

Providing Anonymity in Wireless Sensor Networks

Yi Ouyang^{*†‡}, Zhengyi Le^{*†‡}, Yurong Xu^{*†‡}, Nikos Triandopoulos^{*†}
Sheng Zhang^{*†‡}, James Ford^{*†‡} and Fillia Makedon^{*†‡}

^{*}DEVLAB, Computer Science Department, Dartmouth College, Hanover, NH 03755

[†]Institute for Security Technology Studies, Dartmouth College, Hanover, NH 03755

[‡]Computer Science and Engineering Department, University of Texas at Arlington, Arlington, TX 76019

Email:{ouyang, zyle, yurong, nikos, clap, jford, makedon}@cs.dartmouth.edu

Abstract—Sensor networks are often used to monitor sensitive information from the environment or track sensitive objects’ movements. Anonymity has become an important problem in sensor networks, and has been widely researched in wireless ad hoc and wired networks. The limited capacity and resources of current sensor networks have brought new challenges to anonymity research. In this paper, two efficient methods are proposed based on using a one-way hash chain to dynamically change the identity of sensor nodes in order to provide anonymity, and their anonymity properties are analyzed and compared.

I. INTRODUCTION

Anonymity in sensor networks means preventing a third party other than the message sender and the base station knowing the identity of the two primary parties in a communication. It includes sender anonymity, receiver anonymity, and unlinkability between the sender and receiver. Thus, an adversary cannot determine the sender and receiver’s identities through reading a message intercepted from the network or through reading messages forwarded by a sensor node it has compromised, and the adversary also cannot determine whether two communication segments (i.e., message transmissions between two neighboring nodes) belong to the same communication between a sensor and the base station.

Anonymizing sensor nodes can confuse adversaries about which sensor is the real sender of a message. To protect the real ID of each sensor, pseudonyms can be used for sensor nodes instead of real IDs; however, using fixed pseudonyms cannot prevent leaking identity information of sensor nodes because a long term passive eavesdropper can deduce the topology of the network through traffic analysis. Misra *et al.* [1] proposed two anonymous schemes for clustered wireless sensor networks. They proposed to use a pool of pseudonyms for a sensor node to select randomly from them when it is sending messages, and also proposed a Cryptographic Anonymity Scheme (CAS) in which the pseudonym of a sensor node is generated from keyed hash functions. The two schemes can both provide anonymity under the assumption

This work was supported in part by the National Science Foundation under award number ITR 0312629 and IDM 0308229 and the Institute for Information Infrastructure Protection (I3P) under an award from the Science and Technology Directorate at the Department of Homeland Security. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the U.S. Department of Homeland Security.

that the secret keys shared by sensor nodes and the base station cannot be compromised. However, since many sensor networks are deployed in malicious environments, we need to consider the consequences of key compromise and how to protect the anonymity of sensor nodes even when their secret keys are compromised.

In this paper, we propose two methods based on one-way keyed hash chains: Hashing based ID Randomization (HIR) and Reverse Hashing ID Randomization (RHIR). They can provide more anonymity to a sensor node even when its shared secret keys are compromised.

The rest of this paper is organized as follows. Section II surveys related work. Section III describes models used in this paper. Section IV and Section V describe the hashing based ID randomization method and the reverse hashing based ID randomization method. Section VI analyzes and compares the anonymity properties of both methods and also compares them with existing methods, and Section VII concludes this paper.

II. RELATED WORK

Researchers are increasingly raising concerns about anonymity in sensor networks, and also in related areas such as wireless ad hoc networks, data mining, and location-based services. Chaum’s mixing approach [2] was shown to provide anonymous connections that protect against traffic analysis and were useful in an onion routing mechanism [3]. Kong *et al.* [4] proposed ANODR, an anonymous on demand routing for ad hoc networks. Wu *et al.* [5] proposed AO2P, which is an on-demand position-based private routing protocol. Kong *et al.* [6] use threshold secret sharing, secret share updates and a certificate-based approach based on PKI to address security issues in wireless networks. Wadaa *et al.* [7] proposed an energy-efficient protocol for maintaining the anonymity of the network virtual infrastructure that includes a coordinate system, a cluster structure, and a routing structure.

Many key management schemes have been proposed for sensor networks such as [8] and [9]. Zhu *et al.* [10] presented LEAP, a key management protocol for sensor networks. It divides keys into four categories: individual keys, pairwise keys, cluster keys, and group keys. Using location-based keys, Zhang *et al.* [11] proposed a node-to-node authentication scheme that can also facilitate the establishment of pairwise keys between neighboring nodes. There also has been some research on traffic analysis attack in sensor networks such as

[12], [13], and [14]. Kamat *et al.* [13] extended the work of [12] and proposed phantom routing techniques based on both flooding and single-path routing.

III. SYSTEM MODELS

In this sections, we will describe the system models used in this paper including a network model, a key management method, and an adversary model.

A. Network Model

Let $S = \{s_1, s_2, \dots, s_n\}$ denote the set of sensor nodes in the network of size n . The operations of a sensor network can be divided into two phases: a deployment phase and a regular use phase. In the deployment phase, every sensor node determines its own location using some localization method and notify the base station its location. In the regular use phase, a sensor node collects data from the environment and sends the collected data along with its ID back to the base station.

B. Key Management

There has been much research in key management in sensor networks. In our model, we adopt the method proposed in [10] to generate individual node keys and pairwise shared keys amongst neighboring nodes. An individual node key is the key shared by a sensor node and the base station. Pairwise shared keys are shared by a sensor node and its immediate neighbors.

C. Adversary Model

Based on their capabilities, adversaries can be divided into two categories: passive and active. Passive adversaries eavesdrop on communications between sensor nodes in the network. Active adversaries can compromise or physically capture sensor nodes to obtain their data and encryption keys. In this paper, we assume that adversaries have both of the above capabilities: an adversary can eavesdrop on all the communications in the network and capture a limited number of sensor nodes in the network.

IV. HASHING-BASED ID RANDOMIZATION

In this section, a hashing-based ID randomization (HIR) method is proposed to protect the anonymity of sensor nodes in a sensor network. The basic idea is for each individual sensor node to use a one-way keyed hash chain for producing a sequence of hash values as its IDs. Assuming the hash function is shared and thus known to all the sensor nodes, if an adversary compromised one sensor node, it is clear that he could also compute the next ID for every node using the hash function. Motivated by this concern, we wish to identify a keyed hash function that can evolve IDs such that an adversary cannot link two values as the input and output of a keyed hash function without knowing the key. Without knowing the secret keys, the adversary cannot determine if two messages are sent by the same sensor node through eavesdropping. The anonymity of a message's sender is protected as long as the key is not compromised. A sensor node can delete its previous ID and generate a new one after sending a message. Then through physically capturing a sensor node, the adversary only

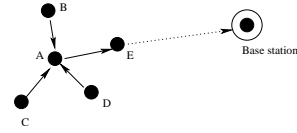


Fig. 1. Example of node A creating table R

TABLE I
AN EXAMPLE OF TABLE R

Hash times (HT)	Hash values (HV)	Link
1	$H_{K_{AB}}^{K_{AB}}(ID_A)$	down
2	$H_{K_{AC}}^2(ID_A)$	down
4	$H_{K_{AD}}^4(ID_A)$	down
2	$H_{K_{AE}}^2(ID_E)$	up

knows the current ID and a secret key. Because of the one way hashing, the adversary cannot reverse the hash function to get the previous IDs used. Thus, the anonymity of a sensor node in the past is protected even when the key is compromised.

A. Deployment Phase

In this phase, a sensor node localizes itself and reports its location back to the base station. It sends a message $M = (ID_i, E_{k_i}(x_i, y_i))$ to the base station, where ID_i is assigned by the base station and (x_i, y_i) are its coordinates. The base station adds $(ID_i, (x_i, y_i))$ into the table mapping between IDs and corresponding real locations. Every sensor node also determines its uplink-node and downlink-nodes. By uplink-node B of node A , we mean that node B will forward A 's message to the base station— B is selected so that it is closer to the base station than A and it is on the routing path of A 's messages to the base station. Conversely, node A is node B 's downlink-node.

After determining the routing information and creating pairwise shared keys with its neighbors, a sensor node s_i will create a table R_i that includes the keyed hash values of neighboring nodes' IDs. Table R_i is a set of tuples $(HT, HV = H_{K_{ij}}^{HT}(x), Link)$, where $x = ID_j$ if s_j is s_i 's uplink-node, $x = ID_i$ if s_j is s_i 's downlink-node, $H_K^{HT}(x)$ means using K as a key to hash x for HT times, and $Link$ denotes uplink or downlink. Node s_i first broadcasts a message to its neighbors with its ID ID_i and node s_j sends back an ACK message with its ID ID_j . Nodes s_i and s_j compute their pairwise key as described in Section III-B. Node s_i then decides if node s_j is its uplink node or downlink node and adds $(HT, H_{K_{ij}}^{HT}(x), Link)$ into its table R_i ($x = ID_j$ or ID_i depending on node s_i and s_j 's relation). An example of R is shown in Table I for node A in Figure 1.

B. Regular Use Phase

After the deployment phase, the sensor network is ready for regular use. Messages sent from a sensor node and routed through the sensor network to the base station are of the form $M = H_1 || H_2 || J || D$, where " $||$ " indicates concatenation, H_1 is a hash value that identifies the receiver of this message, H_2

is a hash value that identifies the original message sender, J is an index that indicates which hash value of H_2 is used, and D is a data block collected by the sender. Node s_i has a counter t to count how many messages it has sent to the base station; this is also the number of times the node has hashed its original ID .

The operations of node s_i originating a message are described in Procedure 1. The operations of node s_j forwarding the message are described in Procedure 2.

Procedure 1 Operation of a single node s_i in regular use phase

- 1: $t = 1$
 - 2: **while** true **do**
 - 3: s_i collects data D_t .
 - 4: s_i sends $M = H_{K_{ij}}^{HT_j}(ID_j) || H_{K_i}^t(ID_i) || t || D_t$ to s_j .
 - 5: $HT_j = HT_j + 1$, $HV_j = H_{K_{ij}}(HV_j)$
 - 6: refreshes table R_i for entry s_j
 - 7: $t = t + 1$
 - 8: **end while**
-

Procedure 2 Operation of node s_j forwarding a message from s_i

- 1: receives $M = H_{K_{ij}}^{HT_j}(ID_j) || H_{K_i}^t(ID_i) || t || D_t$ from s_i .
 - 2: checks its table R_j to find a match of $H_{K_{ij}}^{HT_j}(ID_j)$ and HT_j .
 - 3: **if** there is a match **then**
 - 4: confirms itself as the message's receiver and prepares to forward the message
 - 5: $HT_i = HT_i + 1$, $HV_i = H_{K_{ij}}(HV_i)$ (refreshes table R_j for entry s_i)
 - 6: checks its table R_j for an uplink s_k
 - 7: changes M to $M' = H_{K_{jk}}^{HT_k}(ID_k) || H_{K_i}^t(ID_i) || t || D_t$
 - 8: sends M' to uplink-node s_k
 - 9: $HT_k = HT_k + 1$, $HV_k = H_{K_{jk}}(HV_k)$
 - 10: refreshes table R_j for entry s_k
 - 11: **else**
 - 12: discards the message and returns to original state.
 - 13: **end if**
-

Because of the open medium property of sensor networks, when a sensor is sending messages, all the nodes that are in its radio range can receive the message. To protect the receiver's identity, $H_{K_{ij}}^{HT_j}(ID_j)$ is used to notify the real receiver. If node s_j checks its own table R and finds a match in its table and the message is from a downlink node, node s_j knows that this message is sent to itself and needs to be forwarded to the base station; otherwise, it discards the message. To maintain its table R , each node will update its entries after every use. In this way, the sender and the receiver's table R are synchronized.

The base station receives a message from one of its sensor nodes, but doesn't initially know the sender of this message since the ID of the sender is hidden in $H_{K_i}^t(ID_i)$. The number t included in the message allows the base station to interpret $H_{K_i}^t(ID_i)$ correctly. Note that because of the properties of one-way hash functions, the base station cannot simply compute an inverse hash function to get the original ID of the sender directly; however, the base station has the original IDs of all the sensor nodes, and knowing t and

$H_{K_i}^t(ID_i)$, it is not difficult for the base station to find the ID of the sender by calculating the t th hash for each potential s_i .¹

V. REVERSE HASHING ID RANDOMIZATION

In this section, we propose reverse hashing ID randomization (RHIR) method. In this method, we use a one-way hash chain in reverse—in other words, we assign a sensor node's ID backwards from the end of the hash chain to the beginning. This change improves the security properties of our method, as we will explain below.

RHIR uses the same operations as the HIR method in the deployment phase, and the processes for a node to generate a hash chain and for the base station to determine the sender of a message are also the same, as is message forwarding. Here we focus on the differences between RHIR and HIR method in the regular use phase.

A single node using a keyed hash function generates a chain of values: $ID, H_K(ID), H_K^2(ID), \dots, H_K^k(ID)$. In RHIR, a node uses that hash chain in reverse. The node will use $H_K^k(ID)$ as its first ID and $H_K^{k-1}(ID)$ as its second ID, and so on. Its first message will be $M_1 = H_{K_{ij}}^{HT_j}(ID_j) || H_{K_i}^k(ID_i) || k || D_1$, and its second message will be $M_2 = H_{K_{ij}}^{HT_j}(ID_j) || H_{K_i}^{k-1}(ID_i) || k-1 || D_2$. In this method, a sensor node needs to compute the hash chain first and store it locally. Assuming a 128 bit hash function is used and $k = 1000$, then total storage for hash values is 128×1000 bit = 16K bytes. A MICA2 Mote has a flash memory of 512K bytes. Thus, a sensor node's limited memory storage will not be a problem if a proper k is selected in this method. This method costs more storage but offers better security.

VI. ANONYMITY COMPARISON

In this section, we compare the two methods proposed, HIR and RHIR. For the anonymity property, both methods can provide anonymity to message senders and forwarding nodes in the network. Without knowing the secret keys used for hashing, an attacker can only see the IDs in messages as a series of random numbers. Also because the secret key used to evolve a sensor node's IDs is only shared between this sensor node and the base station, another sensor node's compromise will not affect this sensor node's anonymity. In other words, an attacker cannot compromise a sensor node's anonymity without knowing its secret key, no matter how many other sensor nodes the attacker has compromised in the network. Thus, the anonymity property is the same in RHIR as in HIR if the key is not compromised. From the above, we know that the secret keys used in hash functions are very important to maintaining the anonymity of sensor nodes. What will happen when the secret keys are compromised? Assume a sensor node A 's secret key and the current ID ID_c of A are compromised. We discuss this in two categories, the messages sent before the compromise and the messages sent after the compromise.

¹We assume that for a reasonable hash function, the possibility of a collision is negligible. If a collision does occur, i.e. if there are two (or more) nodes with identical $H_{K_i}^t(ID_i)$ values, the base station can transmit a message to all these nodes requesting retransmission of any data from the t th message.

A. Anonymity w.r.t. Past Messages

In HIR, because the hash chain is used in its normal order, the attacker can compute the next ID $ID_{c+1} = H_K(ID_c)$ based on the ID_c and the compromised secret key K . Assume the attacker records the transmission in the network for a period of time before A 's key compromise. Then the attacker can compute on all the IDs included in previous messages and match them with ID_c . If there is a match, then that previous ID of this node will be compromised. However, the computation cost is very high for the attacker. Because there is no order of messages' arrivals, the attacker may need to do more than n hash function evaluations to retrieve the previous ID of ID_c . Thus, if an attacker wants to trace a sensor node's history readings, its computational cost is $\Omega(n \cdot h \cdot t)$, where h is the cost for a hash function evaluation and t is the number of time points traced back. In RHIR, because the hash chain is used in reverse, an attacker can use the compromised key K and the current ID_c to compute the ID_{c-1} used in the previous message. The attacker only need to search in the existing records for the message that includes this ID_{c-1} . Thus, the computational cost of the attacker tracing back is $\Omega((h+n) \cdot t)$.

B. Anonymity w.r.t. Future Messages

In HIR, when a new message with ID_{c+1} is observed, the attacker can determine that this new message is sent by A because he can compute it based on ID_c and the compromised key. Thus, if the attacker wants to compromise all the messages sent by A after the key compromise, the attacker only needs t hash function evaluations, where t is the number of messages the attacker wants to compromise. Also, because the attacker can compute the next ID itself, the attacker can impersonate the compromised sensor node to send future messages before the real sensor node does. In RHIR, the attacker cannot compute the next ID from the current ID_c and the compromised key K because the hash chain is used in reverse. Thus, the only way for the attacker to compromise the next ID of ID_c is to evaluate the hash function on every ID included in the messages after ID_c . If one message has an ID such that $H_K(ID) = ID_c$, this message is sent by A . Thus, if the attacker wants to compromise all the messages sent by A after the key compromise, the attacker needs nt hash function evaluations. In RHIR, an attacker cannot impersonate a sensor node to send messages to the base station since the attacker cannot compute the next ID even if the secret key is already compromised.

Our scheme resembles the CAS scheme (Section I) in some ways. In that scheme, the one way hash function is operated on a shared fixed random number and a sequence number between the sender and the receiver. The hash values are used as IDs for receivers, and change through incrementing the sequence number. Thus, if a node's key and current ID is compromised, all its past and future IDs are very easy for the adversary to compute, unlike in our two schemes, where the past IDs or future IDs are harder to compute even when the secret key and current ID are compromised. Thus, our methods can provide better anonymity.

VII. CONCLUSIONS

In this paper, we proposed two schemes for protecting anonymity of sensors in sensor networks, HIR and RHIR. Both schemes can provide better anonymity than previous solutions when the secret keys are compromised. HIR is suitable for applications concerned about anonymity of sensor nodes' past behaviors and RHIR is more suitable for applications that need to continue operating when key compromise happens. HIR is more preferable for long run applications since RHIR needs more storage in sensor nodes to store hash values. The more storage sensor nodes have, the longer RHIR can operate. As future work, we will investigate how to combine these two schemes and provide both past and future anonymity for sensor nodes when keys are compromised in sensor networks.

REFERENCES

- [1] S. Misra and G. Xue, "Efficient anonymity schemes for clustered wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 1/2, pp. 50–63, 2006.
- [2] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.
- [3] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 482–494, May 1998.
- [4] J. Kong and X. Hong, "Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks." in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, 2003, pp. 291–302.
- [5] X. Wu, "Ao2p: Ad hoc on-demand position-based private routing protocol." *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 335–348, 2005, fellow-Bharat Bhargava.
- [6] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing Robust and Ubiquitous Security Support for Wireless Mobile Networks," in *Ninth International Conference on Network Protocols (ICNP'01)*, 2001, pp. 251–260.
- [7] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "On providing anonymity in wireless sensor networks." in *10th International Conference on Parallel and Distributed Systems (ICPADS 2004)*, 7-9 July 2004, Newport Beach, CA, USA, 2004, pp. 411–418.
- [8] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, 2002.
- [9] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Security and Privacy*, May 2003, pp. 197–213.
- [10] S. Zhu, S. Setia, and S. Jajodia, "Leap: efficient security mechanisms for large-scale distributed sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2003, pp. 62–72.
- [11] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing sensor networks with location-based keys," in *Wireless Communications and Networking Conference*, vol. 4, 2005, pp. 1909–1914.
- [12] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM Press, 2004, pp. 88–93.
- [13] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 599–608.
- [14] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 113–126.