Keywords: Group-based security, key establishment, wireless sensor networks.

1 Introduction

Wireless sensor networks (WSNs) have become commonplace in applications ranging from simple light, temperature, and sound measurements to sophisticated military and industrial applications. In some cases the security of the sensors, the data collection, and the communication of that data to a collection point is unimportant. However, for sensitive applications, the security of these tasks is paramount.

One of the core components to securing WSN data is the establishment of secret keys between sensor nodes in a network. In contrast to other mobile network applications, WSN have the property that key establishment information can be pre-loaded onto sensor nodes prior to deployment. However, the limitations of sensor nodes have traditionally limited key establishment and data encryption techniques to simple and symmetric cryptographic applications.

Recently, due to advances in sensor hardware, the field of asymmetric cryptography has become a realistic option for various operations in WSN.Using asymmetric cryptography for key establishment greatly simplifies many of the challenges currently faced by schemes relying on symmetric key establishment protocols. For instance, node authentication is possible. Also, compromising even a large subset of nodes and communication keys does not compromise all communication keys for the network - a goal many of the solutions to date have strived to achieve. However, using asymmetric cryptography does not solve all the problems of WSN security. Specifically, three attacks that still present problems for WSN are *node replication*, *Sybil*, and *wormhole* attacks.

In this paper, we present a novel scheme for key establishment in WSN which combines the notion of *group-based* key predistribution with identity based cryptography (IBC). We show how this approach enables secure key establishment in WSN, and show how our solution improves upon the security of existing key establishment schemes. We discuss considerations in design for this type of WSN, including an analysis of group communication dynamics. We also demonstrate how our solution addresses various attacks on WSNs.

The remainder of this paper is structured as follows. Section 2 describes related work, including a description of identity based cryptography components. Section 3 describes our new key establishment protocol, followed by Section 4 which presents a detailed description of various attacks and how our solution addresses them. Section 5 presents a discussion of group distribution decisions and the impact to sensor nodes. In Section 6, we conclude the paper with a discussion of future work.

2 Background and Related Work

Key establishment in WSNs has been an active area of research for many years, and several approaches have been proposed. The simplest method of key establishment between nodes in a WSN is to preload each node with the same shared secret key, and use that key to secure transmission between every node pair. While this is simple in terms of protocol and storage requirements, it has the disadvantage that compromising a single node reveals all communication to an adversary. The simple solution to prevent this is to pre-load each node with a unique shared key for pairwise communication with every other node in the network. This is a more secure network than the previous one, as compromising a single node only reveals the data transmitted between it and its node partners. However, the storage overhead per node is unrealistic, particularly in the case of very large networks.

Several researchers have developed solutions for pairwise key establishment that fall between these two extremes. Recently, proposals have included the notion of *location-based*, or *group-based* schemes. These approaches observe that in large-scale sensor node distributions, nodes deployed in a group tend to end up near each other physically. By tailoring key predistribution to take advantage of this property, efficient schemes have been presented [1,2,3,4].

However, these solutions rely on symmetric key agreement protocols, and largely ignore the application of asymmetric key agreement methods. It has been commonly believed that the increased computational and time constraints required by asymmetric solutions are too costly for WSNs. Recently, though, researchers have studied implementations of asymmetric protocols such as elliptic curve cryptography (ECC) on sensor nodes [5,6,7,8,9]. Results have been promising, and various proposals for key establishment and key management using ECC-based approaches have been suggested [10].

2.1 Identity Based Cryptography

In [11], the authors propose using an *identity based cryptography* technique called *pairing* [12,13] to facilitate secure key establishment between nodes in WSNs. First proposed by [14], identity based cryptography (IBC), or *identity based encryption*, is similar to traditional public-key based cryptography systems, but instead of using a randomly generated public key, entities use unique strings or other short identifiers, such as email addresses, as their public key. For instance, to send an encrypted email to "bob@company.com," Alice would encrypt the message using the email address as the public key. Bob would obtain his corresponding private key from a *Private Key Generator* to decrypt the message. Therefore, IBC helps negate the use of public key certificate in PKC-based applications.

As one of the critical components to optimizing the performance of WSNs is minimizing data storage and transmission size, IBC is an excellent choice for exchanging key establishment information in this regard. The only thing necessary to begin key establishment is the identity of the source node, which only has to be large enough to ensure a unique ID among the entire network. A 16 bit ID would provide unique IDs for $2^{16} = 65536$ sensor nodes.

Definitions. The following definitions will be used in describing our solution:

p, q	large primes
$\mathbb{E}/\mathbb{F}_p$	An elliptic curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$
$\mathbb{G}_1$	A q-order subgroup of the additive group of points of $\mathbb{E}/\mathbb{F}_p$
$\mathbb{G}_2$	A q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$
$\hat{e}$	A mapping $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
H	A cryptographic hash function which maps to elements in $\mathbb{G}_1$
h	A standard cryptographic hash function
$A, B, ...$	Groups of sensor nodes
$x, y, ...$	Individual sensor nodes
ID_A	Unique ID of group A
ID_x	Unique ID of sensor node x
$K_{x:A}$	Secret key for node x among group A
K_{xy}	Shared pairwise key between nodes x and y
m_A	Master key for group A

The modified Weil pairing scheme in [12] is used for the basis of our key establishment protocol. This allows two sensor nodes to establish a shared secret key using only the knowledge of an individual secret key and the unique identity of the other node, as long as the secret key of the first node was generated using the same keying information as the identity information of the second node. The pairing is based on the *bilinear Diffie-Hellman Problem (DLP)*, which is proven to be computationally hard.

Pairing. A *pairing* is the mapping of two points P and Q, from separate subgroups of points on an elliptic curve, to a third point in a finite field, and is denoted $e(P, Q)$. Usually P and Q are linearly independent of each other, otherwise the pairing is considered degenerate, and $e(P, Q) = 1$. To use P and Q from the same group $\mathbb{G}_1$, a *distortion map* ψ is used to make the second element linearly independent of the first. This *distortion pairing* is denoted $\hat{e}(P, Q) = e(P, \psi(Q))$. The distorted pairing is *symmetric*, that is $\forall P, Q \in \mathbb{G}_1$, $\hat{e}(P, Q) = \hat{e}(Q, P)$. For a more thorough description of pairings over elliptic curves, please see [12,13].

Due to the distortion map, we can use a pairing which is a mapping $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. It has the following properties [12]:

– Bilinear: $\forall P, Q \in \mathbb{G}_1$ and $\forall a, b \in \mathbb{Z}, \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
– Non-degenerate: If P is a generator of $\mathbb{G}_1$ then $\hat{e}(P, P) \in \mathbb{F}^*_{p^2}$ is a generator of $\mathbb{G}_2$.
– Computable: Given $P, Q \in \mathbb{G}_1$, there is an efficient algorithm to compute $\hat{e}(P, Q) \in \mathbb{G}_2$

3 Group-Based Key Establishment

While some node deployment scenarios allow for manual or automated placement of nodes, or for key nodes to have known fixed positions [11], we believe that certain critical applications, such as battlefield deployments, do not provide such means. Therefore, it is desirable to have a solution which provides secure key establishment without requiring specific location knowledge for each node. Instead, we propose that in a group-based distribution, assuming nodes deployed together end up near each other [3,15], that each group of nodes deployed be pre-keyed using key information unique to that group. Doing so would allow nodes containing keying information for a specific group A to establish pairwise keys with nodes from group A. We will show that in order to establish this pairwise key, both nodes must have keying information for group A.

To detail our solution, we propose that a successful deployment of a WSN would consist of the following phases:

– Provisioning authority initialization
– Sensor key initialization (key pre-distribution)
– Sensor deployment
– Pair-wise key establishment

3.1 Provisioning Authority Initialization

A separate provisioning authority (PA) is created for each group of nodes to be deployed. A *master PA* is responsible for generating the pairing information for each PA. This information, as described in § 2.1, is $(p, q, \mathbb{E}/\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e})$. The master PA also selects two hash functions, H and h, where H maps to nonzero elements in $\mathbb{G}_1$, and h is a secure cryptographic hash function. Then, for a group of sensors to be deployed, A, the master PA randomly selects m_A as the master secret for A. This sequence is loosely based on the work of [11], but has been modified to meet the specific needs of our solution.

3.2 Sensor Key Initialization

During this phase, we assume each group has a unique identity ID_A, and every node has a unique identity ID_x. The PA provides every sensor x, in group A, a unique identity-based key $K_{x:A}$. The unique key is calculated as $K_{x:A} = m_A H(ID_x)$. Along with the unique $K_{x:A}$, each node in A is pre-loaded with the public information for group A, $(p, q, \mathbb{E}/\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H, h, ID_A)$. Compromising a node will not reveal the group secret key m_A. This is due to the intractability of the discrete logarithm problem for elliptic curves (ECDLP). That is, consider an elliptic curve E defined over $\mathbb{F}_q$ and $P \in E(F_q)$, which is a point of order n. Given $Q \in E(F_q)$, it is infeasible to find an integer $0 \leq d \leq n$ such that $Q = dP$. Therefore, it is also infeasible to find m_A, given $K_{x:A}$ and ID_x, since $K_{x:A} = m_A H(ID_x)$.

In addition to the unique key for group A, each node x is also pre-loaded with a unique key corresponding to each adjacent group B, which is calculated as $K_{x:AB} = m_A m_B H(ID_x)$. Nodes are also given the identifier for group B, ID_B. At this point, it is possible to pre-load x with additional unique keys and group identifiers for the groups beyond those immediately adjacent to A. The number of adjacent and surrounding group keys loaded on each node in A is determined by the needs of the network, noting that each additional layer of depth i adds 2^{i+2} public keys, if adjacent groups are arranged in a grid, as shown in Figure 1.

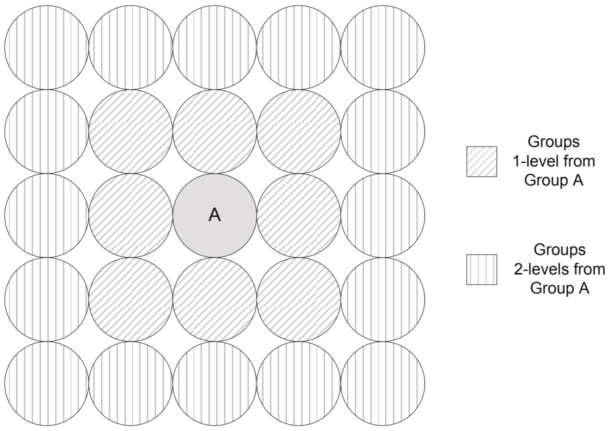

Fig. 1. Layers of sensor network groups

3.3 Sensor Deployment

Sensors are deployed in groups, in a predetermined pattern, over the intended area. No additional bootstrapping is necessary for nodes to begin generating pair-wise keys.

3.4 Pairwise Key Establishment

Pairwise keys between neighboring nodes can be established in one of two ways. The first, *intragroup* key establishment, involves pairwise keys generated between nodes in the same group. The second, *intergroup* key establishment involves nodes generating pairwise keys with nodes in different groups.

For the purposes of our solution, the elements of a pairing $\hat{e}(P, Q)$ are the private keying information of a source node, $K_{x:A}$, and the public keying information (identity) of the destination node, ID_y. Because of the symmetry of pairing, the private keying information of the destination node, $K_{y:A}$, and the public keying information (identity) of the source node, ID_x, will also produce the same value, as long as all elements originate in $\mathbb{G}_1$, due to the symmetry of the pairing.

Intragroup key establishment. To establish pairwise keys with neighboring nodes in the same group, node x broadcasts its identity ID_x, its group identity ID_A, and a random nonce n_x, and waits for responses from neighboring nodes within range. A node y receiving the broadcast from x first checks to see if it is in the same group A. If so, node y responds with its own identity ID_y, another random nonce n_y, and a verification of a shared key, $h_{K_{yx}}(n_x||n_y||1)$, where $K_{yx} = \hat{e}(K_{y:A}, H(ID_x))$. Because this is intragroup communication, the group ID ID_A does not need to be returned. The lack of group ID in the return message is a trigger to x that the return key information in intragroup. In this case, x is able to verify the identity of y, because $K_{yx} = K_{xy}$, due to the pairing symmetry. Node x responds with a verification message $h_{K_{xy}}(n_x||n_y||2)$, where $K_{xy} = \hat{e}(K_{x:A}, H(ID_y))$. This verifies to y that x has established the correct key.

Intergroup key establishment. To establish a pairwise key with a node in another group, B, node x follows a similar protocol. The initial broadcast step is the same: ID_x, the group identity ID_A, and a random nonce n_x. Upon receiving this message, node y, in group B, recognizes that x is in another group, A. y checks to see if it possesses a key corresponding to A, $K_{y:BA}$. If so, y responds with its own identity ID_y, its group identity, ID_B, another random nonce n_y, and a verification of a shared key, $h_{K_{yx}}(n_x||n_y||1)$, where $K_{yx} = \hat{e}(K_{y:BA}, H(ID_x))$. When x receives this message, it sees that a group identity was returned. x checks to see if it possesses a key corresponding to B, $K_{x:AB}$. If so, x responds with a verification message $h_{K_{xy}}(n_x||n_y||2)$, where $K_{xy} = \hat{e}(K_{x:AB}, H(ID_y))$. This verifies to y that x has established the correct key.

4 Attacks and Resistance

Various attacks on WSNs have been described in the literature. Some of the more difficult to defend against are the node replication, node addition, Sybil, and wormhole attacks. Our solution either prevents these attacks from occurring, or minimizes their impact to the point of inefficacy.

4.1 Node Replication Attack

A *node replication* attack [16] consists of an adversary compromising one or more existing nodes, duplicating (and perhaps modifying) the information retrieved, provisioning new nodes with the stolen information, and placing the replicated nodes into the network. Under simple authentication schemes, a network is unable to detect the duplicated nodes, particularly if they are placed far apart, and they are able to authenticate and interact with existing nodes. This provides the attacker with a means of intercepting communication, affecting network routing, or even skewing reputation-based trust management systems.

Our solution eliminates the node replication attack over multiple groups. To show this, consider a compromised node x, from group A. If placed in a group where nodes do not share keying information with A, the replicated node (denoted x') would be unable to establish any pairwise keys. To be successful, an attacker could only place x into a group where the nodes have shared key information with A. In a large network, the number of such groups is likely to be small. Additionally, placing x' into one of these groups would be likely to cause detection, as the real x is nearby, and nodes receiving another authentication request from a duplicate node proclaiming to be x would detect the attack.

4.2 Node Addition Attack

In a *node addition* attack [17], an adversary injects new nodes into a WSN. This differs from a node replication attack because the new nodes are not duplicates of an existing node, but rather are provisioned correctly as unique nodes in the network. To carry out this attack in a network with pairwise key establishment, an adversary must have access to keying information, such as a master key (m_A in our solution.) While some solutions allow the master key to exist for a short period of time on individual nodes during the distribution of a new network [11,17], ours does not allow m_A to exist on any node or base station in the network. Without m_A, an attacker can only gain information such as ID_x, $K_{x:A}$, ID_A, etc. Even with the pairing components $(ID_x, K_{x:A})$, m_A cannot be deduced, due to the properties of the pairing scheme.

4.3 Sybil Attack

The *Sybil attack* [18] describes a situation where an attacker generates multiple fake identities in one location, appearing to be many legitimate nodes, and uses those identities to influence the existing WSN. This influence can be against routing, trust management, reputation, or other elements of network management. In contrast to the node replication attack, the nodes in the Sybil attack are not real nodes, nor are they distributed throughout the network.

Our solution prevents this attack by preventing an attacker from spoofing legitimate nodes. Without the master key to generate pairing information, an attacker cannot generate correct pairing information. Without this information, a Sybil node cannot establish pairwise keys with any legitimate node in the network.

4.4 Wormhole Attack

One of the more interesting attacks in WSNs, the wormhole attack involves an attacker using a covert, high-speed side channel to tunnel information across the network. The effect is to allow two distant nodes to communicate with each other, fooling them into believing they are neighbors. This can disrupt network routing, potentially preventing critical messages from being received by the correct locations. It can also affect networks which use relative node location for decision-making. The wormhole attack is shown in Figure 2.

Detecting and preventing wormhole attacks are challenging issues. Most solutions to date involve the use of mechanisms based on the physical location of nodes, or the physical distance between them. These solutions use either location-fixing devices, such as GPS, or tight time synchronization between nodes to detect wormholes [11,19,20].

Our solution prevents wormhole attacks between nodes in distant groups because they do not share key information, and cannot form pairwise keys, as shown in Figure 3. However, a wormhole between two nearby groups is possible. Creating a wormhole for nodes within adjacent groups would have a limited effect, and would be physically more difficult, because creating a high-speed side-channel and processing information through it faster than the normal network routing becomes tricky as the distance between the wormhole ends grows smaller.

With this in mind, a network designer can carefully plan node group size to reduce the threat of wormhole attacks to an acceptable level. For instance, if an

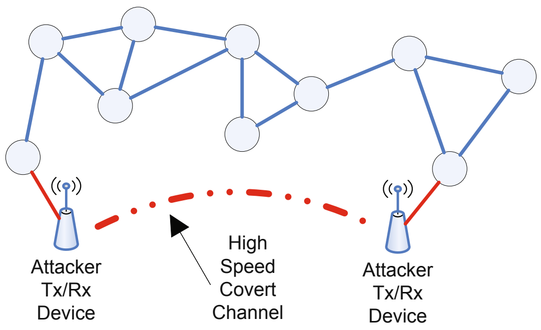

Fig. 2. A wormhole attack

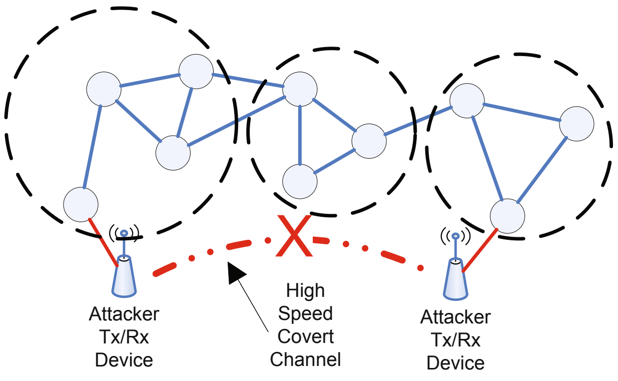

Fig. 3. A wormhole attack prevented in a group-based network

attack between nodes 50 m apart would not compromise the overall integrity of the system, then groups can be distributed with a diameter of 25 m, so the greatest distance between any two nodes allowed to communicate is 50 m.

Wormhole attacks can be prevented by a carefully designed security policy as well. If a policy is designed such that a group A only trusts connections to group B, then a wormhole attack is only possible from group A to group B. Because this is an asymmetric trust, a wormhole attack from B to A is not possible. This is an important step towards preventing a wormhole attack that seeks to “skip” a sensor or group of sensors in a sequence by generating a wormhole around it.

5 Discussion

The discussion of our solution will focus on two areas. The first part deals with the network design. Next, we discuss the impact to sensor nodes themselves.

5.1 Network Design

When looking at a group-based deployment, we must consider how the nodes will be distributed. Will they be distributed evenly over an area, or will they tend to be more densely populated near the center of an area? How does this affect the chances of multi-group communication? We attempt to answer these questions here.

Node distributions are not always uniform. Sometimes the reasons for this are simple physics (i.e. dropping a box of sensor nodes from an airplane at different heights, with different wind patterns, will result in greatly different distributions on the ground). Other times, the reasons for a certain distribution are more sophisticated, such as the desire to prevent a wormhole attack between nodes of a certain distance, n.

To help analyze communication between groups in our solution, we consider two variations of node distribution, one where nodes are distributed evenly over

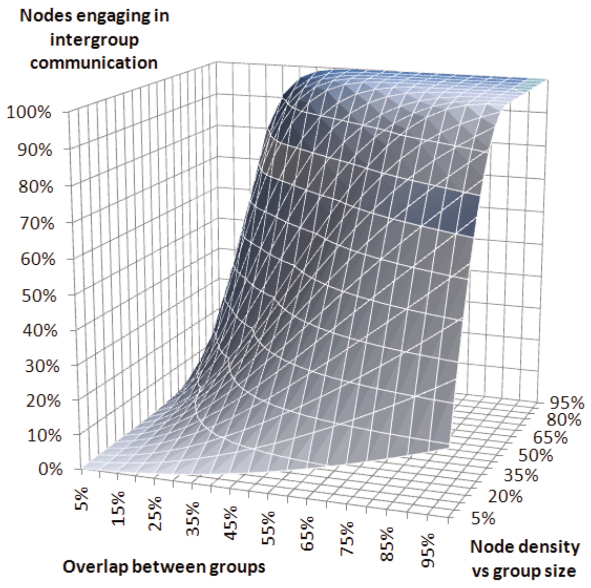

Fig. 4. Nodes able to communicate with other groups using an even distribution

the entire group, and one where nodes are distributed according to a normal (Gaussian) distribution.

Two key factors in determining how well two groups will communicate are the *node coverage* over the group, and the *overlap* between groups. Our simulation results show the percentage of nodes within a group able to communicate with adjacent groups, given the node coverage and overlap size of the network. Figure 4 shows the results for an even distribution of nodes over a group, and Figure 5 shows the results for a Gaussian distribution of nodes over a group.

Based on these results, it is clear that a more even distribution of nodes is desirable when implementing a group-based model. It is interesting to note the much higher percentage of nodes able to communicate in even distributions. The lower figure for the Gaussian distribution is due to the lower density of nodes distributed over the edge of the Gaussian group. Based on these results, we can make decisions regarding network architecture, taking into account the distribution method, and how nodes are likely to be distributed in their final state.

5.2 Sensor Impact

A critical analysis of any sensor network application must include the impact to the sensors in terms of space requirements, processing time, and power consumed. We believe the solution presented here has an acceptable footprint upon the sensor nodes. For analysis purposes, we will consider an implementation where p is a 512-bit prime and q is a 160-bit prime.

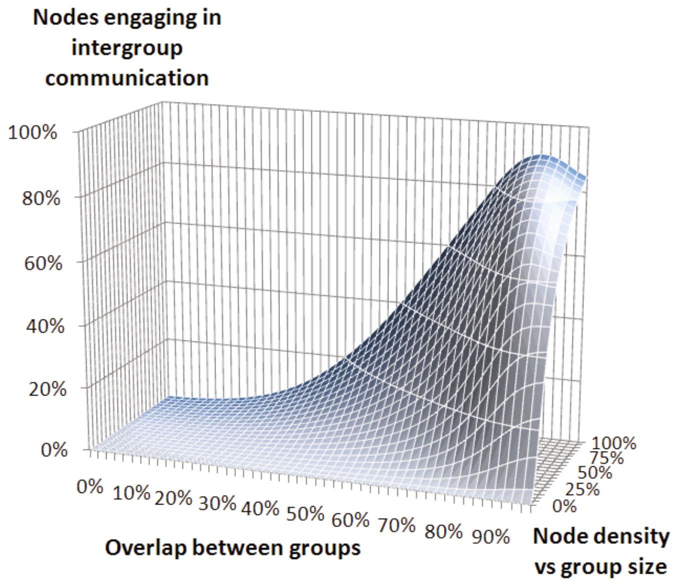

Fig. 5. Nodes able to communicate with other groups using a normal (Gaussian) distribution

Sensor memory requirements. The use of elliptic curve cryptography (ECC) based mechanisms means that less space is required to store key information. To achieve the same public key security as a 1024-bit RSA [21] implementation, only 160-bit keys are needed for ECC [6]. Standard ECC-based digital certificates, used in traditional key establishment, are ∼530 bytes in size. This can be reduced to 86 bytes by removing all but the most important data [8]. Our proposal reduces this size even further, requiring only 40 bytes to store node identification and keying information. This improvement becomes even more important when we consider how much energy is consumed during data transmission - sending a singe bit of information can take the same power as 2090 clock cycles of execution [8].

However, in our solution, more than just one key is stored on each node. Adding keys for adjacent group key establishment depends on the topology of the network. Without loss of generality, we assume a topology similar to Figure 1. In this instance, the total number of keys to be stored on each node is $\sum_{i=1}^{n} 2^{i+2}$. To store 1 layer of key establishment information requires 160 bytes. For 2 layers, 480 bytes are needed. 3 layers require 1120 bytes. This must be considered in network design.

Group public information, $(p, q, \mathbb{E}/\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H, h, ID_A)$, must also be loaded onto each node. Due to space requirements, we will not detail each component, but will note that p is a 512 bit prime, $\mathbb{G}_1, \mathbb{G}_2$ are 160 bit primes, ID_A can be relatively small (16 bits), and the components relating to key generation code can be estimated at ∼3.6KB [6] for 160-bit ECC operations.

We note that the Atmel ATmega128 [22], a common micro-controller used on many sensor devices, has 128KB of FLASH program memory. Another common sensor network node, the Crossbow Imote2 has 256KB of SRAM memory, 32MB of SDRAM memory, and 32MB of FLASH memory [23]. We believe sensor deployments where > 3 layers of shared key information are unlikely, so a total overhead of < 5KB seems reasonable for this application.

Processing time. Several works have analyzed processing time for ECC-based operation on various sensor devices. While some report times in excess of 30 seconds for generating shared secret keys [7], improvements in hardware, as well as algorithm design, have made significant improvements. In [6], an average time of 0.81s for 160-bit ECC operations is given.

Other papers have shown efficient implementations of pairing schemes. In [11], it is shown that a commonly used processor, the Intel PXA255, can generate a Tate pairing in approximately 62 ms. However, this is a 32-bit processor typically used on network gateway devices, such as the Crossbow Stargate, not on actual sensor nodes. Typical 8-bit sensor nodes take > 30 seconds on average to compute Tate pairings [24].

Recent advances in sensor nodes have included the addition of more powerful 32-bit processors at the node level. In [13], it is shown that it takes 290 ms to compute a Tate pairing on a 32-bit MIPS-32 based SmartMIPS architecture for smartcards, running at 36MHz. The Crossbow Imote2, uses a 32-bit Intel PXA271 scalable processor (13MHz - 416 MHz), and would require $\sim$23.2 ms for the same operation at its highest speed, 92 ms at 104MHz (normal operating speed), and 736 ms at 13MHz (low-power operating speed).

Processing power. We show the power consumption for various components of our solution, using an Imote2 sensor node running at the most conservative power level. The power consumption figures, including both computational operations and radio transmission, for establishing a pairwise key between two sensor nodes are shown in Table 1.

Transmission power. Because our solution uses such a small footprint for identity and group identification information, the transmission power necessary to establish a pairing is small. The total power required to establish a pairing,

Table 1. Power consumption of pairwise key establishment operations on an Imote2 Sensor node

Operation	Energy
All transmissions for pairing	203 μJ
Compute pairing (low power)	27.53 mJ
Compute pairing (normal power)	41.29 mJ

using a CC2420 IEEE 802.15.4 radio transciever [25], is approximately 203 μJ. This figure represents all transmission and reception power for a single node involved in the pairing operation, regardless of whether the node initiates the transaction or not.

6 Conclusion

We have demonstrated a method for key establishment in WSNs which leverages the properties of node authentication and group-based keys to strengthen the security of a network. Our solution is a novel approach to key establishment in WSNs that uses a group-based technique combined with identity based cryptography to provide node authentication and intergroup communication. We showed how our solution is resistant to some of the more difficult attacks to prevent in WSNs, including the node replication, Sybil, and wormhole attacks.

Future work will include multi-hop key establishment, which would further enhance the capabilities of a network of this type. Another interesting area of study would be to leverage the capabilities provided by this solution to establish a fine-grained security policy for WSN. Using knowledge such as authenticated node identity and group membership, this security policy could be fine-tuned to provide differing levels of access control to different sensors based on group membership.

Sensor networks will continue to be part of everyday life. As sensors become smaller and more capable, their potential uses will only increase. In many cases, the security of the networks they form is critical to maintain, and we believe this will continue to be an important area of research.

Acknowledgements

This work was partially supported at the Secure Computing Laboratory at New Mexico Tech by the grant from the National Science Foundation (NSF-CNS-0709437).

References

1. Du, W., Deng, J., Han, Y., Chen, S., Varshney, P.: A key management scheme for wireless sensor networks using deployment knowledge. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 1, p. 597 (2004)
2. Huang, D., Mehta, M., Medhi, D., Harn, L.: Location-aware key management scheme for wireless sensor networks. In: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, pp. 29–42. ACM, Washington (2004)
3. Liu, D., Ning, P.: Location-based pairwise key establishments for static sensor networks. In: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, pp. 72–82. ACM, Fairfax (2003)

4. Yu, Z., Guan, Y.: A key management scheme using deployment knowledge for wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems 19(10), 1411–1425 (2008)
5. Gaubatz, G., Kaps, J., Sunar, B.: Public key cryptography in sensor Networks – Revisited. In: The Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks, ESAS (2005)
6. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 119–132. Springer, Heidelberg (2004)
7. Malan, D., Welsh, M., Smith, M.: A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In: 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004, pp. 71–80 (2004)
8. Wander, A.S., Gura, N., Eberle, H., Gupta, V., Shantz, S.C.: Energy analysis of Public-Key cryptography for wireless sensor networks. In: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, pp. 324–328. IEEE Computer Society, Los Alamitos (2005)
9. Watro, R., Kong, D., fen Cuti, S., Gardiner, C., Lynn, C., Kruus, P.: TinyPK: securing sensor networks with public key technology. In: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, pp. 59–64. ACM, Washington (2004)
10. Zhou, Y., Zhang, Y., Fang, Y.: Access control in wireless sensor networks. Ad Hoc Networks 5(1), 3–13 (2007)
11. Zhang, Y., Liu, W., Lou, W., Fang, Y.: Location-based compromise-tolerant security mechanisms for wireless sensor networks. IEEE Journal on Selected Areas in Communications 24(2), 247–260 (2006)
12. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing, pp. 213–229 (2001)
13. Scott, M., Costigan, N., Abdulwahab, W.: Implementing cryptographic pairings on smartcards. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 134–147. Springer, Heidelberg (2006)
14. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
15. Liu, D., Ning, P., Du, W.: Group-based key predistribution for wireless sensor networks. ACM Trans. Sen. Netw. 4(2), 1–30 (2008)
16. Parno, B., Perrig, A., Gligor, V.: Distributed detection of node replication attacks in sensor networks. In: SP 2005: Proceedings of the 2005 IEEE Symposium on Security and Privacy, pp. 49–63. IEEE Computer Society, Washington (2005)
17. Zhu, S., Setia, S., Jajodia, S.: LEAP: efficient security mechanisms for large-scale distributed sensor networks. In: Proceedings of the 10th ACM conference on Computer and communications security, pp. 62–72. ACM, Washington (2003)
18. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
19. Hu, Y.-C., Perrig, A., Johnson, D.: Packet leashes: a defense against wormhole attacks in wireless networks. In: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2003, vol. 3, pp. 1976–1986. IEEE, Los Alamitos (2003)

20. Lazos, L., Poovendran, R., Meadows, C., Syverson, P., Chang, L.: Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach. In: Wireless Communications and Networking Conference, vol. 2, pp. 1193–1199. IEEE, Los Alamitos (2005)
21. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)
22. Amtel Corporation, Amtel ATmega128 (2009), http://www.atmel.com/dyn/products/product_card.asp?part_id=2018
23. Crossbow Technology, iMote2 (2009), http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf
24. Oliveira, L., Aranha, D., Morais, E., Daguano, F., Lopez, J., Dahab, R.: Tinytate: Computing the tate pairing in resource-constrained sensor nodes. In: Sixth IEEE International Symposium on Network Computing and Applications, NCA 2007, pp. 318–323 (2007)
25. Texas Instruments, CC2420 (2008), http://focus.ti.com/lit/ds/symlink/cc2420.pdf