ARTICLE IN PRESS

Ad Hoc Networks xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Achieving distributed user access control in sensor networks

Haodong Wang [a,*], Qun Li [b]

[a] Department of Computer and Information Science, Cleveland State University, Cleveland, OH 44115, United States
[b] Department of Computer Science, College of William and Mary, Williamsburg, VA 23187-8795, United States

ARTICLE INFO

Article history:
Received 20 April 2009
Received in revised form 4 December 2009
Accepted 25 January 2011
Available online xxxx

Keywords:
Sensor networks
User access control
Public key cryptography
Elliptic Curve Cryptography

ABSTRACT

User access control in sensor networks defines a process of granting user an access right to the stored information. It is essential for future real sensor network deployment in which sensors may provide users with different services in terms of data and resource accesses. A centralized access control mechanism requires the base station to be involved whenever a user requests to get authenticated and access the information stored in the sensor node, which is inefficient, not scalable, and is exposed to many potential attacks along long communication paths. In this paper, we propose a distributed user access control under a realistic adversary model in which sensors can be compromised and user may collude. We split the access control into local authentication conducted by a group of sensors physically close to a user, and a light remote authentication based on the endorsement of the local sensors. We implement the access control protocols on a testbed of TelosB motes. Our analysis and experimental results show that our schemes are feasible for real access control requirements.

1. Introduction

Access control defines a process of identifying users and granting them the access right to information or resources. A sensor network is a computing platform for users to collect data, transmit data, and process data. The access control pertaining to sensor network predominantly aims to protect the network usage and collected data. Unauthorized users should not be allowed to use the network since network bandwidth is very limited and, more importantly, the battery power of each node may be depleted after malicious users aggressively send messages to the network. The data collected or processed, many times, are classified so that data of different classifications require different authorizations for legitimated accesses. For example, a high-rank officer may need to know more information about the field deployment than a soldier. In another scenario, the information with the same classification may be physically or logically compartmented according to its other properties, such as ownership, so that the access to the data has to be verified for the corresponding properties. An example would be a user is authorized to access the data from the sensors in his own office, but not other people's offices.

To achieve access control, it is essential for sensor nodes to authenticate the identities of the requester. This paper aims to explore an efficient and secure authentication scheme for sensor nodes. A natural way for the authentication check is to use a centralized mechanism. After receiving a request, the sensor node sends the user information to the base station. Then the base station decides whether the access is granted or not and replies to the sensor node. This solution may yield a good security result because of the fact that a base station is considered secure, and communication channels between sensors and the base station are assumed secure. However, this scheme suffers two major problems. First, the centralized authentication requires at least one round-trip communication between a sensor and a base station. If a number of users are accessing the network at the same time, the authentication traffic may

* Corresponding author. Tel.: +1 804 338 5573.
E-mail addresses: hwang@cis.csuohio.edu (H. Wang), liqun@cs.wm.edu (Q. Li).

1570-8705/$ - see front matter © 2011 Published by Elsevier B.V.
doi:10.1016/j.adhoc.2011.01.011

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

easily cause network congestion. Second, this authentication pattern is vulnerable to adversary DoS attacks. Sensor nodes have no knowledge about user access right until they get replies from the base station. An adversary can easily launch DoS attacks by forging a large number of user access requests, which will in-turn trigger the same amount of authentication requests. The resulting authentication traffic will saturate the network and quickly deplete the sensor power.

This paper gives a thorough exploration for sensor network data access control problems in a general setting. We consider a data access scenario that a user can access in-network stored data at any location from anywhere in a network, which includes local data accesses from users' nearby sensors and remote data accesses. Moreover, we consider the access control problem in a much harsher environment in which users may collude and sensors may be compromised. Compromised sensors can get the information from user authentication processes and may disclose this information to an adversary, which may potentially help the adversary to gain more access privileges. Colluding users may analyze their information and design a scheme to counteract the access control system. Besides, we also addresses node duplication attacks and DoS attacks by inundating authentication messages to the network.

It is our belief that our general data access model and realistic adversary threat model define a very realistic problem for future sensor deployment. Our work has following four contributions. First, we propose a practical and scalable certificate-based local authentication. Public key cryptography (PKC) eliminates the complicated key management and pre-distribution required by symmetric key schemes, and provides a very clean interface between users and sensors. The advantage of certificate-based authentication is that sensors do not need the storage for users' public keys or a third party for public key verification. Users' public keys can be constructed from their certificates and published system information. Second, we propose a novel group endorsement scheme to authenticate a user locally by a group of sensors and transfer the endorsement to a remote sensor. This scheme is resilient to the compromise of a limited number of sensors and the DoS attack launched in a form of remote access requests. Third, our scheme eliminates the possibility of user collusion attacks. The polynomial based secret sharing scheme proposed in [23] suffers user collusion attacks. The collusion by a number of users can easily reconstruct the secret polynomial and reveal the system secrecy. Our certificate-based authentication is resilient to any user collusion attack. Fourth, we show our schemes are feasible in a real sensor network deployment. We have implemented both local authentication and remote authentications on TelosB motes, which are based on our implementation of 160-bit ECC security primitives.

2. Related work

Sensor network security has received the great attention. Message authentication is an important security component, and can also be a part of user access control context. Perrig et al. [12] constructed μTesla and introduced the asymmetric mechanism through a delayed symmetric keys disclosure: the base station broadcasts an encrypted message first, and then releases the secret key in scheduled time frame. Wang et al. proposed a public-key based approach and allow sensors to authenticate the broadcast messages in a distributed way. The similar schemes are also proposed in [20,25,22]. The distinction of user access control is that a secure channel between a user and accessed sensors has to be established. Message authentication schemes, however, generally do not have such requirement.

To establish a secret communication channel, a user and the accessed sensor need to share a common secret, such as a pairwise key. Eschenauer and Gligor proposed a random graph based key pre-distribution scheme [6]. The scheme assigns each sensor a random subset of keys from a large key pool, and allows any two nodes to find one common key and use that key as their shared symmetric key. Based on their contribution, a number of researches [3,5,9,21,16,17] have been delivered to strengthen the security and improve the efficiency. In above schemes, each sensor has to hold a subset of system secret information. This requirement is not appropriate for user access control. The reason is that the users, the components outside of sensor networks, may collude together by sharing their partial secret shares and aggregate the information to subvert the system security.

The most related research to the user access control is [23,10,14,24]. Zhang et al. [23] proposed several schemes to restrict and revoke the access privilege of a mobile sink. Their approaches are based on Blundo's scheme to establish secret key between a mobile sink and sensor nodes, and then use Merkle tree technique to reduce the overhead. The limitation of the scheme is that the moving track of the mobile sink has to be predetermined by the base station. In comparison, our schemes address a more general user/sensor communication problem. The mobile sink can be regarded as one type of special users in our schemes. The restriction of the scheme in [10] is the assumption that requires no security attack during the initial network deployment period. While the public key based access control schemes proposed in [14,24] are resilient to various security attacks, the cost is unfortunately very high. The scheme in [14] needs the support of a centralized server, and the bilinear pairing scheme in [24] is too expensive for resource constrained sensors.

3. System model

We consider a large scale wireless sensor network deployed in a variety of environments, e.g., at a hostile battlefield, in an office building, or in a national park. Data accesses to the stored data on each node are protected according to the attributes of the data, e.g., data type, data location, data collection time, and so on. For a specific data, only authorized users are allowed to access from the storing node. Since data are distributed in the entire network instead of in a central position, data protection by relying

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

on a powerful base station with all data access authorization information and computational power is not practical. Instead, data access authorization should be done in a distributed fashion. After the user access right is verified, the data access is granted and the data content is delivered to the user.

A user equipped with a powerful computing device, such as a PDA, interacts with the sensor network for data query and possibly network control, such as network provisioning or resetting. The PDA is an interface for the user to interact with the sensor network. Since the PDA is more powerful than sensor nodes, it is capable of more computationally intensive tasks. Users can query data at any location in the network through multi-hop sensor node relay. The data access capability, however, must be granted by a central authorization center. A data access list is associated with the user about the types, locations, and the durations of the authorized data access. This information is encrypted in a way that the user is unable to forge and can be efficiently authenticated by sensors that receive the access requests.

The sensor network is managed by a Certification Authority (CA), which is responsible for generating all security primitives (i.e. random numbers, one-way hash function, access list) and revoking users' access privilege if necessary. CA distributes secret keys through the base station. To access a sensor network, users need to apply for the access permission from CA. CA maintains a user access list pool and associated user identifications. The access list defines the user's access privilege. A typical access list is composed of *uid* and *user access privilege mask*. *uid* is a unique number that identifies a user. *user access privilege mask* is a number of binary bits; each bit represents a specific information or service. An access list example is shown in Fig. 1. CA issues a proper access list to each applicant. The information stored at each sensor node is divided into multiple access privilege levels. The user with a lower access privilege is not allowed to access the information that requires a higher privilege. We assume users can securely acquire their access lists from CA through out-of-band secure communication channels. Once a user passes the authentication check, the accessed sensor nodes provide their stored information to the user. If the required information is not available locally, for the reason we will discuss later, a group of sensor nodes have to collaborate and request the information from a remote sensor that holds the information.

In this paper, the adversary is assumed to be able to launch various attacks to access the data that is not authorized to him. We not only consider the common attacking schemes, including message eavesdropping and message reply, we also focus on more hazard attacks, such as sensor node compromise and user collusion. We assume the adversary can capture all information stored after capturing a sensor. It is even worse that the adversary may inject his own program to the compromised sensors, which, under the control of the adversary, pretend to be trustworthy gaining as much information as possible. A user may also collude with the adversary for mutual benefits by attacking the access control system. The base station and CA, however, cannot be compromised.

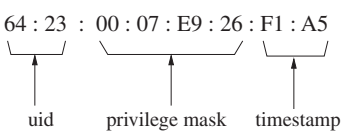


Fig. 1. A typical user access list.

In particular, we consider the following two potential attacks. First, compromised sensors may capture user information and give to an unauthorized user so that the adversary may access data by impersonating another user. Second, user collusion may help malicious users to subvert the system and gain more access right than that of anyone among the colluding group. We assume that at most t sensors can be compromised. The assumption is reasonable because compromising sensors takes time and efforts. On the other hand, we assume unbounded number of users can collude since it is not hard for mischievous users to share information and orchestrate an aggregated analysis to the collected information. The fact that a compromised sensor is hard to identify prevents a user from trusting any of the sensors. A user may have to disclose information for authentication, but the revealed information has to be specific to the sensor in contact and should not be used for authentication at another sensor.

We do not explicitly address the introduction of duplicated compromised sensors. However, since the duplicated compromised sensors do not reveal more information to the adversary, our carefully designed protocols do not allow the adversary to access the data from an uncompromised sensor.

4. Proposed access control schemes

A user may request data stored locally or in a distant sensor. We first define following two types of sensor nodes. The sensor nodes which are directly within the communication range of a user are called *local* sensors. The sensors which cannot establish direct communication links with a user but hold the requested data are called *remote* sensors.

In this section, we first propose a PKC-based local access control scheme. Then we develop a remote access control approach (we assume that the ID of the remote sensor for data access is known by some scheme that is beyond the scope of this paper, e.g., resource discovery or geographic location-based routing). We provide the security analysis for the both schemes. Finally, we give a discussion regarding resource discovery and user certificate issuing.

4.1. PKC-based local authentication

PKC has been used extensively in data encryption, digital signature, and user authentication. Compared with many symmetric-key schemes proposed sensor networks, PKC provides a more flexible and simple interface requiring no complicated key pre-distribution and management as normally required in symmetric-key schemes. It is a popular belief, however, in sensor network research community that public-key cryptography is not practical

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

because the required computational intensity is not suitable for resource constrained sensor nodes. The recent progress in Elliptic Curve Cryptography (ECC) implementation on small devices, however, proves the public key is viable on resource constrained sensors. It is reported [7] that the ECC point multiplication only takes less than one second on Atmel ATmega128 processor, an 8-bit and 8 MHz CPU.

We present our ECC based local authentication scheme as follows. Certification authority (CA) selects a particular elliptic curve over a finite field $GF(p)$ (where p is a large prime), and publishes base point P with order q (where q is also a large prime). CA picks a random number $x \in GF(q)$ as the system private key, and publishes the corresponding public key $Q = x \times P$. Given point P and Q, it is computationally infeasible to get system secret x.

A straightforward user authentication scheme can be described as follows. A user uses her private key to sign her access list and sends to a local sensor. The sensor verifies the signature by using user's public key. However, it is difficult for the sensor to find an trusted third party to verify that the user is who she claims to be. To solve this problem, we adopt the certificate-based authentication in our local authentication scheme. To access the sensor network, a user has to present her certificate first. Based on the certificate, the sensor generates the user public key, and then uses it to encrypt a random number as the challenge. A successful response proves that the user is legitimate.

To access the data stored in the sensor network, the user comes to CA to apply for an access permission. CA picks a random number $c_A \in GF(p)$, and then calculates the user's public key constructor $C_A = c_A \times P$. Based on the user's request, CA issues a proper access control list ac_A, and attaches it to public key constructor C_A as the certificate. Meanwhile, a digest e_A is generated for the access list, where $e_A = H(T_A)$ (H is a $\{0,1\}^* \to \{0,1\}^q$ hash function), where $T_A = C_A \| ac_A$ ($\|$ is the concatenation). Then, CA constructs the user's private key $q_A = e_A c_A + x$ and public key $Q_A = e_A \times C_A + Q$. Note q_A and Q_A satisfy $Q_A = q_A \times P$. Finally, the user holds q_A, Q_A and T_A. We assume above procedure is conducted at an out-of-band secure channel.

The user authentication protocol is illustrated in Fig. 2. We denote s_l as a local sensor. When a user approaches a sensor node s_l, she sends her access request with access list T_A. Given access list T_A, s_l constructs the public key $Q_A = e_A \times C_A + Q$. To verify that the user indeed holds private key q_A, s_l starts the following challenge procedure. First, s_l selects a random number $r \in GF(p)$ (to be used as the session key with the user), and calculate its digest $H(r)$ over $GF(q)$. Second, s_l generates a temporary public key $Y_r = H(r) \times P$, and computes $Z_r = H(r) \times Q_A$. Third, s_l encrypts the session key by doing an *XOR* operation $r \oplus X(Z_r)$, where $X(Z_r)$ is the X coordinate of point Z_r. Finally, s_l sends ciphertext $\langle z_r, Y_r \rangle$ to the user, attached with the encryption of a nonce N_A. We denote $ENC(k,M)$ as an encryption on M by using the secret key k. Similarly, we denote $DEC(k,C)$ as a decryption on ciphertext C by the secret key k.

$$
\begin{aligned}
&(1) && user \to s_l : T_A = (C_A \| ac_A)\\
&(2) && s_l \text{ computes } : Q_A = e_A \times C_A + Q, \text{picks a random } r\\
&(3) && : Z_r = H(r) \times Q_A, Y_r = H(r) \times P\\
&(4) && : z_r = r \oplus X(Z_r), ENC(r, N_A)\\
&(5) && s_l \to usesr : z_r, Y_r, ENC(r, N_A)\\
&(6) && user \text{ computes } : Z_r = q_A \times Y_r, X(Z_r) \oplus z_r = r, DEC(r, N_A)\\
&(7) && user \to s_l : ENC(r, N_A \| ac_A)\\
&(8) && s_l \to user : DEC(r, reply)
\end{aligned}
$$

Fig. 2. User access list authentication protocol. We let s_l be the local sensor, T_A be the user certificate, which includes a public-key constructor C_A and an access list ac_A.

With her private key q_A, the user can regenerate Z_r because $q_A \times Y_r = q_A \times H(r) \times P = H(r) \times Q_A = Z_r$. She then decrypts session key $r = z_r \oplus X(Z_r)$, and verifies if Y_r equals to $H(r) \times P$. If yes, She uses r as the session key to encrypt the nonce N_A concatenated with her access privilege ac_A, and sends to s_l.

Local sensor s_l decrypts the ciphertext and verifies N_A and ac_A. A successful verification proves that the user is the legitimate owner of access list T_A. Finally, s_l replies the information requested by the user, which again is encrypted by the session key r.

4.2. Remote access control

In remote access control, the *remote* sensor node cannot directly contact the user due to the limitation of radio transmission range. Therefore, the user query has to travel multiple hops to reach the *remote* sensor. With this communication pattern, the authentication scheme used in local access control cannot be applied to remote access control. In other words, it is improper for the user to directly contact the *remote* sensor. Otherwise, the adversary can easily take the advantage and launch the bogus data injection attack to deplete the sensor network. We thus develop a remote access scheme that uses *local* sensors to endorse the user query to the *remote* sensor. It is widely accepted [11,12] that a single sensor node cannot be trusted. In our scheme, the user's remote access request has to be endorsed by k local sensor nodes, where k is a system parameter. We assume the adversary cannot compromise k sensors at a time. Any user remote access query without k local endorsements will be dropped immediately by either forwarding sensor nodes or the *remote* sensor. A caveat is that some sensors may be compromised if a valid user cannot be authenticated by a group of sensors. In that case, the user can move to find another group of sensors for authentication or report the failure to the base station for analysis.

The above *local* sensor endorsement raises a new security challenge: how does the remote sensor verify that the user is indeed endorsed by k *local* sensors? If each local endorsing sensor can share a secret with the *remote* sensor, then the endorsement can be easily verified by the *remote* sensor. We use the polynomial-based scheme for secret sharing between the local and remote sensors. More specifically, CA randomly generates a bivariate t-degree polynomial $f(x,y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$ over the finite field $GF(q)$. The polynomial has the symmetric property such that $f(x,y) = f(y,x)$. To endorse a user access list, each local sensor can encrypt the access list with the key shared with the remote sensor, computed by substituting x and y with the sensor IDs. This scheme, however, has to provide the remote

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

sensor with the IDs of the local sensors for verification, which leads to a long message. To reduce the message size, we adopt the following optimization. Before the deployment, sensor nodes are divided into k groups $\{g_1, g_2, \ldots, g_k\}$, where $g_j (1 \leqslant j \leqslant k)$ is a group ID. From now on, we denote a sensor node as $s_i^{g_j}$, where s_i is the sensor id, and g_j is the group ID. During configuration procedure, each sensor $s_i^{g_j}$ is pre-loaded with two shares of the polynomial, $f(x, s_i)$ and $f(x, g_j)$. Given the *remote* sensor ID s_r, a local sensor $s_i^{g_j}$ can establish a pairwise key with the *remote* sensor by plugging s_r in $f(x, g_j)$. Similarly, the remote sensor can also generate the pairwise key by plugging group ID g_j in its $f(x, s_r)$. By using group ID instead of sensor ID, we can achieve a shorter message due to a small number of groups. For the remote sensor to check the authentication list, we attach a bitmap in the message showing which group IDs are used for authentication. We incorporate the remote sensor ID in the polynomial computation rather than the group ID of the remote sensor to avoid the attack due to the scenario that a compromised sensor has the same group ID with the remote sensor and then can decode the shared keys between the local sensors and the remote sensor.

The remote access control protocol is described in Fig. 3. To start a remote access procedure, a user has to find k endorsing sensors $s_i^{g_j}$ such that no two sensors have the same group ID. In the beginning, the user broadcasts a remote access request. The local sensors receiving the request reply with their group IDs. Then, the user select k local sensors with different group IDs to form an endorsing sensor group. Note that the user may have to broadcast the request several times due to the possible transmission collisions. After the endorsing group is formed, each endorsing sensor performs the local authentication as described previously. Once the user is authenticated, each sensor $s_i^{g_j}$ computes the pairwise key $f(s_r, g_j)$ with the remote sensor, and use the key to encrypt the verified user access list. The user collects k encrypted endorsements from these local sensors and generates a hash digest, $mac = H(mac_1 \| \cdots \| mac_k)$, where $g_1 < g_2 < \cdots < g_k$.

After computing the hash digest, the user encrypts her access list T_A and N_A with *mac*. Again, N_A is a nonce to guarantee the message freshness. Then, the user sends it along with her access list T_A and the local endorsing sensor group list, to the *remote* sensor.

(1) user finds k local sensors $s_i^{g_j}$ with different j
(2) for (each sensor $s_i^{g_j}, i, j = 1, 2, \cdots, k$)
  $s_i^{g_j}$ authenticates user access list T_A
  $s_i^{g_j} \rightarrow user : mac_i = ENC(f(s_r, g_j), T_A)$
(3) user computes: $mac = H(mac_1 || \cdots || mac_k)$
(4) $user \rightarrow s_r : ENC(mac, T_A || N_A) || T_A ||$ group list
(5) s_r : compute $f(g_1, s_r), \cdots, f(g_k, s_r)$
(6) s_r : reconstruct $mac = H(mac_1 || \cdots || mac_k)$
(7) $s_r : T_A || N_A = DEC(mac, ENC(mac, T_A || N_A))$
(8) $s_r \rightarrow user : ENC(mac, reply || N_A || N_B)$

Fig. 3. The polynomial based remote access control protocol.

Upon the receipt of the access request from the user, s_r retrieves the information in the group list. Given the group list, s_r easily generates k encrypted endorsements by plugging in the group IDs to its secret polynomial share, and correspondingly, the digest *mac*. The successful decryption of T_A by using the derived *mac* proves that the user has already been authenticated and endorsed by k local sensors. Sensor s_r replies the user with the requested information, along with a nonce N_B randomly picked by s_r. Again, all data is encrypted by *mac*.

4.3. Security analysis

In two proposed access control schemes, the authentication messages are encrypted by the encryption algorithm *ENC* in the access control protocol, except the user certificate. As long as *ENC* is secure (such as RC5 [15]), and the secret key is large enough (at least 80 bits), any number of compromised sensors cannot break the ciphertext in the messages.

In the local authentication, the sensor nodes cannot capture any secret from the user, nor can the user gain more access privilege than granted due to the hardness of discrete logarithm problem in ECC. The 160-bit elliptic-curve crypto-system is considered to have the same security level as 1024-bit RSA. Given an elliptic curve E over finite field $GF(p)$, to find system secret x from the relation $Q = xP$ (where P, Q are published system parameters) is equivalent to solve the discrete logarithm problem, which is considered computationally infeasible. In the local authentication, the user's certificate T_A (with the access list ca_A) is transmitted in plaintext. The malicious sensors may duplicate the user certificate, or the adversary may capture the certificate by eavesdropping. The certificate information, however, cannot help the adversary to impersonate the user and get the data service. The reason is that the local sensors use the derived user public key to launch the challenge. It is easy for the adversary to calculate the public key given the captured certificate, but it is computationally infeasible to acquire the private key that is associated to the public key. As the result, the adversary is not able to correctly respond the challenge, so the access request will be rejected by local sensors. Due to the same reason, the user cannot forge or alter her access list to acquire more access privileges or to extend the allowed access time period. Otherwise, the user will not be able to decrypt the challenge message from the local sensor because she does not have the private key associated to the certificate she claims. More importantly, the certificate-based local authentication effectively defends against user collusion attacks. The collusion among any number of users does not jeopardize the system secret for the reason explained above.

The security features of our remote access scheme lie on the local sensor group endorsement. The combination of our local endorsement scheme with existing false report filtering schemes, including the symmetric-key based SEF [20] scheme or the public-key based PDF [18] scheme, can effectively prevent the potential DoS attacks. To integrate the SEF scheme, each of the local endorsing sensors generates an event report and forward to the user. The user

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

collects k reports and attach them to the remote access request message. Actually, each report is the encryption of the user's access list T_A, which will be verified the the forwarding sensors on the routing path to the remote sensor. In the original SEF scheme, the report encryption keys are randomly pre-distributed to each sensor node. In our scheme, the complicated key pre-distribution can be avoided because the encryption keys can be easily generated from the secret polynomial share in each sensor node. In particular, considering a sensor $s_i^{g_j}$ (which has a sensor ID s_i and a group ID g_j), the encryption key is $f(g_j, h(T_A))$. When the message is on the way to the remote sensor, any forwarding node with the same group ID can verify whether or not the report is legitimate. Any report that fails the verification is immediately dropped. The robustness of this filtering scheme relies on the fact that the different groups should be evenly distributed cross the sensor field. Given the assumption that the number of compromised sensor is limited, the forged report by compromised sensors can be effectively detected and dropped. The obvious disadvantage of the symmetric-key based scheme is the overhead. Each user remote access request has to be attached with k reports. An alternative scheme to reduce the message overhead is to use the public-key based PDF scheme. In that case, the k endorsing sensors jointly generate a system digital signature. As the result, each forwarding sensor can easily verify the remote request by using the system public key. Compared to the symmetric key scheme, the public-key solution only costs a fixed message overhead (the length of a system signature) no matter how large k is. The tradeoff, obviously, is the computation overhead in the signature generation and the verification.

In our scheme, users are not allowed to send requests directly to the remote sensor. Any remote access request has to be enforced by k local sensors. Since the adversary cannot compromise up to k sensors (the system assumption), there is no way for an illegitimate user to get k endorsements to access the remote sensor. If the adversary attempts to forge k endorsements, the bogus request will be immediately dropped by forwarding sensors in false report filtering. Again, the user still cannot alter or forge her access list in the remote access request. The reason is that the endorsements are generated and encrypted using the authenticated user access list. If the user forges her access list in the remote access request, the verification at the remote sensor will fail, and the remote access will be rejected.

4.4. Other design issues

4.4.1. Resource discovery

Careful readers may notice that the proposed access control scheme requires the user to identify the sensors that hold the requested information. The user can acquire such information in the following two ways. First, since the base station knows the approximate locations where the sensors are deployed, a coarse-grained data sensing map can be generated to help the user to locate the interested sensors (e.g., by using GPS localization). After arriving at the desired location, the user can listen to the sensor broadcasts (the sensors can periodically broadcast the attributes of their collected data) and find the local sensors that hold the user interested data. Second, in case the user cannot find the interested information at a certain location, some resource discovery protocols [13] can be applied to identify the ID of the remote sensor that contains the user interested data. The implementation detail of such protocols is beyond the scope of this paper.

4.4.2. User certificate issuing

The user can receive her accessing certificate (including secret keys and the digital certificate) from a central certificate authority (CA) through an off-line transaction (as the way how the driver's licenses are issued by the Department of Motor Vehicles) or an out-of-band security channel (email message delivery). Since the certificate is not delivered through the sensor network, the centralized certificate issuing does not limit the user's in-network data queries. The sensors enforce the access control by verifying the user's certificate and checking the user's corresponding accessing privilege. The public-key cryptography provides a flexible way to specify the access privilege level with a time window by using the access control list discussed previously. It is possible that two sensors hold the user interested data at the same time. Our proposed scheme requires the user to be authenticated twice to get all data even if the two sensors are very close to each other. The design of the more efficient scheme that only needs one authentication in this situation is arranged in our future work.

5. Revocation

It is possible that a user's access list would be revoked due to security reasons. For example, a group of sensors may find a misbehaving user, and those sensors will generate a report and send back to the base station. The base station collects all the reports and makes the decision whether the user's access list should be revoked or not. In this section, we propose two revocation schemes.

5.1. Revocation using blacklist

A simple revocation scheme is to use a blacklist. When the base station decides to revoke a user, it broadcasts the user access list to all the network through the secure channels between the base station and sensor nodes [12]. Each sensor node maintains a table to store the revoked access lists. This screening check can be conducted immediately before the local authentication.

The blacklist revocation scheme is effective when the number of revoked users is small. This simple scheme however will have the scalability issue when the blacklist is inflating. A blacklist with hundreds of revoked users will consume too much precious memory space. Moreover, a large blacklist also costs extra energy and time when user access lists are scanned during the authentication check.

5.2. Revocation using Bloom filters

To solve the scalability problem in blacklist revocation, we propose an efficient revocation scheme by using Bloom

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

filters [1]. Bloom filter is a space-efficient data structure to support membership queries. Given a user blacklist $U = \{u_1, u_2, \ldots, u_n\}$, we allocate a bit vector with the space of m bits (initially cleared to 0), and then choose k independent hash functions $H_1, H_2, \ldots, H_k$, with range $\{0, 1, 2, \ldots, m-1\}$. For each user access list $u_i \in U (1 \leqslant i \leqslant n)$, the bits at position $H_1(u_i), H_2(u_i), \ldots, H_k(u_i)$ are set to 1. An example of a Bloom filter is illustrated in Fig 4.

To check if a user access list T is in the blacklist, we apply $H_1, H_2, \ldots, H_k$ to T. If any one of the results is 0, the access list T is not in the blacklist. If all results are 1, we then consider T is in the blacklist. Note Bloom filter may result in false positive with a certain probability (i.e. a user access list T is not in the black list but all hash functions yield 1). Suppose the hash functions are uniformly random, the probability for a specific bit to be 0 after n member insertions is $\left(1 - \frac{1}{m}\right)^{kn}$. Thus, the probability of the false positive after n member insertions is

$$\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k} \approx (1 - e^{-kn/m})^{k}. \quad (1)$$

The detail revocation scheme (choose $k = 6$, $m = 4096$ as an example) using Bloom filters is described as follows. The six hash functions $H_1, H_2, \ldots, H_6$ are pre-loaded in the sensor nodes during the configuration period. Each sensor node allocates a 512-byte memory space and clears with 0 as the bit vector. Same as in blacklist revocation scheme, sensor nodes report suspicious users to the base station. The base station maintains a revocation blacklist. Whenever there is an insertion in the blacklist, the base station broadcast the results of six hash functions applied on the new item. The results are the positions of the bit vector where need to be set to 1. Fig 5 illustrates the false positive probabilities of a Bloom filter with six hash functions and different bit vector length. Consider an example $m = 4096$ (or 512 bytes), the false positive probability is less than 10^{-4} when more than 200 users are blacklisted. Comparatively, in the blacklist scheme, the sensor has to allocate at least 1.6 KB memory space to store the blacklisted users.

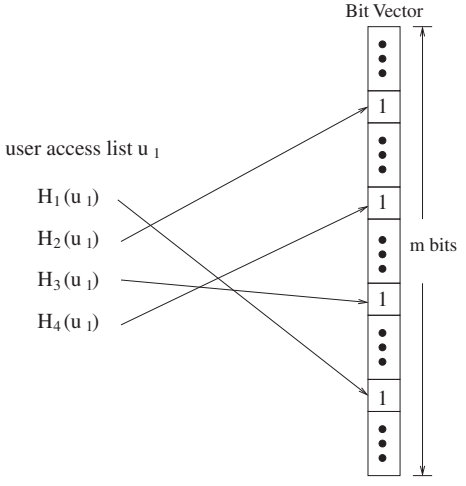


Fig. 4. A Bloom filter with four hash functions.

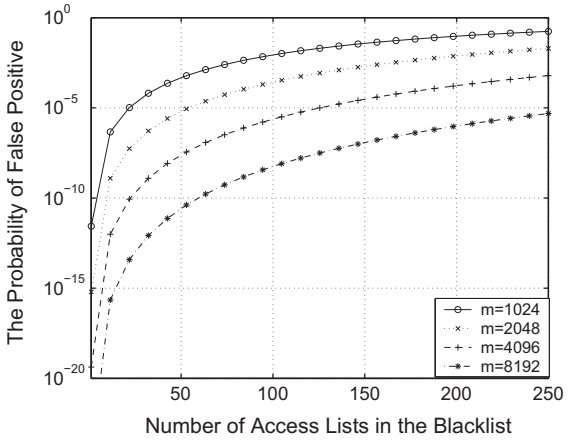


Fig. 5. The probability of false positives (log-scale) as the function of the number of access lists in the blacklist of a Bloom filter with six ($k = 6$) independent hash functions and different bit vector length.

6. Evaluation

In this section, we first study the performance of the proposed scheme by a real world implementation on a commodity sensor platform. Then, we perform the performance comparison between the proposed distributed access control and a centralized scheme. The comparison study is based on the experimental data in the real world experiments.

6.1. Metrics and methodology

In the experiments, we use four metrics: authentication time, computation cost, communication cost, and power consumption, to evaluate the performance of access control protocol. The authentication time measures the user perceived waiting time from sending out the request to receiving the authentication confirmation. Computational cost is the amount of energy consumed in data processing. Similarly, communication cost is the energy used by RF transceiver. The power consumption is the total amount of energy used by all participating sensor nodes to assist the user access request. The two metrics, query response time and power consumption, are used in the comparison study.

The processing energy consumption E can be calculated by $E = U \cdot I \cdot t$, where U is the voltage, I is the current and t is the time duration. TelosB motes are powered by two AA batteries, so U is approximated equal to 3.0 V. The current value varies in different operations as shown in Table 1 (abstracted from [4]). We use authentication time as the time duration for MCU data processing. The energy consumption measuring for communication, however, is

Table 1
The amount of current draw on different operations for TelosB motes.

Operation	Normal (mA)	Max (mA)
MCU On, Radio Off	1.8	2.4
MCU On, Radio Rx	21.8	23
MCU On, Radio Tx	19.5	21

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

more complicated. The data transmitting time is determined by multiple factors, such as wireless channel condition and the corresponding data rate. For simplification, we estimate the communication energy consumption (for either sending or receiving) by multiplying the total amount of data length by an average 18 μJ/bit [2].

6.2. Experiment of local access control

We have implemented both local access control and remote access control scheme on TelosB motes, a research oriented mote developed by UC Berkeley. TelosB is powered by an MSP430 micro-controller. MSP430 incorporates an 8 MHz, 16-bit RISC CPU, 48 KB flash memory (ROM) and 10 KB RAM. The RF transceiver on TelosB is IEEE 802.15.4/ZigBee compliant, and can have up to 250 kbps data rate. We choose SECG recommended 160-bit elliptic curve, secp160r1, in our ECC implementation because large integer multiplication and reduction over prime number finite field can be more effectively optimized than those over binary finite field. The most expensive operation in ECC exponentiation is point multiplication. To achieve the better performance as possible, we have adopted a number of techniques including hybrid multiplication, modular reduction over pseudo-Mersenne prime field, Great Division and mixed Jacobian Coordinate. Due to the space limit, we omit the detail implementation and corresponding optimization of our ECC implementation on TelosB motes. Interested readers may refer to [19] for detail explanation. On average, it takes 1.4 s for a TelosB sensor mote to do a fixed point multiplication, and 1.5 s to do a random point multiplication.

Our local access control implementation strictly follows the protocol presented in Section 4. For the operation of encryption and decryption, we adapt the RC5 block cipher to TelosB platform. Our experiment result shows RC5 is very efficient on TelosB and only produces around 1ms computational overhead.

The challenge generation produces the most time latency in local access control procedure. Recall that a sensor node needs to perform two ECC random point multiplications and one fixed point multiplication to generate a challenge. The three point multiplications combined contribute at least 4.5 s delay. To reduce this delay in challenge generation, we further adopt Shamir's trick [8] to efficiently compute the two random point multiplications so that the improved challenge generation time reduces from 4.5 s to 3.8 s. Accordingly, the energy consumption for computation is 3.8 s × 3.0 V × 1.8 mA = 20.5 mJ.

To estimate the communication energy consumption, we need to count the amount of data sent and received by the sensor node. The user certificate T_A has 48 bytes, including 40-byte public key constructor and 8-byte access list. The challenge from the sensor node has 80 bytes, including a 40 byte ECC point, 20 byte z_r and a 20 byte ciphertext. The challenge response contains 20 byte in size. Overall, the power consumption is $88 bytes \times 18$ μJ/bit = 12.7 mJ.[1]

[1] We ignore the message overhead in the estimation for the simplicity.

Overall, the computing and communication consume similar amount of energy in the location access control. Even though our estimation shows that authentication operation uses a little more power, the difference could be much smaller if the message header in communication power consumption is considered in practice.

6.3. Experiment of remote access control

The essential part of the experiment of remote access control is the polynomial based local endorsement scheme and endorsement decryption at the remote sensor. We are particularly interested in the performance of the t-degree polynomial computation in sensors. Given a share of the polynomial $f(x) = a_0 + a_1x + \cdots + a_tx^t$ over $GF(q)$, the computation of $f(x)$ requires t modular multiplications and t modular additions, plus the computation of values $x^2,\ldots,x^t$. A typical block cipher (e.g., RC5) suggests q should be at least 64 bits. Therefore, t 64-bit × 64-bit modular multiplications are required to compute the polynomial. On TelosB's 16-bit CPU platform, each 64-bit × 64-bit multiplication costs 16 word multiplications. To reduce the computational cost, we adopt the simplification proposed in [9]. The simplification is based on the fact that variable x is either sensor id or group id, which is normally a 16-bit integer. We can use another finite field $GF(q')$ for $x,x^2,\ldots,x^t$. Therefore, the modular multiplication in polynomial $f(x)$ is always performed between a 64-bit integer and 16-bit integer. As the result, the cost of multiplication is reduced by four times.

The modular reduction operation is as important as modular multiplication. Each multiplication must be followed by a reduction operation. To further reduce the computational cost, we pick a pseudo-Mersenne prime as q because modular reduction cost on field of a pseudo-Mersenne prime can be optimized to a negligible amount. A pseudo-Mersenne prime can be represented as $q = 2^m - \omega$, where $\omega \ll 2^m$. Given a 2 m-bit multiplication result $B = (b_1, b_0)$, (b_1, b_0 are two m-bit halves), the reduction can be computed based on the congruence $2^m \equiv \omega$:

$$\begin{aligned} &\text{while } (b_1 \neq 0) \\ &\quad (b_1, b_0) = b_1 * \omega + b_0 \\ &B = b_0 \bmod q. \end{aligned} \tag{2}$$

In our experiment, we choose $q = 2^{64} - 2^8 - 1$, $q' = 2^{16} - 2^4 - 1$. We test the average time delay and power consumption for computing the polynomial with different t values. In each test, we randomly generate $t + 1$ 64-bit coefficients and a 16-bit variable x, we repeat 20 times to get the average time delay. The test results are shown in Fig. 6.

The test results show the polynomial computation is efficient even in low-power sensor nodes. Considering we require 16 local sensors to endorse user's remote access, and each sensor has to store two shares of the polynomial, the system should at least be deployed with a 32-degree polynomial. Therefore, it only takes an endorsing sensor 17.1 ms time to generate pairwise key with the remote sensor.

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

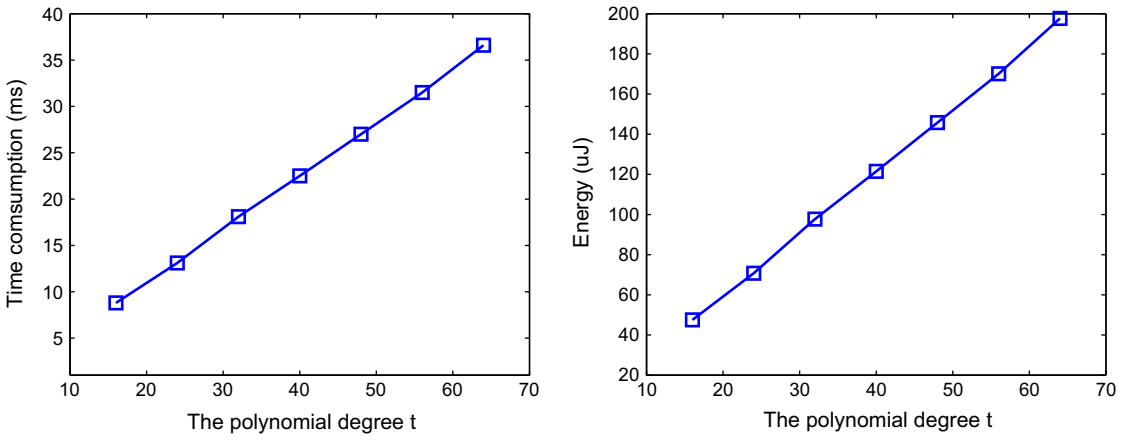


Fig. 6. The time consumption and power consumption to calculate the polynomial.

To evaluate the remote access control procedure, we divide the experiment into two parts. The first part includes local sensor discovery, local sensor authentication and endorsement collecting. In the second part, we perform the endorsement reconstruction and verification at the remote sensor. The message routing between the user and the remote sensor is a typical communication process that has been investigated extensively and the time delay is very small, so in our experiment we omit the message routing between the user and the remote sensor.

During the experiment, we assume the sensor field is dense enough so that the user can reach all the sensors from different groups without moving. To acquire the endorsements from local sensors, the user first broadcasts a remote access request. Each local sensor replies the user with its group id. The user picks those sensors from different groups to fill in her endorsing list. Due to the message collision, some replying messages are corrupted, so the user may not find enough endorsing sensors with one broadcast. The user thus has to broadcast several times to find all k endorsing nodes. Our experiments show the user needs to broadcast at least twice if $k \geqslant 6$. After successfully finding k endorsing sensors, the user unicasts an endorse acknowledge to each of the k sensors. The endorsing sensors process the user authentication in parallel. The user first broadcasts her certificate, and then sequentially receives and responses the challenge from each local sensor. A simple scheduling algorithm can be used for the endorsing sensors to send challenges without packet collision. In our implementation, we arrange the endorsing sensors to send the challenge in ascending order of their group IDs. If the user is successfully authenticated, then each endorsing sensor generates the endorsement and returns it to the user. After collecting all k endorsements from local sensors, the user finally generates the digest *mac* and sends the access request to the remote sensor. We perform the experiment with k changing from 2 to 16. The result of endorsing time consumption is shown in Fig. 7. Note that the time duration includes the time for user's broadcasts for request, receiving the group id reply from sensors, unicasts to sensors for acknowledging receiving their group IDs, and sensor nodes' data processing time to generate the endorsements.

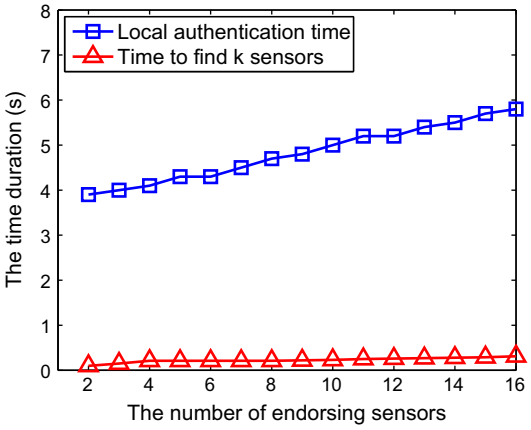


Fig. 7. The upper line shows the time duration for the user to get authenticated by k local sensor, k is changing from 1 to 16. The lower line reveals the time delay for the user to find k endorsing sensors.

We first perform a separate experiment just to test the time delay to find k sensors only (without local authentication and endorsement generation). The result is shown as the dotted line in the same Fig. 7. It is interesting to find that it takes 105 ms to find just 2 endorsing sensors and considerable time for discovering 4, 8, and 16 sensors, which is surprisingly slow, considering 1 ms transmitting/receiving delay. Two factors contribute to the long delay. First, as discussed in previous section, the user may not get all information from local endorsing sensors after the first broadcast. The user may have to broadcast the request more than twice. Second, more importantly, a timer is set between any two broadcasts in our implementation to regulate the packet transmission and reception. Every time the timer fires, the user checks whether the endorsing list is complete. If not complete, the user will do broadcast again. The time delay between the fires of the timer predominantly accounts for the sensor discovery delay. We can reduce this time duration by setting a higher timer frequency.

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

The total endorsing time is presented in Fig. 7. Apparently, the expensive local authentication still dominates other delays. However, because k local sensors authenticate the user in parallel, the total endorsing time is practical and not much longer than the local authentication delay. When $k = 16$, it only takes 5.8 s for the user to get all endorsements.

Although the endorsement can be done in parallel, the energy consumption has to be calculated for each endorsing sensor individually and will increase linearly as the number grows. For energy consumption in processing, we ignore the sensor discovery period because authenticate time dominates the latency, so the energy cost is simply the product of the time latency in Fig. 7 and 3.0×1.5 mW (with the radio off). The energy cost for communication can be estimated in a similar way as in the local authentication. As indicated in Fig. 3, in addition to the authentication, each endorsing sensor sends an extra 48-byte *mac*, which costs extra $48 \times 8 \times 18 = 6.9$ mJ.

Upon receipt of the user remote access request, the remote sensor has to verify whether the request is endorsed by k local sensors. To do so, the remote sensor reconstructs k secret keys by using the received group IDs and its own share of polynomial. These derived secret keys are immediately used to generate the endorsements and finally verify the digest *mac*. In the experiment, we measure the time duration for the remote sensor to do the verification with $k = 4, 5, \ldots, 16$ endorsing sensors. The experiment results are shown in Fig. 8. The corresponding energy consumption at the remote sensor can also be calculated given the processing time and the communication data size.

Finally, we estimate the total time and the power consumption for a user to be authenticated for remote data access. Suppose the network requires the user to get 16 endorsing sensors to access a remote sensor. First, the user has to get local authentication by all 16 local sensors and receive corresponding endorsements. This procedure costs 5.8 s according to Fig. 7. Then, the remote sensor needs 283 ms to reconstruct and verify 16 endorsements. In total, a remote access with 16 local sensor endorsements will cost around 6 s. Note that our estimation does not include the message traveling time from the user to the remote sensor and then back to the user. The power consumption for the remote access control is also plotted in the above figure, which shows the combined amount of the power consumed for processing and communication, including the local authentication at the endorsing sensors and the processing at the remote sensor. We notice the remote access control consumes much more energy then the local access control. When the number of endorsing sensors is 16, almost 500 mJ energy is need to support a remote access control (not including the message transmission between the user and the remote sensor).

6.4. Performance comparison with a centralized scheme

Finally, we investigate the performance comparison between the proposed user access control and a centralized user access control scheme. Without loss of generality, we describe a centralized scheme in the following way. A user enters the sensor field and queries a nearby sensor for the information. The queried sensor has no information about the user, and has to forward the request to the base station through the multi-hop communication. The base station verifies the user and then sends the reply to the corresponding sensor. The sensor finally either replies the user with the queried information or rejects the access based on the base station's response. For the convenience, we only compare the local user access control.

Two performance metrics, query response time and the network energy consumption, are used in the performance comparison. The query response time indicates the responsiveness of the user query by the sensor system. The energy consumption, in the other hand, measures the total network energy cost to support the query.

In the centralized scheme, we ignore the user verification processing delay in the base station since the base station is a resource-rich computer. The user response time thus is the communication latency between the queried sensor and the base station, which is determined by the hop-count distance. In an ideal situation where there is no congestion and radio communication disturbance, the communication delay is the product of the hop count

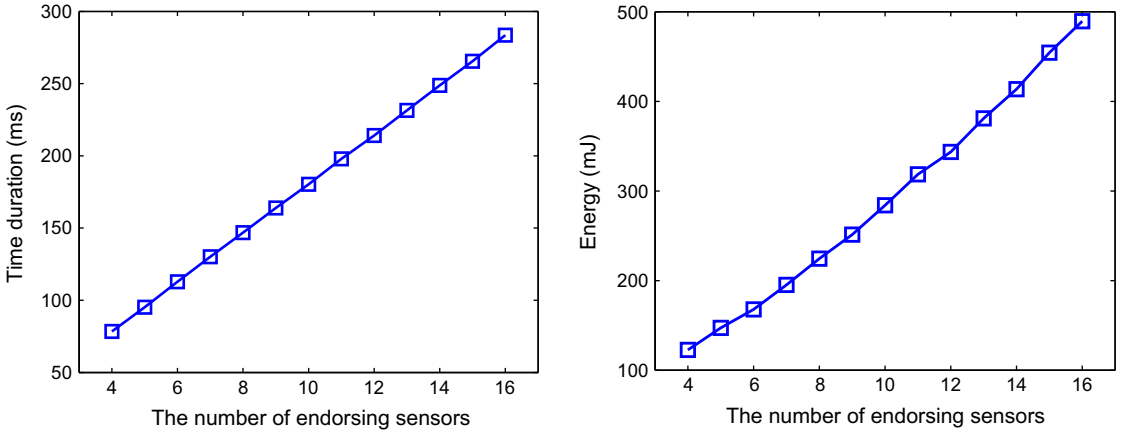


Fig. 8. The time duration and energy cost for the remote sensor to verify k endorsing local sensors.

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

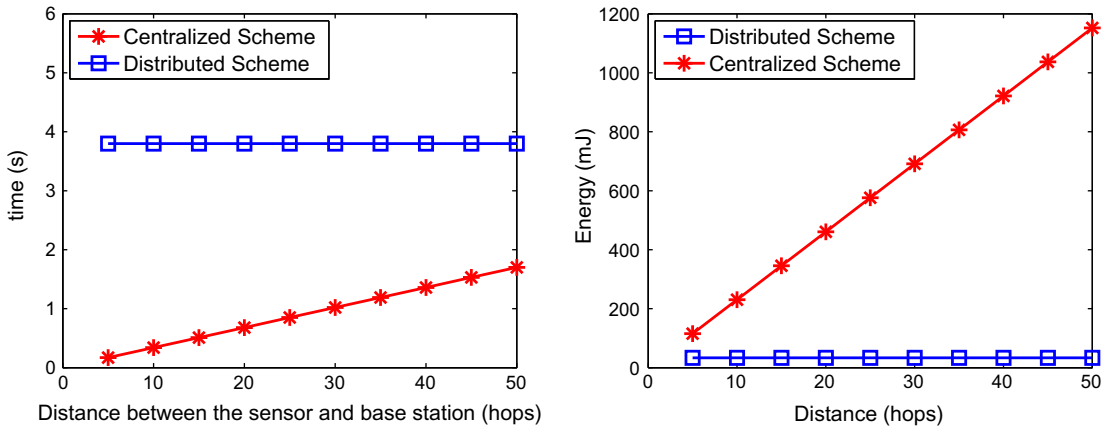


Fig. 9. The comparison of time duration and energy cost.

with the transmission time in each hop. Our experiment shows that on average it takes about 17 ms to transmit a 40-byte data packet on a TelosB mote. Thus, we can easily plot the figure of the user response time as shown below in Fig. 9.

Obviously, the centralized scheme has the edge in user response time when the network size is relatively small. When the sensor is 50 hops away from the base station, the user response time in the centralized scheme is only half of that in the distributed scheme. However, the distributed user access control significantly reduces the network energy consumption compared to the centralized counterpart. Regardless of the network size, the distributed scheme only consumes 33.2 mJ for each user local query. The energy consumption is proportional to the distance between the queried sensor and the base station. When the sensor is 50 hops away from the base station, the centralized scheme consumes 34 times more energy than the distributed approach.

Comparing the 3.8 s user response time and the network energy consumption. We argue that the latter is a more important factor in the sensor network design. Further, the user response time estimation for the centralized scheme is under an ideal situation that there is no communication loss and no network congestion. In practice, the communication latency would be longer due the wireless channel fluctuations. More importantly, as we previously indicated in Section 1, the communication pattern in the centralized scheme is vulnerable to the adversary's DoS attacks. By compromising any one sensor, the adversary can easily flood the network by sending a large amount of messages through the compromised sensor and deplete the battery power of the sensor nodes. Combining all above factors, we believe the distributed scheme scores a clear-cut win over the centralized scheme.

7. Conclusion

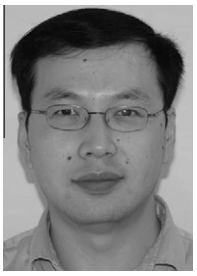
In this paper, we show our effort in designing access control scheme for sensor networks. We describe our local access control and remote access control under a very realistic adversary model. We also discuss the revocation scheme to efficiently deprive a misbehaving user of her access right. Finally, We implement the protocols on a TelosB mote test-bed. The security and performance analysis and the experimental results show that our access control schemes are efficient and feasible for real world applications.

References

[1] Burton Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of ACM 13 (7) (1970) 422–426.
[2] D. Carman, B. Matt, P. Kruus, D. Balenson, D. Branstad, Key Management in Ditributed Sensor Networks, in: DARPA Sensor IT Workshop, 2000.
[3] H. Chan, A. Perrig, D. Song, Random Key Predistribution Schemes for Sensor Networks, in: IEEE Symposium on Security and Privacy, Berkeley, California, 2003, pp. 197–213 (May).
[4] Moteiv Co. Telos Datasheet. <http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf>.
[5] W. Du, J. Deng, A pairwise key pre-distribution scheme for wireless sensor networks, in: ACM CCS, 2003.
[6] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: ACM CCS, 2002 (November).
[7] N. Gura, A. Patel, A. Wander, H. Eberle, S.C. Shantz, Comparing elliptic curve cryptography and RSA on 8-bit CPUs, in: CHES, Cambridge, MA, 2004 (August).
[8] D. Hankerson, A.J. Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, Springer-Verlag, 2004.
[9] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: ACM CCS, Washington, DC, 2003 (October).
[10] Donggang Liu, Efficient and distributed access control for sensor networks, in: The International Conference on Distributed Computing in Sensor Systems (DCOSS), 2007 (June).
[11] A. Perrig, J. Stankovic, D. Wagner, Security in wireless sensor networks, Communications of The ACM 47 (6) (2004) 53–57. June.
[12] A. Perrig, R. Szewczyk, V. Wen, D. Culler, D. Tygar, SPINS: security protocols for sensor networks, ACM/Kluwer Wireless Networks Journal (WINET) (September) (2002).
[13] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, F. Yu, Data-centric storage in sensornets with GHT: a geographic hash table, Mobile Networks and Applications 8 (4) (2003) 427–442.
[14] Kui Ren, Wenjing Lou, Privacy enhanced access control in pervasive computing environments, in: Proceedings of BroadNet, October 2005.
[15] Ronald L. Rivest, The RC5 encryption algorithm, in: Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, Springer, 1995, pp. 86–96.

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011

[16] P. Traynor, H. Choi, G. Cao, S. Zhu, T.L. Porta, Establishing pair-wise keys in heterogeneous sensor networks, in: INFOCOM, Barcelona, Spain, 2006 (April).
[17] P. Traynor, R. Kumar, H.B. Saad, G. Cao, T.L. Porta, LIGER: implementing efficient hybrid security mechanisms for heterogeneous sensor networks, in: MOBISYS, Uppsala, Sweden, 2006 (June).
[18] H. Wang, Q. Li, Achieving robust message authentication in sensor networks: a public-key based approach, ACM Journal of Wireless Networks (WINET) (2009).
[19] Haodong Wang, Bo Sheng, Chiu C. Tan, Qun Li, WM-ECC: an Elliptic Curve Cryptography Suite on Sensor Motes, Technical Report WM-CS-2007-11, College of William and Mary, Computer Science, Williamsburg, VA, 2007.
[20] F. Ye, H. Luo, S. Lu, L. Zhang, Statistical En-Route Filtering of Injected False Data in Sensor Networks, in: INFOCOM, 2004.
[21] Zhen Yu, Yong Guan, A key pre-distribution scheme using deployment knowledge for wireless sensor networks, in: The 4th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Los Angeles, CA, USA, 2005.
[22] Zhen Yu, Yong Guan, A dynamic en-route scheme for filtering false data in wireless sensor networks, in: INFOCOM, Spain, 2006 (April).
[23] W. Zhang, H. Song, S. Zhu, G. Cao, Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks, in: MOBIHOC, Chicago, IL, 2005 (May).
[24] Y. Zhang, W. Liu, W. Lou, Y. Fang, Location-based compromise-tolerant security mechanisms for wireless sensor networks, IEEE Journal on Selected Areas in Communications 24 (2) (2006) 247–260.
[25] S. Zhu, S. Setia, S. Jajodia, P. Ning, An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks, in: IEEE Symposium on Security and Privacy, Oakland, CA, 2004 (May).



Haodong Wang is an assistant professor of Computer and Information Science at Cleveland State University. He received his PhD in Computer Science at College of William and Mary, VA, USA, in Aug 2009. He also earned his Master of Science in Electrical Engineering from Penn State University, University Park, PA, USA, and Bachelor of Engineering in Electronic Engineering from Tsinghua University, Beijing, China. Before joining Cleveland State University, Haodong was an Assistant Professor in the Department of Math and Computer Science at Virginia State University. His main research focuses on pervasive computing, including security and privacy in wireless embedded networks, efficient information management system, privacy-aware routing in wireless sensor networks, mobile 802.11 WLAN performance enhancements and cognitive radio MAC design.

Qun Li is an assistant professor in the Department of Computer Science at College of William and Mary. He holds a PhD degree in computer science from Dartmouth College. His research interests include wireless networks, sensor networks, RFID, and pervasive computing systems. He received the NSF Career award in 2008.

Please cite this article in press as: H. Wang, Q. Li, Achieving distributed user access control in sensor networks, Ad Hoc Netw. (2011), doi:10.1016/j.adhoc.2011.01.011