FDAC: Toward Fine-grained Distributed Data Access Control in Wireless Sensor Networks

Shucheng Yu
Department of ECE
Worcester Polytechnic Institute
Email: yscheng@wpi.edu

Kui Ren
Department of ECE
Illinois Institute of Technology
Email: kren@ece.iit.edu

Wenjing Lou
Department of ECE
Worcester Polytechnic Institute
Email:wjlou@ece.wpi.edu

***Abstract*—Distributed sensor data storage and retrieval has gained increasing popularity in recent years for supporting various applications. While distributed architecture enjoys a more robust and fault-tolerant wireless sensor network (WSN), such architecture also poses a number of security challenges especially when applied in mission-critical applications such as battle field and e-healthcare. First, as sensor data are stored and maintained by individual sensors and unattended sensors are easily subject to strong attacks such as physical compromise, it is significantly harder to ensure data security. Second, in many mission-critical applications, fine-grained data access control is a must as illegal access to the sensitive data may cause disastrous result and/or prohibited by the law. Last but not least, sensors usually are resource-scarce, which limits the direct adoption of expensive cryptographic primitives. To address the above challenges, we propose in this paper a distributed data access control scheme that is able to fulfill fine-grained access control over sensor data and is resilient against strong attacks such as sensor compromise and user colluding. The proposed scheme exploits a novel cryptographic primitive called attribute-based encryption (ABE), tailors, and adapts it for WSNs with respect to both performance and security requirements. The feasibility of the scheme is demonstrated by experiments on real sensor platforms. To our best knowledge, this paper is the first to realize distributed fine-grained data access control for WSNs.**

I. Introduction

Wireless sensor networks (WSN) have been an area of significant research in recent years [1]–[5]. A WSN usually consists of a large number of sensor nodes that can be easily deployed to various terrains of interest to sense the environment. WSNs have found their wide applications in both civilian and military domains. To accomplish the targeted application and fulfill its functionalities, a WSN usually generates a large amount of data continuously over its lifetime. One of the biggest challenge then is how to store and access these sensed data.

Data storage and access in WSNs mainly follows two approaches, namely, centralized and distributed approaches [6]. In the centralized case, sensed data are collected from individual sensors and transmitted back to a central location, usually the sink, for storage and access. In the distributed approach, after a sensor node has generated some data, it stores the data locally or at some designated nodes within the network, instead of immediately forwarding the data to a centralized location out of the network. The stored data later on can be accessed in distributed manner by the users of the WSN. Compared to the centralized case, distributed data storage and access consumes less bandwidth since sensed data are no longer necessarily transmitted to a centralized location out of the network. As energy-efficient storage devices are now possible to be equipped with sensor nodes [7]–[10] thanks to recent advances in IC manufacturing, reading data from local storage becomes much more efficient than transmitting over radio. Employment of distributed data storage and access thus also implies energy-efficiency. In addition, distributed data storage and access can avoid weaknesses such as single point of failure, performance bottleneck, which are inevitable in the centralized case. These advantages together have led to recent increasing popularity of distributed data storage and access.

As a large amount of sensed data are distributedly stored in individual sensor nodes, data security naturally becomes a big concern. Actually, in many application scenarios data sensed by WSNs are closely related to security and/or privacy issues and should be accessible only to authorized users. Moreover, in a mission-critical application scenario various types of data generated by all kinds of sensors may belong to different security levels, and thus are meant to be accessed only by selected types of users. That is, accessibility of a particular type of data to users is based solely on necessity. For example, the general in the battle field scenario should be able to access all types of data for the purpose of overall coordinating but a solider may only need to access the type of data relevant to his mission. In this way, the security of data can be best protected as, for example, a soldier has much larger risk being compromised as compared to the general, and a tank is much better protected than a simple mobile sink. With such a fine-grained data access control, we can effectively minimize the negative consequence due to user compromise. However, past research on data security mainly focused on communication security, such as key management, message authentication, intrusion detection, and etc [11]–[14]. Distributed data storage and access security has gained limited attention so far, not to mention fine-grained data access control. This becomes a more severe issue given the trend that more and more distributed data storage and retrieval schemes are being proposed.

To provide distributed data access control, a naive solution is to equip each sensor node with an access control list (ACL) as is usually adopted in wired networks. Upon each data access request, the sensor node verifies the user's identity with the

ACL, and the access request is approved only if the user is in the list. However, this naive solution is not applicable to WSNs due to the following facts. First, sensor nodes are often deployed without physical protection and lack of tamper-resistant hardware. Attackers may capture and compromise sensor nodes, and then read historical data stored in the sensor nodes. Second, the ACL method is not scalable as it requires the sensor nodes to remember each legitimate user. For the purpose of finding a secure yet efficient solution for fine-grained distributed data access control in WSNs, we naturally shift our attention to data encryption which would introduce two branches, namely symmetric key cryptography (SKC) based approaches and public key cryptography (PKC) based ones.

In SKC based approaches, data encryption and decryption share the same key. If the attacker has compromised the sensor node, he is able to read the data encryption key stored in the sensor's memory and thus decrypt the historical data generated by the same sensor. To avoid this kind of attacks, a natural solution is to divide the lifetime of each sensor into series of periods, and the data encryption keys for these periods are independent of each other. Each sensor just stores the data encryption key for current period, and erases all the previously used keys. The problem that follows is to efficiently update data encryption keys for sensor nodes as well as distribute the keys to legitimate users. State-of-the-art SKC based approaches adopt techniques such as perturbed polynomial [15] to manage the keys. However, current SKC based approaches have two major drawbacks: first, fine-grained data access control is hard to realize due to the complexity introduced by key management; second, collusion attacks are possible given an appropriate number of colluding users. Therefore, further research is still desired for fine-grained distributed data access control using SKC based approaches.

PKC-based approaches can provide better data access security than their SKC-based peers. In such approaches, sensor nodes encrypt the data items with public keys. One apparent advantage of this is that if data storage sensors are compromised, the attacker will not be able to recover the stored data due to lack of the corresponding private keys. Therefore, by applying PKC-based approaches to data access control in WSNs, we can immediately enjoy the perfect resilience against sensor compromise. In traditional public key cryptosystems including identity-based encryption, the encryption is usually targeted to only one recipient, in the sense that any message encrypted using a particular public key can be decrypted only with the corresponding secret key. However, for the purpose of distributed data access control in WSNs, the fundamental encryption paradigm is one-to-many such that one encrypted data item can be decrypted by a number of different authorized users. To achieve this goal, a straightforward approach is to use one-to-one public key cryptosystems, which is obviously inefficient since both the number of encryption operations and the size of ciphertexts are linear to the total number of authorized users. A better solution is broadcast encryption [16]–[19], which achieves improved efficiency. But it requires that receivers are represented individually. An encryptor must have the priori knowledge of all the prospective receivers as well as the authorization information associated with each receiver. However, in our targeted applications, it is desirable for sensors to be able to encrypt without exact knowledge of the set of intended receivers. This is because when WSN is deployed, it may be impossible to know the exact information of its future users. It may only be possible to know certain priori attributes of its users. Therefore, neither solution can be applied in our scenario.

From the above discussion, it is clear that achieving fine-grained data access control with efficiency is still an open challenge in WSNs. Towards addressing this challenge, we propose in this paper a Fine-grained Distributed data Access Control scheme, namely FDAC, specially tailored for WSNs. We base our design on the observation of the inherent nature of the sensor data. As WSNs are in general deployed for specific application(s), it is usually easy and convenient to specify individual sensors (and hence their collected data) through a set of predefined attributes such as sensor type, location, time, owner, etc. We further find that this nice property can also be utilized to describe data accessibility in a very expressive manner, that is, it can allow fine-grained tunable data access control. Based on this observation, we propose to associate each attribute of sensor nodes with a predefined keying material. And then we further examine each user of the WSN with respect to their data access privileges and associate him with an access structure accordingly. Such an access structure in our design is implemented via an access tree which specifies the types of data that this user is authorized to access. Sensor data are then protected by being encrypted under their attributes such that only the users whose access structures satisfy the required data attributes can decrypt. In the access structure, every leaf node maps to a sensor/data attribute, and the interior nodes can be threshold gates. The access structure thus can represent sophisticated logic expressions over the attributes, that is, be able to specify data access privileges of users in the fine-grained manner. By exploring a novel PKC primitive called key-policy attribute-based encryption (KP-ABE), we seamlessly integrate our access structure with data encryption. Our solution has several advantages. First, FDAC is efficient in terms of key storage, computation and communication overhead at the sensor node side. Second, it is resistant against user collusion, i.e., the cooperation of colluding users will not lead to the disclosure of additional sensor data. Last but not least, FDAC provides efficient user revocation via a single broadcast, and the length of the broadcast message is only of several hundred bits.

In summary, our paper makes the following contributions. 1) It introduces the fine-grained data access control problem for the first time in WSNs. 2) FDAC applies and tailors KP-ABE to WSNs for achieving fined-grained access control. 3) The applicability of FDAC is demonstrated on the current generation of sensor nodes.

The rest of this paper is organized as follows. Section II discusses our system models and assumptions as well as

some technical preliminaries on which our scheme is based. In section III, we present our scheme in detail. Section IV analyzes our scheme in terms of security and performance. We conclude this paper in section V.

II. Models and Assumptions

A. Network Model

In this work, we consider a wireless sensor network composed of a network controller which is a trusted party, a large number of sensor nodes, and many users. Throughout this paper, we will denote the network controller with the symbol $\mathcal{T}$. Symbol $\mathcal{U}$ and $\mathcal{N}$ are used to represent the universe of the users and the sensor nodes respectively. Both users and sensor nodes have their unique IDs. Symbol $\mathcal{U}_i$ will be used to denote user i, and $\mathcal{N}_i$ is defined similarly. The trusted party $\mathcal{T}$ can be online or off-line. It comes online merely on necessity basis, e.g., in the case of intruders detected. Each sensor could be a high-end sensor node such as iMote2 which has greater processing capability and a larger memory than conventional sensor nodes. Sensor data could be stored locally or at some designated in-network location using data storage schemes such as TTDD [20]. As is conventionally assumed, we consider a user $\mathcal{U}$ to have sufficient computational resources to execute some expensive cryptographic operations. In addition, we assume there is a loose time synchronization among the sensor nodes.

B. Adversary Model

This paper considers attackers whose main goal is to obtain sensor data which they are not authorized to access. The adversaries could be either external intruders or network users who are unauthorized to access the target type of data. Due to lack of physical protection, sensor nodes are usually vulnerable to strong attacks. In particular, we consider the adversary with both passive and active capabilities, which can (1) eavesdrop all the communication traffics in the WSN, and (2) compromise and control a small number of sensor nodes. In addition, (3) unauthorized users may collude to compromise the encrypted data.

C. Security Requirements

For the purpose of securing distributed data storage, some common security aspects such as data confidentiality and integrity, which are also desired in any other WSN security scheme, should be provided. With respect to data access control in WSNs, we recognize the following unique but not necessarily complete security requirements.

- *Fine-grained Data Access Control:* As is mentioned in the previous section, fine-grained data access control is often desired by many mission-critical application scenarios. To provide fine-grained data access control, the proposed scheme should provide a strategy that is able to precisely specify the capability of different kinds of users to access sensor data of different types or security levels.
- *Collusion Resilience:* As described by our adversary model, unauthorized users may cooperate for the purpose of obtaining the sensitive sensor network data. Therefore, it is very critical to equip our data access control scheme with the resilience against collusion attacks such that the cooperation of the unauthorized users will not give them additional advantages over what they can directly obtain from executing attacks individually.
- *Sensor Compromise Resistance:* Due to lack of compromise-resistant hardware, a small number of sensor nodes are inevitably to be compromised by the adversary in hostile environments. Now that the adversary can always obtain the sensor data generated by a sensor node after it is compromised, we should at least secure sensor data such that, (1) compromising the sensor node does not disclose the sensor data generated before the sensor is compromised, and (2) compromising one sensor node does not give the adversary any assistance to obtain sensor data generated by other sensor nodes.
- *Backward Secrecy:* User management is an important functionality required by most application scenarios. In particular, the system should be able to handle user revocation in the case of user leaving request or malicious behavior detected. To support such a functionality, the data access control mechanism should guarantee that the revoked users are not able to access the sensor data generated after they are revoked.

D. Preliminaries

This section briefly describe the technique preliminaries on which our scheme is designed.

1) Bilinear Map: Our design is based on some facts about groups with efficiently computable bilinear maps.

Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order *p*. Let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. A bilinear map is an injective function $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following properties:

1. *Bilinearity*: for $\forall\ u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.

2. *Non-degeneracy*: $e(g_1, g_2) \neq 1$.

3. *Computability*: There is an efficient algorithm to compute $e(u, v)$ for each $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$.

2) Key-Policy Attribute-Based Encryption: In KP-ABE [21], each ciphertext is associated with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of ciphertexts the key can decrypt. A user is able to decrypt a ciphertext if and only if the attributes associated with a ciphertext satisfy the key's access structure. A KP-ABE scheme is composed of four algorithms:

Setup This algorithm takes as input a security parameter κ. and returns the public key PK as well as a system master secret key MK. PK is used by message senders for encryption. MK is used to generate user secret keys and is known only to the authority party.

Encryption This algorithm takes a message m, the public key PK, and a set of attributes γ as input. It outputs the ciphertext E.

Key Generation This algorithm takes as input an access structure $\mathcal{P}$, the master secret key MK, and the public key PK. It outputs a secret key SK that enables the user to decrypt a message encrypted under a set of attributes γ if and only if γ matches $\mathcal{P}$.

Decryption It takes as input the ciphertext E, which was encrypted under the attribute set γ, the user's secret key SK for access structure $\mathcal{P}$, and the public key PK. This algorithm outputs the message m only if the attribute set γ satisfies the user's access structure $\mathcal{P}$.

Please refer to [21] for more details on KP-ABE algorithms.

III. FDAC: Fine-grained Data Access Control Scheme

This section presents our data access control scheme for distributed data storage. We first introduce our access control strategy. Next, we give an overview of FDAC. Then, we illustrate the detailed description of our basic scheme, which is followed by an advanced design.

A. Access Control Strategy

For the purpose of achieving fine-grained data access control in WSNs, we need to first explore the inherent natures of sensor networks. In general, the deployments of most WSNs are aimed at data collection for specific application(s). Therefore, we are able to specify individual sensors (and hence the data collected by them) through a set of predefined attributes. For example, in the battlefield, sensor nodes are usually deployed to collect military information in certain geographic location. Each sensor node may be responsible for collecting specific types of data, e.g., vibration, smoke, so on and so forth. Sensor nodes may also have their owners, i.e., persons or units who are in charge of them. In particular, some nodes may be jointly owned by different units. Hence, we may specify sensor nodes using these attributes, e.g., {location = *village*, data type = (*vibration*, *smoke*), owner = (*explosion experts*, *officers*, *scouts*)}. This further enables us to specify data access privileges of users based on these attributes. In the above example, we may designate the access structure of a user as "(location is village) AND (type is vibration)", which allows the users to obtain vibration data within the village area. We may also define more sophisticated access structures such as "(location is village) AND (type is vibration OR smoke) AND (at least owned by 2 of the following: explosion experts, officers, scouts)". In this case, the user can only access vibration and smoke data collected within the village area. In addition, the last condition implicitly requires the user to belong to at least two of the three designated groups. With these fine-grained access structures explicitly defined, the remaining is to seek a way forcing the users to their respective predefined rules. To achieve this, we can predefine keying materials for each of the attributes, and encrypt sensor data under the keys corresponding to their attributes such that only those whose access structures "accept"[1] the data attributes are able to decrypt.

[1]That is to say, the logic expression of the access structure returns TRUE.

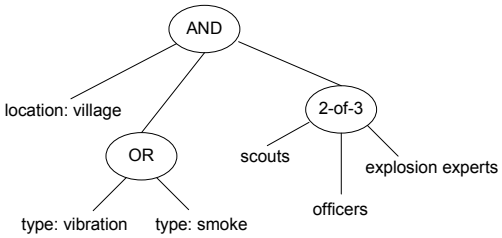

Fig. 1. An example access structure in the battlefield scenario

Having demonstrated the intuitive idea of our access control strategy, we further present our access control strategy more formally as follows. In FDAC, we associate each sensor node (and hence his collected data) a set of attributes, for each of which we define a public key. Each user is assigned an access structure, which is implemented via an access tree and embedded in the user secret key. Every leaf node of the access tree is labeled with an attribute and the interior nodes are threshold gates [2]. Access structures are thus able to be represented using the logic expressions over the attributes. In particular, this kind of definition enables access structures to represent sophisticated logic expressions, and be able to specify data access privileges of users in the fine-grained manner. Actually, we are able to represent any general access structures if we define the NOT of an attribute as a separate attribute, which in turn will double the number of attributes in our system. Fig 1 illustrates the aforementioned access structure in the battlefield scenario.

Notably, the above fine-grained data access control can also be realized in an alternative way as follows. We can define a set of attributes for each user, and associate each sensor node with an access structure which designates the logic combination of the intended attributes of the target receivers. Sensor data later on are encrypted under the access structure such that only those whose attributes satisfy the access structure are able to decrypt. In this way, we are able to realize the same functionality in terms of fine-grained data access control. However, this strategy may not be applicable to WSNs due to the concerns on the performance in terms of computation and communication overload. In our proposed strategy, the complexity in terms of computation and communication overload is linear to the number of data attributes assigned to the sensor node. In practice, this complexity could be arguably low due to the fact that each sensor node (and hence their collected data) can be specified via a small number of attributes, though their universe in the whole network could be large. On the contrary, the complexity in the alternative strategy is determined by how complicated the access structure is. In practice, this complexity could be extremely high if we assign a complicated access structure to the sensor node for the expressiveness purpose. Because sensor nodes are usually not resource abundant, the alternative strategy is not suitable for WSNs.

[2]A t-of-n threshold gate is satisfied if and only if at least t out of the n inputs are satisfied. Two extreme examples are AND gates (n-of-n) and OR gates (1-of-n).

Formally, in FDAC we will denote the universe of all the sensor attributes in a WSN application by a symbol $\mathcal{I}$. The set of attributes owned by each single user is denoted by a symbol $\mathcal{I}_i$, where i is the sensor node ID. We have $\mathcal{I} = \bigcup_{\forall i \in \mathcal{N}} \mathcal{I}_i$. Let $k = \max_{\forall i \in \mathcal{N}} |\mathcal{I}_i|$. k will be a system parameter used by our scheme. The access structure is generally denoted by $\mathcal{P}$.

B. Scheme Overview

In our basic scheme, each sensor node is preloaded with a set of attributes as well as the public key PK. Each user is assigned an access structure and the corresponding secret key SK. The lifetime of the sensor network is divided into m *stages*, numbering as $1, 2, \cdots, m$. The stage number is reset to 1 when it increases to $m+1$. Each period is further divided into n *phases*, numbering as $1, 2, \cdots, n$, where we set $n < k$, $k = \max_{\forall i \in \mathcal{N}} |\mathcal{I}_i|$. Sensor nodes encrypt and store sensor data on the phase basis. For each sensor node, the sensor data are encrypted by symmetric encryption such as AES, and the data encryption keys during one stage form an one-way key chain, one data encryption key for each phase. We call the first key on this key chain by the $master\ key$, denoted by K. The master key of each stage is always generated during its preceding stage, and encrypted under the preloaded attributes. Upon request for sensor data, the sensor node responds with the encrypted master key as well as the ciphertext of the sensor data. If the user is an intended receiver, he is able to decrypt the master key and derive the data encryption key, and then obtain the sensor data. Based on the basic scheme, our advanced scheme goes one step further by providing the functionality of user revocation, which is demanded by most WSN application scenarios. In the advanced scheme, $\mathcal{T}$ is able to revoke any user via broadcasting a user revocation message to all the users and all the sensor nodes respectively. In particular, the user revocation message for the sensor nodes contains merely a group element on $\mathbb{G}_T$.

C. The Basic Scheme

1) System Initialization: On initialization, $\mathcal{T}$ executes the following steps:

a) Select two multiplicative cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_T$ of prime order p as well as a bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$. Let g be the generator of $\mathbb{G}_1$.

b) Choose a number t_i uniformly at random from $\mathbb{Z}_p$ for each attribute $i \in \mathcal{I}$, and y randomly from $\mathbb{Z}_p$. Output the public key as follows:

$$PK = \langle G_1, g, Y = e(g,g)^y, T_1 = g^{t_1}, \cdots, T_{|\mathcal{I}|} = g^{t_{|\mathcal{I}|}} \rangle$$

The master secret key is $MK = (y, t_1, \cdots, t_{|\mathcal{I}|})$.

c) Choose a secure one-way hash function, denoted as $h(\cdot)$.

d) Preload the following information to each sensor node $\mathcal{N}_i$:

$$\mathcal{T} \rightarrow \mathcal{N}_i: \ \mathcal{I}_i, \ h(\cdot), \ \langle G_1, g, Y, \{T_x\}_{x \in \mathcal{I}_i} \rangle$$

e) For each user $\mathcal{U}_j$, $\mathcal{T}$ first generates an access structure $\mathcal{P}$ and computes his secret key SK as follows. Starting from the root node r of $\mathcal{P}$ and in the top-down manner, $\mathcal{T}$ constructs a random polynomial q_x of degree $d_x + 1$ using Lagrange interpolation for each node x in $\mathcal{P}$, where d_x is the degree of node x. For each non-root node x in $\mathcal{P}$, it sets $q_x(0) = q_{parent(x)}(index(x))$, where $parent(x)$ is the parent of x and x is the $index(x)^{th}$ child of its parent. In particular, $q_r(0) = y$. SK is output as follows

$$SK = \langle \{D_i = g^{\frac{q_i(0)}{t_i}}\}_{i \in \mathcal{L}} \rangle$$

where $\mathcal{L}$ denotes the set of leaf nodes in $\mathcal{P}$. Then, $\mathcal{U}_j$ is preloaded with the following information

$$\mathcal{T} \rightarrow \mathcal{U}_j: \mathcal{P}, \ SK, \ h(\cdot)$$

2) Master Key Encryption: In each stage, say stage $v \in [1, m]$, $\mathcal{N}_i$ generates a new master key for the next stage, i.e., stage $v + 1 \ mod \ m$, and encrypts it as follows:

a) Select a number s uniquely at random from $\mathbb{Z}_p$.

b) In each phase, calculate one item $E_i = T_i^s$ for attribute $i \in \mathcal{I}_i$. After $|\mathcal{I}_i|$ phases, $|\mathcal{I}_i| \leq k < m$, $\mathcal{N}_i$ has the complete set $\{E_i = T_i^s\}_{i \in \mathcal{I}_i}$.

c) Randomly select a number $K \in \mathcal{K}$ as the master key of the key chain, where $\mathcal{K}$ denotes the key space. Then, compute $E' = KY^s$. Finally, store the ciphertext as follows: $E^{v+1} = \langle v+1 \ mod \ m, \mathcal{I}_i, KY^s, \{E_i = T_i^s\}_{i \in \mathcal{I}_i} \rangle$, where E^{v+1} represents the encrypted master key for the $(v+1)^{th}$ stage.

3) Data Storage: $\mathcal{N}_i$ encrypts and stores the sensor data generated in the current phase, namely phase $t \in [1, n]$ of stage $v \in [1, m]$, as follows:

a) Calculate the data encryption key $K_t = h(K_{t-1})$. In particular, we set $K_0 = K$.

b) Encrypt the sensor data, denoted by D, with current data encryption key K_t. Then, it stores the item $\langle v, t, \{D\}_{K_t} \rangle$, where $\{D\}_{K_t}$ represents the encrypted sensor data.

c) Erase K_{t-1} from the memory.

For each sensor node, all the data encryption keys used during one stage form an one-way key chain. The sensor node just keeps the latest data encryption key in his memory, while erasing all the previous ones.

4) Data Access: Assume user $\mathcal{U}_j$ is requesting for sensor data generated by sensor node $\mathcal{N}_i$ during phase t of stage v. $\mathcal{N}_i$ responds the data query request with the following message:

$$\mathcal{N}_i \rightarrow \mathcal{U}_j: \langle E^v, \ \{D\}_{K_t} \rangle$$

On receiving the response from $\mathcal{N}_i$, $\mathcal{U}_j$ executes the following steps to obtain the sensor data:

a) Decrypt the master key K of stage v from E^v. The decryption process starts from the leaf nodes and in the bottom-up manner. First, $\mathcal{U}_j$ computes the value F_i for each leaf node i in $\mathcal{P}$ as shown in (1).

$$F_i = \begin{cases} e(D_i, E_i) = e(g,g)^{sq_i(0)}, & \text{if } i \in \mathcal{I}_i \ ; \\ \perp, & \text{otherwise.} \end{cases} \quad (1)$$

Then, it proceeds in the recursive way from the second last layer as follows: for node x which is a d_x-of-n gate, if more than $n - d_x$ children returns $\perp$, $F_x = \perp$. Otherwise,

$$F_x = \prod_{i \in S_x} F_i^{\delta_i(0)} = \prod_{i \in S_x} e(g,g)^{sq_i(0)\delta_i(0)} = e(g,g)^{sq_x(0)}$$

where S_x denotes the set of x's children and $\delta_i(0)$ is the Lagrange coefficient which can be calculated by the user himself. If $\mathcal{P}$ "accepts" $\mathcal{I}_i$, $\mathcal{U}_j$ will finally obtain $e(g,g)^{sq_r(0)} = e(g,g)^{sy}$ and the message can be decrypted. Otherwise, the decryption algorithm returns $\perp$.

b) If the decryption algorithm returns $\perp$, terminate. Otherwise, $\mathcal{U}_j$ calculates the data encryption key as follows: $K_t = h^t(K)$.

c) Decrypt the sensor data with K_t.

In this basic scheme, we assign each sensor node a set of attributes and each user an access structure. Sensor data are encrypted under the attributes such that only the users whose access structures "accept" these attributes can decrypt. As the access structure is very expressive, we are able to precisely control the capability of each user on data access, and thus enjoy fine-grained data access control. To alleviate the computation overload, we divide the lifetime of sensor nodes into stages and phases, and distribute the computation overload into each phase. In this way, we make the expensive operations on the attributes affordable to the sensor nodes.

One drawback of this basic scheme is that it does not support user revocation efficiently. In this scheme, user revocation can be realized either by updating the secret keys individually for all the users, or by demanding each sensor node to encrypt data with additional attributes that are not "accepted" by the leaving user's access structure. The former solution is obviously inefficient when there are a large number of users presented in the sensor network. The latter, on the other hand, may cause an increasingly large number of operations on attributes as the number of revoked users increases. However, as user revocation is such an important security requirement particularly for many mission-critical applications, lack of it may cause severe security and/or privacy issues. To solve this problem, we propose an advanced scheme based on the basic one.

D. The Advanced Scheme

The basic idea of our user revocation solution is to enable update of the master secret key y embedded in the user secret key SK via broadcast. If we manage to update SK for legitimate users and prevent the leaving user's SK from being updated, the leaving user will not be able to decrypt any future sensor data. Based on this idea, we update our basic scheme as follows. For brevity, we just present the parts that need to be changed.

1) System Initialization: $\mathcal{T}$ executes the following steps.

a) The same as step a) of 1) in the basic scheme.

b) In addition to the elements generated by step b) of 1) in the basic scheme, $\mathcal{T}$ selects a number β uniquely at random from $\mathbb{Z}_p$. The public key PK and the master secret key MK are then output as follows.

$$PK = \langle G_1,\ g,\ Y,\ \{T_i = g^{t_i}\}_{i \in \mathcal{I}},\ g^{\beta} \rangle$$
$$MK = \langle y, t_1, \cdots, t_{|\mathcal{I}|}, \beta \rangle$$

c) The same as step c) of 1) in the basic scheme.

d) Sensor node $\mathcal{N}_i$ is preloaded with the following

$$\mathcal{T} \rightarrow \mathcal{N}_i :\ \mathcal{I}_i,\ h(\cdot),\ \langle G_1, g, Y, \{T_x\}_{x \in \mathcal{I}_i}, g^{\beta} \rangle$$

e) The process of key generation is similar to step d) of 1) in the basic scheme. $\mathcal{T}$ outputs the user secret key SK as follows.

$$SK = \langle g^{\frac{y-\theta}{\beta}}, \{D_i = g^{\frac{q_i(0)}{t_i}}\}_{i \in \mathcal{L}} \rangle$$

Compared to the basic scheme, this algorithm introduces a new element $g^{\frac{y-\theta}{\beta}}$ into SK, where $\theta = q_r(0)$ is randomly selected from $\mathbb{Z}_p$, and q_r denotes the polynomial for the root node r in $\mathcal{P}$. $\mathcal{U}_j$ is then preloaded with $\langle \mathcal{P},\ SK,\ h(\cdot) \rangle$.

2) Master Key Encryption: Similar to 2) in the basic scheme. The advanced scheme introduces a new element $g^{\beta s}$ into the ciphertext as follows:

$$E^{v+1} = \langle v+1\ mod\ m, \mathcal{I}_i, KY^s, \{E_i = T_i^s\}_{i \in \mathcal{I}_i},\ g^{\beta s} \rangle$$

3) Data Storage: The same as 3) in the basic scheme.

4) Data Access: This part is the same as 4) in the basic scheme except for step a).

a) The decryption process is similar to that in the basic scheme. When the data attributes satisfy the user's access structure $\mathcal{P}$, the user obtains $e(g,g)^{\theta s}$. Then, he decrypts the message as follows.

$$M = \frac{Me(g,g)^{ys}}{e(g^{\frac{y-\theta}{\beta}}, g^{\beta s})e(g,g)^{\theta s}}$$

In this advanced scheme, $\mathcal{T}$ is able to update the master secret key y embedded in the user secret key SK by broadcasting $g^{\frac{\Delta y}{\beta}}$ to the users, where Δy is the incremental of y. In addition, this enhanced design is provably secure against chosen message attacks under the Decisional Bilinear Diffie-Hellman (DBDH) assumption. Due to the space limit, we do not present the proof in this paper. A formal security proof to our enhancement will be available in the full version of this paper. With the above enhancement, we can present our user revocation scheme as follows.

5) User Revocation: To revoke a user $\mathcal{U}_j$, $\mathcal{T}$ needs to update the master secret key y for the sensor nodes as well as the remaining users. The process can be illustrated as follows.

$$\mathcal{T} :\ y' \leftarrow \mathbb{Z}_p,\ \Delta y \leftarrow y' - y,\ Y' \leftarrow e(g,g)^{y'},\ \ g^{\frac{\Delta y}{\beta}}$$
$$\mathcal{T} \rightarrow \mathcal{N} :\ Y'$$
$$\mathcal{T} \rightarrow \mathcal{U} \backslash \mathcal{U}_j :\ g^{\frac{\Delta y}{\beta}}$$

First, $\mathcal{T}$ chooses a random number $y' \in \mathbb{Z}_p$ as the new value of the master secret key y. The incremental is set as $\Delta y = y' - y$. Then, it calculates the new public key $Y' = e(g,g)^{y'}$ and the group element $g^{\frac{\Delta y}{\beta}}$. Finally, $\mathcal{T}$ broadcasts Y' to all the sensor nodes and $g^{\frac{\Delta y}{\beta}}$ to all the users excluding the one to be revoked. Upon receiving the master secret key update message, each sensor node simply replaces the public key Y with Y'. The master key for the next stage will be encrypted under the new public key. Each user updates his secret key as follows: $g^{\frac{y-\theta}{\beta}} g^{\frac{\Delta}{\beta}} = g^{\frac{y'-\theta}{\beta}}$. The master secret key y is thus updated as y'. In this user revocation scheme, one challenging issue is to selectively broadcast $g^{\frac{\Delta y}{\beta}}$ such that all but the leaving users are able to receive it. If the number of users is small, we can realize such a selective broadcast via multiple unicasts. Otherwise, we need to employ an efficient selective broadcast scheme to deliver the update message to a large number of users. For this purpose, we can employ CP-ABE as the fundamental technical tool. CP-ABE operates in the opposite way as compared to KP-ABE in that the encryptor encrypts data under certain access structure, and only users with intended attributes are able to decrypt. Compared to traditional broadcast techniques [16]–[19], CP-ABE has both efficiency and security advantages as is studied in previous work [22]. The rough idea of our selective broadcast scheme comes as follows.

a) When the user is registered at $\mathcal{T}$, he is assigned a set of attributes which are able to uniquely identify the user.
b) To broadcast $g^{\frac{\Delta y}{\beta}}$, $\mathcal{T}$ composes an access structure $\mathcal{P}'$ which enables all but the leaving users to decrypt.
c) $\mathcal{T}$ encrypts $g^{\frac{\Delta y}{\beta}}$ using CP-ABE under $\mathcal{P}'$ and broadcast the ciphertext to all the users.

Due to the space limit, we will not present the technical details on the CP-ABE based selective broadcast. We refer to [23] for details of this kind of solution.

IV. Scheme Evaluation

This section evaluates FDAC in terms of security and performance aspects.

A. Security Analysis

We evaluate the security of our work by analyzing the its fulfillment of the security requirements described in section II.

- *Fine-grained Data Access Control:* To provide fine-grained data access control, the proposed scheme should provide a strategy that is able to precisely control the capability of different kinds of users to access sensor data of different types or security levels. In our scheme, the master key of the key chain in each stage is encrypted under a set of attributes. Without the master key, the adversary is not able to derive the data encryption keys due to the one-wayness of the key chain, which can be guaranteed by choosing secure one-way hash functions such as SHA-1. Our encryption of the master key is provably secure under the DBDH assumption, the proof of which is similar to that of the standard KP-ABE. This turns out that the adversary is not able to decrypt the master key unless he owns the intended access structure. Therefore, FDAC is able to control the accessibility of sensor data to only authorized users. In addition to the security aspect, we can show the "fine-grainedness" of FDAC as follows. In FDAC, the access structure is able to represent complicated logic expressions such as threshold gates as well as expressive predicates such as "temperature $>$ 80". The combination of the threshold gates and the predicates are able to represent sophisticated access structures. In fact, FDAC is able to support general access structures if we define the not of an attribute as a separate attribute, which in turn will double the number of attributes in our system.
- *Collusion Resilience:* To compromise the sensor data, the main task for the colluding users is to decrypt the master key of the target data. Since the master key is encrypted under our enhanced scheme, we have to prove that it is collusion-resistant. Actually, as our scheme is provably secure against chosen message attacks under the DBDH assumption, it implies collusion resilience. Instead of providing the formal security proof, we can sketch the intuitive proof of the collusion-resistance as follows. Recall that the master key is encrypted in the form of $Ke(g,g)^{ys}$. The user has to cancel $e(g,g)^{ys}$ to recover K. To compose $e(g,g)^{ys}$, the only way is to execute the following: $e(g^{\frac{y-r}{\beta}}, g^{\beta s}) = e(g,g)^{ys}/e(g,g)^{rs}$. To extract $e(g,g)^{ys}$, the user should compute $e(g,g)^{rs}$. Actually, for each user, r is randomly and independently selected from $\mathbb{Z}_p$. The secret key from one unauthorized user does not give the other user any help in terms of computing $e(g,g)^{rs}$.
- *Sensor Compromise Resistance:* To meet this security requirement, we should achieve two security goals: (1) compromising the sensor node does not disclose the sensor data generated before the sensor is compromised, and (2) compromising one sensor node does not give the adversary any advantage to obtain data generated by other sensor nodes. We can show the fulfillment of our scheme with respect to these two security goals as follows: (1) In our scheme, each sensor node just keeps current data encryption key in the memory, while erasing the previously used keys. Because of the one-wayness of the key chain, the compromiser is not able to derive the previously used keys from current key. (2) is easy to prove since each sensor node encrypts data independently.
- *Backward Secrecy:* As is described in the previous section, our advanced scheme is able to update the master key y for legitimate users while excluding those to be revoked. Since the new sensor data will be encrypted under the latest master key, the revoked users are not able to decrypt. In addition, the security of the user revocation message can be guaranteed because our user revocation message is assumed to be encrypted under the standard

CP-ABE, given that CP-ABE is provably secure. One problem in our scheme is, the user revocation instruction will not take effort until a new stage starts. Such a delay occurs because the sensor node needs one stage to encrypt the master key under the set of attributes. This delay may differ for different systems. For example, if a system has a stage with 30 phases and each phase is 1 second, the delay will be at the most half a minute. Generally, if a system has a stage with less phases and each phase takes less time, i.e., each sensor node is assigned a smaller number of attributes and has a more powerful computational capability, the delay will be shorter. We leave this delay as a system parameter, and the system designer can adjust this parameter by changing the number of attributes assigned to each sensor node or using a different type of sensor nodes.

In addition to the security goals listed above, there are also some other security requirements such as data integrity and authenticity, which are desired by conventional WSN applications. As this paper mainly focuses on fine-grained data access control, we do not explicitly address all those security problems. In fact, security requirements such as message integrity can be easily supported in our scheme with minor modifications. A challenging issue may be the data authenticity. However, this problem is independent to our interested ones. We refer to some current work such as [24] for the solution to this problem.

B. Performance Evaluation

This section evaluates the performance of FDAC in terms of computation and communication overheads. In our scheme, sensor data are generated and encrypted by sensor nodes, and retrieved and decrypted by users. As sensor nodes are usually resource constrained, they may not be able to execute expensive cryptographic primitives efficiently and thus become the bottleneck of the scheme. For this reason, our evaluation focuses on the performance of sensor nodes. In the following section, we first discuss the numeric results in terms of computation and communication overheads for sensor nodes. Then, we present our implementation results on real sensors.

1) Numeric Result: In FDAC, each sensor node is responsible for the following operations in each stage: (1) generate the master key and encrypt it using our revised KP-ABE, (2) derive the data encryption keys based on the master key, and (3) encrypt sensor data using the data encryption keys. These operations are further distributed to each phase. Specifically, in each phase the sensor node executes at the most one scalar multiplication on elliptic curve, one one-way hash, and one symmetric key data encryption. Table I lists all these operations.

TABLE I
COMPUTATIONAL COMPLEXITY ON EACH SENSOR

	Scalar Mul	Hash	Data Encryption
Each Stage	$\|\mathcal{I}_i\|$ +1	m	m
Each Phase	1 or 0	1	1

On each data retrieval request, the sensor node responds with $\langle E^v, \{D\}_{K_t} \rangle$ for sensor data of phase t in stage v, where E^v contains $|\mathcal{I}_i| + 1$ group elements on $\mathbb{G}_1$ and one on $\mathbb{G}_T$, and $\{D\}_{K_t}$ is the data payload. On user revocation, $\mathcal{T}$ only needs to broadcast one group element of $\mathbb{G}_T$ to all the sensor nodes. The communication overload for each sensor node is shown in Table II.

TABLE II
COMMUNICATION OVERLOAD

Data Retrieval	User Revocation
($\|\mathcal{I}_i\|$ +1) $\mathbb{G}_1$ + 1$\mathbb{G}_T$ + data payload	1$\mathbb{G}_T$

2) Implementation: In our implementation, we choose Tmote Sky and iMote2 as the target platforms. We use SHA-1 as the one-way hash function and AES (supported by CC2420 Radio module of the motes) as the the data encryption algorithm. Our implementation shows that it takes about $0.06ms$ for SHA-1 to execute one hash operation and $0.4ms$ for AES to encrypt 64 bytes data. Our implementation also shows that one scalar multiplication takes several seconds in the worst case. The scalar multiplication operation is thus the bottleneck of the sensor performance. To optimize this operation, the key issue is to find appropriate parameters for the elliptic curve.

In past years, many work have efficiently implemented Elliptic Curve Cryptography (ECC) on various sensor platforms. In these work, elliptic curves are ususally chosen according to standards such as NIST and SECG, which enable most of the optimization methods. Although these elliptic curves serve perfectly for security schemes such as ECDH, ECDSA, et al, they are not pairing-friendly, i.e., they can not be used as bilinear map groups. In FDAC, however, the elliptic curve is required to be pairing-friendly. In current work, most studied pairing-friendly elliptic curves fall into two categories, namely Supersingular (SS) curves and MNT curves. In the case of SS curves, the two elliptic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ in section II.D could be the same. For MNT curves, $\mathbb{G}_1$ and $\mathbb{G}_2$ are different. To choose an appropriate elliptic curve, several factors should be taken into account as follows. Let l be the group size of the elliptic curve and k be its embedding degree. To achieve a comparable security strength of 1024-bit RSA, we should have lk to be larger than 1024, or at least close to 1024. Given the security level, a higher k results in a shorter group size. Therefore, choosing a high embedding degree for the elliptic curve in our scheme may result in not only a short ciphertext, but also an efficient scalar multiplications on each sensor. However, the embedding degree k of the elliptic curve can not be arbitrarily large. Choosing an appropriate embedding degree for the elliptic curve is actually another research area. According to the benchmark of Pairing-Based Crypto (PBC) library [25], elliptic curves with $l = 512$ and $k = 2$ results in the fastest bilinear pairing as compared to those with $k > 2$ for SS curves. The case is on the opposite for MNT curves. According to our testing of the PBC library on Linux platform with an Intel Pentium D 3.0GHz CPU, SS curves with $l = 512$ and $k = 2$ (type a curves in PBC) take about $6ms$ to execute

a pairing, while MNT curves with $l = 159$ and $k = 6$(type d curves in PBC) take about $14ms$ (Actually, on the user side of FDAC decryption time is linear to the number of pairings). Although both results are acceptable to users, MNT curves imply a much shorter ciphertext as well as key size to sensor nodes. More importantly, scalar multiplication over 512-bit curves may not be supported by low-end sensor nodes such as Tmote Sky because it consumes too much RAM. For these reasons, we believe MNT curves with high embedding degrees are suitable for FDAC.

In our implementation, the elliptic curve is a MNT curve over F_q with embedding degree of 6, where q is a 159-bit prime number. The curve has the form $y^2 = x^3 + ax + b$. Our implementation is based on the TinyECC library [26] with curve specific optimization disabled since the group size q is not a Mersenne prime. Our result shows that iMote2 consumes about $35ms$ to execute a scalar multiplication when working at 416MHz, $69ms$ at 208MHz, and $139ms$ at 104MHz. Tmote Sky consumes $4.1s$. For the 512-bit SS curve, iMote2 consumes $170ms$ at 416MHz, $341ms$ at 208MHz, and $682ms$ at 104MHz. Tmote Sky does not have enough RAM to support 512-bit SS curve.

3) Discussion: In FDAC we assume each phase to be the period that the sensor node transmits its data to the designated distributed storage node. To save communication cost, sensor nodes may store its data at an appropriate frequency. For example, in TTDD [20] it assumes that the sensor node sends one 64-byte data package per second. If we follow this assupmtion, we can see that FDAC is affordable to high-end sensor nodes such as iMote2 because it costs less than $36ms$ per phase. For low-end sensor nodes, however, FDAC may be too expensive in terms of time cost and energy consumption.

V. Conclusion

In this paper, we analyzed a novel yet important issue of fine-grained data access control for distributed storage in WSNs. To address the problem, we proposed a scheme called FDAC in which each sensor node is assigned a set of attributes, and each user is assigned an access structure which designates the access capability of the user. The sensor data is encrypted under the attributes such that only the users with the intended access structure are able to decrypt. As the access structure is extremely expressive, we are able to control data access precisely, and thus achieve fine-grained access control. Moreover, FDAC is able to provide security assurance such as resilience to user colluding and sensor compromising attacks as well as user revocability. Our experiment shows that the system overload in FDAC is reasonable in practical scenarios. An interesting future work of FDAC may be on its efficient implementation on WSNs with low-end sensor nodes.

Acknowledgments

This work was supported in part by the US National Science Foundation under grants CNS-0831628, CNS-0716306, and CNS-0831963.

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Networks Journal (Elsevier)*, vol. 2, no. 4, pp. 351–367, Oct. 2004.
[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–116, Aug. 2002.
[3] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. of IEEE*, Aug 2003.
[4] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," in *ACM SIGCOMM Workshop Data Comm. Latin America and the Caribbean*, Costa Rica, Apr. 2001.
[5] D. Estrin, D. Culler, and K. Pister, "Connecting the physical world with pervasive networks," *IEEE Percasive Computing*, Jan-Mar 2002.
[6] B. Thuraisingham, "Secure sensor information management and mining," *Signal Processing Magazine, IEEE*, vol. 21, no. 3, pp. 14–19, May 2004.
[7] A. Banerjee, A. Mitra, W. Najjar, D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopulos, "Rise co-s : High performance sensor storage and co-processing architecture," in *SECON'05*, Santa Clara, California, 2005.
[8] A. Mitra, A. Banerjee, W. Najjar, D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopulos, "High-performance low power sensor platforms featuring gigabyte scale storage," in *MobiQuitous'05*, San Diego, CA, 2005.
[9] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, "Capsule: An energy-optimized object storage system for memory-constrained sensor devices," in *ACM Sensys*, November 2006.
[10] D. Zeinalipour-Yazti, V. Kalogeraki, D. Gunopulos, A. Mitra, A. Banerjee, and W. Najjar, "Towards in-situ data storage in sensor databases," in *Proc. of 10th Panhellenic Conference on Informatics (PCI'05), LNCS 3746*, Volos, Greece, 2005, pp. 36–46.
[11] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Lhap: A lightweight hop-by-hop authentication protocol for ad-hoc networks," in *23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)*, Providence, Rhode Island, USA, May 2003.
[12] F. Ye, H. Luo, S. Lu, and L. Zhang, "Stastical en-route filtering of injected false data in sensor networks," in *IEEE INFOCOM'04*, Hong Kong, China, Mar. 2004.
[13] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Security & Privacy*, Oakland, CA, May 2003.
[14] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based security mechanisms in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 247–260, February 2006.
[15] N. Subramanian, C. Yang, and W. Zhang, "Securing distributed data storage and retrieval in sensor networks," 2007.
[16] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Advances in Crytology CRYPTO05*, 2005.
[17] L. Cheung, J. Cooley, R. Khazan, and C. Newport, "Collusion-resistant group key management using attribute-aased encryption," in *Cryptology ePrint Archive Report 2007/161*, 2007.
[18] A. Fiat and M. Noar, "Broadcast encryption," in *Advances in Crytology CRYPTO93*, 1993.
[19] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *CRYPTO*, 2001.
[20] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *ACM MOBICOM'02*, Atlanta, Georgia, Sep 2002, pp. 148–159.
[21] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM CCS*, 2006.
[22] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient," in *AFRICACRYPT'08*, Casablanca, Moracco, Jun. 2008.
[23] S. Yu, K. Ren, and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity," in *SecureComm'08*, Istanbul, Turkey, Sep. 2008.
[24] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks," in *IEEE INFOCOM'06*, Barcelona, Spain, Apr. 2006, pp. 1–12.
[25] PBC Library. http://crypto.stanford.edu/pbc/times.html.
[26] TinyECC Library. http://discovery.csc.ncsu.edu/software/TinyECC/index.html.