AGREE: exploiting energy harvesting to support data-centric access control in WSNs

Giuseppe Bianchi[a], Angelo T. Capossele[b], Chiara Petrioli[b], Dora Spenza[b]

[a]*Department of Electrical Engineering University of Rome "Tor Vergata", Rome, Italy*
[b]*Department of Computer Science University of Rome "Sapienza", Rome, Italy*

Email: giuseppe.bianchi@uniroma2.it, {capossele, petrioli, spenza}@di.uniroma1.it

Abstract

This work is motivated by a general question: can energy harvesting capabilities embedded in modern sensor nodes be exploited so as to support security mechanisms which otherwise would be too demanding and hardly viable? More specifically, in this work we focus on the support of extremely powerful, but complex, fine-grained data-centric access control mechanisms based on multi-authority Ciphertext Policy Attribute Based Encryption (CP-ABE). By integrating access control policies into the (encrypted) data, such mechanisms do not require any server-based access control infrastructure and are thus highly desirable in many wireless sensor network scenarios. However, as concretely shown by a proof-of-concept implementation first carried out in this paper on TelosB and MicaZ motes, computational complexity and energy toll of state-of-the-art multi-authority CP-ABE schemes are still critical. We thus show how to mitigate the relatively large energy consumption of the CP-ABE cryptographic operations by proposing AGREE (Access control for GREEn wireless sensor networks), a framework which exploits energy harvesting opportunities to pre-compute and cache suitably chosen CP-ABE-encrypted keys, so as to minimize the need to perform CP-ABE encryptions when no energy from harvesting is available. We assess the performance of AGREE by means of simulation and actual implementation, and by validating its operation with real-world energy-harvesting traces collected indoors by Telos B motes equipped with photovoltaic cells, as well as publicly available traces of radiant light energy. Our results show that complex security mechanisms may become significantly less demanding when implemented so as to take advantage of energy harvesting opportunities.

Keywords: Energy harvesting, Environmentally-powered sensor nodes, Data access control, Ciphertext Policy Attribute-Based Encryption

1. Introduction

Wireless sensor nodes with energy harvesting capabilities (EH-WSN) are motes that, in addition to traditional sensing and wireless communications capabilities, are able to extract energy from the surrounding environment and to convert it into usable electrical power. A EH-WSN platform generally includes one or more harvesters, which scavenge power from the environment, and an energy storage device, typically a rechargeable battery or a supercapacitor, which can store excess energy for later use. Energy harvesting is quickly emerging as a key technology to enable virtually perpetual operation of wireless sensor networks [1, 2], supplementing or even replacing traditional power sources, such as primary batteries, which fail to meet the lifetime requirements of long-term deployments of WSNs. Applications of EH-WSNs include, among others, health care and assisted living [3, 4], structural health monitoring [5, 6], terrestrial and aquatic environmental monitoring [7, 8], etc. In many of such scenarios, providing reliable mechanisms to duly control access to the collected data is of paramount importance.

Health care and assisted living applications, in particular, present a number of unique challenges. Access to sensitive data must be allowed only to a given set of privileged users, who can belong to different institutions (multi-authority) and whose identities are not necessarily known a priori. Moreover, different types of data (e.g., health data versus patient location versus environmental data) may be meant for different sets of users. Finally, the recipients of a given sensed information stream may further depend on the context, and change when the context does. For instance, consider a patient remotely monitored through sensing devices placed over her/his body. The gathered data (temperature, hearth rate, blood pressure, etc.) is generally not meant for a *specific doctor* (i.e., as an individual), but for someone having the *role* of doctor, whose identity may vary over time and may not even be known a-priori. Moreover, critical health conditions (as attested by anomalies in the gathered data) require prompt intervention. Such data should hence become suddenly accessible by emergency personnel not originally in charge of handling *normal* health parameters.

Data-centric Access Control

The problem of granting access privileges to given users is known as 'data access control', and it is traditionally addressed by dedicated server-based privilege management infrastructures. However, such infrastructures come along with several drawbacks in a scenario of pervasive deployment

of wireless sensor networks, including the significant management burden posed to the WSN in terms of coordination and signaling.

Such drawbacks may be overcome by using different approaches, such as the novel category of Attribute Based Encryption (ABE) [9] schemes, which permit to address the access control problem through a completely different *data-centric* perspective. ABE permits a recipient to decrypt a given ciphertext only if she/he satisfies a given access control policy. ABE has been designed in two flavors. In the original Key-Policy (KP-ABE) constructions [9, 10], an encrypted data is labeled with a set of descriptive attributes and the access control policies reside on the data recipients' terminals. An opposite approach is instead promoted by ABE schemes called Ciphertext-Policy (CP-ABE), originally proposed in [11]. With CP-ABE, access control policies (unlike attributes) are embedded in the ciphertext. Attributes are instead issued to users, possibly from multiple independent authorities (e.g., when using the construction in [12]). A user may decrypt the cyphertext only if the set of attributes she/he holds satisfies the access control policy embedded in the data. For instance, a monitored patient may freely decide that a given data may be accessed only by 'doctors AND nurses', or only by 'doctors OR personnel from a specific hospital'.

Embedding the access control policies inside the ciphertext, rather than having them enforced on external servers or policy enforcement points, makes CP-ABE schemes particularly well-suited for WSN scenarios. In fact, differently from KP-ABE, CP-ABE allows each sensor to dynamically and independently change the access control policies attached to the data , e.g., to promptly respond to a change of context or environmental conditions. This in turn gives full control to end users which can decide the access rules to their data and how they should evolve with context.

Our contribution

Despite the appeal of CP-ABE schemes, there is considerable skepticism on their viability over battery-powered and resource-constrained sensors, especially when considering multi-authority schemes whose computational complexity and overhead scales at best linearly with the number of attributes involved in a policy. Indeed, to the best of our knowledge, our paper is the first to document an implementation of a multi-authority CP-ABE scheme (actually, we are not even aware of previous works documenting "just" single-authority CP-ABE implementations over WSN motes - works [13, 14] in fact report single-authority KP-ABE implementations), and our results show that performance appear still far from being practical for battery-powered nodes.

Our work stems from the observation that the ever increasing emergence of energy harvesting technologies for sensor nodes leads to a radical rethinking of energy efficiency strategies. In the traditional scenario of battery-powered devices, the only and obvious strategy to prolong the lifetime of a WSN was to reduce energy consumption as much as possible. This translated into believing that energy demanding operations (such as those mandated by some security schemes or cryptographic routines) were not feasible for energy constrained embedded devices. Conversely, energy harvesters opportunistically draw energy from the environment[1]. The result is an alternation between periods in which energy must be sparely used, and situations in which there may even be an **excess of energy available**, energy which would be wasted unless used in the short term. Moreover, even if energy availability cannot be controlled, it can be predicted [15, 16, 17] to some extent, thus allowing the development of proactive energy management strategies. This opens up new opportunities: the question is no longer restricted to quantify how demanding an operation (say a costly CP-ABE encryption of a key using a given policy) is in terms of energy toll, but it extends to further understand *when* such an operation has to be performed and *whether*, and *to what extent* demanding computations can be pushed to periods where energy is harvested and is in excess.

This new area of *green wireless sensor network security*, i.e., how to exploit the opportunities provided by energy harvesting for revisiting WSN security schemes, has so far been overlooked. This paper takes a first step in the direction of re-thinking WSN security schemes so as to exploit the opportunities provided by energy harvesting. Specifically, we make the following contributions:

- We assess the feasibility of CP-ABE in WSNs via an actual implementation in TinyOS for Telos B and Mica2 platforms. Such implementation allowed us to determine its energy consumption, memory requirements and computational complexity and guided us through the development of specific optimizations aiming at reducing the large overhead of CP-ABE over resource-constrained motes. To the best of our knowledge, we are the first to fully implement a CP-ABE scheme (actually, the more complex multi-authority case) over sensor platforms.

[1] In health-care applications using wearable medical devices, potential sources of energy harvesting include indoor light energy, mechanical energy produced by movements, and heat transfer between the human body and the ambient.

- Around such multi-authority CP-ABE core, we have designed AGREE, an energy-harvesting-aware Access control framework for GREEn WSNs. AGREE mitigates the energy consumption of the CP-ABE scheme by pushing most of the costly encryption operations to energy harvesting periods, pre-computing and storing the CP-ABE encryption of as many keys as possible. Since the memory of the motes is clearly unable to hold all possible access control policies, AGREE implements a caching strategy designed to store information so as to minimize the need to invoke a CP-ABE operation before the next predicted energy harvesting phase occurs.

- We provide a simulation-based performance evaluation framework for EH-WSNs. In our experiments, we use traces of the availability of indoor light energy that we obtained by interfacing TelosB nodes with photovoltaic cells, collecting data for a week in the student office of the CS Department building of Sapienza University of Rome. We also validate our approach by using two additional datasets obtained from the EnHANTs (Energy Harvesting Active Networked Tags) project of Columbia University.

- We performed a comparative performance evaluation of AGREE and of two other caching strategies which do not leverage information about the harvesting process and the dynamics of the application. Our validation clearly shows that AGREE is able to efficiently operate based on the excess harvested energy and that it significantly outperforms other harvesting-unaware caching strategies.

The remainder of this paper is organized as follows. We discuss related work in Section 2. In Section 3 we review known results by giving an overview of the operation of CP-ABE schemes. We present our scheme, AGREE, in Section 4. In Section 5 we evaluate our proposed approach, discussing practical implementation challenges of CP-ABE schemes and presenting a simulation-based evaluation of AGREE. Finally, we present our conclusions in Section 6.

2. Related works

Application scenarios of EH-WSNs are ever increasing and for many of them providing reliable security support is a critical requirement. Despite extensive research has been devoted to devise security solutions specifically

tailored to WSNs [18, 19], such works generally target traditional battery-powered wireless sensor motes. Energy harvesting techniques, however, by providing virtually unlimited energy to nodes, change the way WSNs operate and the general underlying assumption that the energy reservoir of the network is finite and monotonically decreasing over time. This calls for new dedicated approaches that can leverage harvesting opportunities, but so far only a few works [20, 21] have addressed security topics in such context. In [20], Taddeo et al. proposed an optimization mechanism that allows a EH-WSN to change the communication security settings over time, based on the energy state of the network. Different types of packets, each having different priority level and security requirements, are handled, and a quality of service mechanism is introduced to favor high-priority packets when the harvesting energy intake is scarce. Pelissier et al. proposed in [21] a scheme that applies to stream ciphers, which allows energy harvesting systems to precompute and store keystream bytes, and to use them when the system energy availability is low. However, stream ciphers, being symmetric encryption algorithms, consume impressively less time and energy with respect to asymmetric cryptography schemes: indeed, the time and energy required to perform a single public-key operation can be the same as encrypting tens of megabits using symmetric encryption [22].

In this paper we show that smart caching and energy intake prediction can be combined to make computationally involved asymmetric cryptography schemes feasible in real wireless sensor networks with energy harvesting. In particular, in our paper we focus on data access control and CP-ABE schemes. The problem of data access control in WSNs, which is of paramount importance in health care and assisted living applications, has received notable attention by the research community, but, to the best of our knowledge, no solution for networks with harvesting capabilities has been proposed so far. Recent works targeting data access control in traditional WSNs are based on Attribute Based Encryption (ABE), a cryptographic primitive introduced by Sahai and Waters in [9] and later extended by [10, 11], which proposed Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE), respectively. The technical feasibility of KP-ABE techniques in wireless sensor networks have recently been demonstrated [13, 14]. In [13], Yu et al. presented a centralized fine-grained data access control scheme, based on KP-ABE, for distributed storage in wireless sensor networks, which has been specifically adapted to WSNs performance and security requirement. However, their solution only addresses single authority scenarios, in which compromising the single authority jeopardizes the security of the whole system. The framework proposed by Ruj et al. in [14]

partially solved this limit, but their solution can only support strict "AND" policies and it requires a pre-determined set of authorities.

CP-ABE schemes, instead, support multi-authority [12] and provide a framework for dynamic access control which well fits the application requirements of traditional WSN applications. However, since they suffer from a significant higher overhead than KP-ABE approaches, their viability over both traditional and energy harvesting wireless sensor networks is still to be proven. Making CP-ABE schemes applicable in real-life is the objective of this paper.

3. CP-ABE overview

Our work capitalizes on a decentralized CP-ABE scheme recently proposed by Lewko and Waters, referred to in what follows as LW. In the next subsections we give an overview of such scheme and of its functionalities. The interested reader is referred to [12] for formal proofs of the security of the scheme.

3.1. Preliminaries

The runtime operations of CP-ABE comprise two functionalities: *encryption*, performed by the sensor nodes in charge of gathering and delivering the sensed information, and *decryption*, performed by the data recipient which we non restrictively assume to be a back-end infrastructure device or an end user's terminal (i.e., not a sensor node). CP-ABE operation is asymmetric. Similarly to ordinary asymmetric encryption (e.g., RSA), a sensor node does not need to store any secret key. Rather, in the general context of multiple authorities, the information needed by the system are:

- An *access control policy*, namely a boolean predicate over a set of attributes, which specifies the set of users that are allowed to decrypt the data.
- A set of *attributes*, which are ordinary strings of text arbitrarily formatted.
- A set of *public keys*, one per each attribute, potentially released by different authorities. An encrypted data for a given attribute may be decrypted only by a user possessing a secret key associated to the attribute name and to the authority releasing the attribute.

3.2. Setup and decentralized attribute issuing

In CP-ABE schemes, involved parties agree on the following public parameters:

- Two multiplicative cyclic groups, G and G_T, of same prime order N, chosen such that the discrete logarithm problem is hard to solve on both G and G_T;
- a generator g for the group G;
- a global hash function $H : \{0,1\}^* \rightarrow G$ that maps arbitrary strings into elements of the group G;
- a bilinear map $e : G \times G \rightarrow G_T$, satisfying the following properties: bilinearity, non degeneracy and computability [11].

CP-ABE supports multi-authority system, in which any party can be an independent Attribute Authority (AA) by creating and publishing a verification key coupled with a list of attributes it will manage. For each issued attribute i, the AA chooses two random exponents $\alpha_i, y_i, \in \mathbb{Z}_{\mathbb{N}}$, and publishes $PK_i = \{e(g,g)^{\alpha_i}, g^{y_i}\}$ as its public key. We recall that attributes are *permissions* to access encrypted data, and as such are issued not to encrypting sensor nodes, but to users. To identify different users, a global identity GID_u (a text string, e.g., the user's social security number) is associated to each user u.

3.3. Message encryption

Messages are encrypted along with an access control policy over a set of attributes. Access structures are described through Linear Secret Sharing Scheme (LSSS) matrices [23]. To encrypt a message D, the first step consists of modeling the applicable access control policy in terms of an $a \times l$ LSSS matrix LS, where a is the number of attributes involved in the policy and l is a parameter depending on the considered policy[2].

We define $\rho(x)$ as a function mapping rows x of LS to the corresponding attribute. We also also recall that, by construction, the encrypting node knows the public key of each attribute $\rho(x)$.

The encryption algorithm chooses a random secret $s \in \mathbb{Z}_{\mathbb{N}}$ and a random vector $v \in \mathbb{Z}^l_{\mathbb{N}} =< s, v_2, v_3, \cdots, v_l >$, having the secret s as its first entry

[2]The reader can refer to Appendix G in [12] for a practical procedure to convert an arbitrary boolean policy into an LSSS matrix.

and random values as the subsequent entries. It calculates $\lambda_x = LS_x \cdot v$, where LS_x is a row of LS. Similarly, it chooses a random vector $w \in \mathbb{Z}_{\mathbb{N}}^l =< 0, w_2, w_3, \cdots, w_l >$ with 0 as the first entry and it defines $\omega_x = LS_x \cdot w$. For each row LS_x of LS, it chooses a random $r_x \in \mathbb{Z}_{\mathbb{N}}$. It finally encrypts the message D computing the following parameters:

$$\begin{aligned} C_0 &= De(g,g)^s \\ C_{1,x} &= e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\rho(x)} r_x} \; \forall x \\ C_{2,x} &= g^{r_x} \; \forall x \\ C_{3,x} &= g^{y_{\rho(x)} r_x} g^{\omega_x} \; \forall x \end{aligned} \tag{1}$$

3.4. Message decryption

To decrypt a message a user u needs to possess a secret keys K_{x,GID_u} for each attribute x belonging to a set of attributes which satisfy the access control policy embedded in the ciphertext. This is verified by checking whether there exists a subset X of attributes owned by the user, such that a linear combination of the relevant rows in the LSSS matrix LS yields the vector $(1, 0, \ldots, 0)$. If this condition is verified, the user may decrypt the message D. For each $x \in X$, the user u computes:

$$\frac{C_{1,x} e(H(GID_u), C_{3,x})}{e(K_{\rho(x),GID_u}, C_{2,x})} = e(g,g)^{\lambda_x} e(H(GID_u), g)^{\omega_x} \tag{2}$$

Then user u chooses the constants $c_x \in \mathbb{Z}_{\mathbb{N}}$, such that $\sum_{x \in X} c_x LS_x = (1, 0, \ldots, 0)$ and computes:

$$\prod_{x \in X} (e(g,g)^{\lambda_x} e(H(GID), g)^{\omega_x})^{c_x} = e(g,g)^s \tag{3}$$

Note that $\lambda_x = LS_x \cdot v$ and $\omega_x = LS_x \cdot w$, where $v \cdot (1, 0, \ldots, 0) = s$ and $w \cdot (1, 0, \ldots, 0) = 0$.

Finally the user can obtain the original message D as:

$$D = C_0 / e(g,g)^s. \tag{4}$$

4. AGREE

In this section, we present our scheme, AGREE. In Section 4.1 we discuss the specific optimizations we devised to reduce the large overhead of CP-ABE in WSNs. In Section 4.2 we present a technique to react to critical

situations and changes of context. A mechanism that allows nodes to leverage energy harvesting opportunities to pre-compute policies is presented in Section 4.3. Finally, we introduce in Section 4.4 a caching strategy to pre-compute the most likely set of policies, as quantified by a Markov model of the sensor node's application state evolution.

4.1. WSN specific optimizations

The main source of overhead introduced by CP-ABE schemes is due to message encryption, as several scalar multiplications must be performed to compute the parameters $C_0, C_{1,x}, C_{2,x}$ and $C_{3,x}$ described in Section 3.3, Equation (1). The first specific optimization we propose to adapt CP-ABE to resource-constrained devices is to have nodes ascribing the access policy to a *session key*, SSK, instead that to the data itself, as in traditional CP-ABE schemes. More in details, a session key SSK is generated and encrypted using CP-ABE. Each sensed data D is encrypted by means of a symmetric-key algorithm, such as AES [24], using SSK as secret key. Upon request for sensor data, the mote responds with both the encrypted session key SSK and the ciphertext of the sensed data D. If the user is an intended receiver, he will be able to decrypt the session key and to derive the data encryption key. Such optimization does not affect the security of the CP-ABE scheme, because, as explained in Section 3.4, to decrypt the session key the user should be an intended receiver, i.e., she/he would need to possess a secret key K_{x,GID_u} for each attribute x belonging to a set of attributes that satisfy the access control policy embedded in SSK. The rationale behind such optimization is that encrypting or decrypting data with a symmetric encryption algorithm, such as AES, is much more efficient than directly using ABE, which is several orders of magnitudes more resource demanding than symmetric encryption. Moreover, by using this approach, sensor nodes can pre-compute and store the parameters $C_0, C_{1,x}, C_{2,x}$ and $C_{3,x}$ when they have high energy and use them when the access policy changes or when a session key is refreshed.

As for further optimizations, the choice of the type of elliptic curve is important because it directly affects the performance of operations such as scalar multiplication and pairing. A generic pairing function is defined as $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two distinguished subgroups of order p in which the Elliptic Curve Decisional Diffie-Hellman problem (ECDDH) is hard to solve [25]. CP-ABE requires a prime order group with a symmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, which only exists on supersingular elliptic curves. In order to reduce ciphertext size, we choose a supersingular curve over a binary field, having the form $y^2 + y = x^3 + x$ with an embedding

degree of 4. This kind of curve is well suited for WSNs environments and it can be implemented very efficiently in terms of memory and processing requirements, when compared with implementations on ordinary curves [26].

4.2. Access policies updates

In many WSNs scenarios, the access provided to the data is highly dependent to the current context. For this reason, it is very important to consider the occurrence of critical events in the system and to react to them by providing fast mechanisms to change the access policies when needed. Critical events must be timely handled and a dynamic access control system is essential to this end. Any access provided to roles in response to an emergency is temporary and is rescinded after a specific amount of time, which depends on the specific emergency happened. For example, in a health care application scenario, a critical event may be the patient suffering from a heart attack while being at home. In such context, sensitive data that are normally accessible only to her/his doctor should also be made available to the emergency paramedical team as quickly as possible. To handle such situations, the basic idea is to have a hierarchy of access policies. Each level of the hierarchy corresponds to a level of criticality. Let L be the number of levels that the system can support. If the system is using the access policy level n, it will ignore all the access policies of level $n+1, \ldots, L$. For example, consider this set of access policies associated with an ECG sensor of a patient:

1. p_1 = *Doctor AND Patient's consent*
2. p_2 = *(Doctor OR Nurse OR Paramedic) AND (Patient's consent)*
3. p_3 = *Doctor OR Nurse OR Paramedic*

Figure 1 shows the description of the policy p_2 as a tree representing the corresponding boolean function. Initially, the sensor node adopts the policy p_1. Policies p_2 and p_3 are not used, as their hierarchical level is higher than the current criticality level of the system, but the sensor node may pre-compute them for future use, if it has enough energy and cache memory available. Whenever a critical event occurs, according to the context in which the patient is located and to the importance of the event, the sensor node will adopt a policy of higher level, thus ensuring the timely adjustment of data access. Changing the access policy of the data implies updating the session key SSK by running the encryption algorithm described in Section 3.3. More in detail, if a new policy p' is adopted, the following operations should be performed by the node:

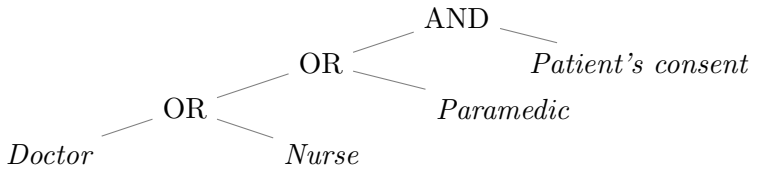


Figure 1: Description of the policy p_2 as a tree representing the corresponding boolean function.

1. generate a new session key $SSK_{p'}$;
2. convert the access policy p' into an LSSS matrix;
3. apply CP-ABE to encrypt $SSK_{p'}$ with the access policy p', by computing the new parameters C_0, $C_{1,x}$, $C_{2,x}$ and $C_{3,x}$;
4. locally store the parameters of the new policy p': ($SSK_{p'}$, C_0, $C_{1,x}$, $C_{2,x}$, $C_{3,x}$);
5. send the new encrypted session key to the users.

Such operations, however, are quite slow and expensive in terms of energy consumption and pose a significant burden to resource and energy constrained WSNs. In the next Section 4.3, we propose a mechanism to mitigate such energy consumption by leveraging characteristics of EH-WSNs.

4.3. Pre-computation of policies

In energy harvesting enabled WSNs, available energy varies over time in a non monotonic yet partially predictable [15] fashion and there might even be situations in which we have an excess of energy available, which is wasted unless used in the short term. In fact, since the storage device has a finite size, some energy may be lost if the energy buffer is full while the node is harvesting energy. Figure 2 shows an example of such situation. As described in Section 5.2, we interfaced Telos B motes with photovoltaic cell, we deployed them indoors, and collected data about the energy that the nodes were able to harvest from artificial light, i.e., by ceiling and table lamps, and from solar light entering the room from the windows. Figure 2 shows the energy harvested during two different days with this setup and the energy surplus that occurred during this time frame. On the first day, due to low light energy coming from the windows and from artificial illumination, the node does not harvest enough energy to fully recharge its capacitor. The surplus energy is thus zero. During the second day, on the contrary, the node supercapacitor is recharged up to its maximum capacity at around 12:00 PM. After then, the energy harvested should be used immediately by

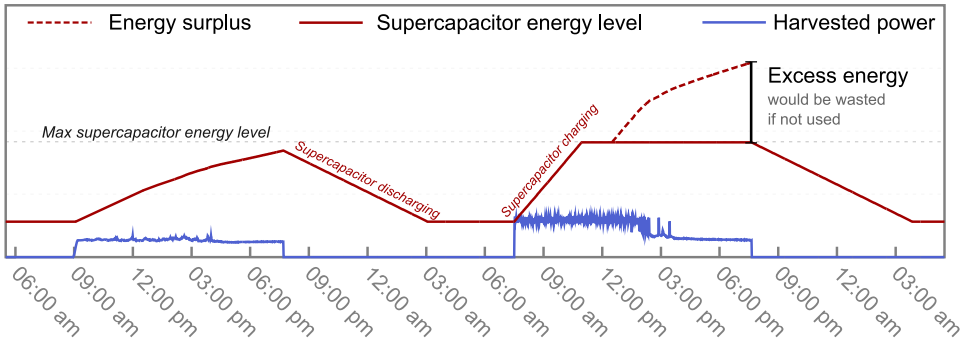


Figure 2: Traces of the light energy harvested indoors over two days. There is an energy surplus between 12:00 PM and 7:30 PM on the second day: since the supercapacitor is full, the energy harvested by the node during such period of time should be used immediately or it would be wasted.

the node or it is going to be wasted. Short after 7:30 PM, the harvesting power decreases and the supercapacitor starts discharging.

To avoid wasting energy, policies pre-computations are performed when there is an energy surplus, i.e., when the storage device is fully charged and the harvesting power is higher than the power consumption of the node. By exploiting such energy surplus, the number of CP-ABE operations that must be performed by the nodes when the harvested power is low can be minimized, thus allowing to save energy and to exploit recharge opportunities more effectively. The drawback of this approach, however, is the fact that pre-computed policies must be stored in the typically limited RAM memory of the nodes.

4.4. Caching strategy

Since the memory available on a sensor node is clearly unable to store all the possible sets of access control policies, we propose in this section a caching strategy to pre-compute and store the set of policies that are the most likely to be useful at runtime. Table 1 summarizes the notations used in this section. The system is characterized by a finite set of application states, $S = \{S_1, \dots S_{|S|}\}$. To each state of the system, S_i, a set of policies, $P_i = \{p_i^1, p_i^2, \dots p_i^{n_i}\}$ is associated, which define the policies that the system needs in such state. A stored policies is a tuple of five elements: ($SSK^{p_i^k}$, C_0, $C_{1,x}$, $C_{2,x}$, $C_{3,x}$). We denote the size of each policy $p_i^k \in P_i$ as $l(p_i^k)$.

When the system is in the state S_i and a given amount of excess harvested energy is available, such energy surplus is used to pre-compute and cache policies that may be useful to the system in the future. The number

Table 1: Table of notation.

Symbol	Explanation
S	set of system states
P_i	set of policies associated to the state S_i
R	amount of cache memory available in each state
$l(p_i^k)$	size of the k^{th} policy of the state S_i
M	Markov chain modeling system states and transitions probabilities
T	matrix of transition probabilities
O_x	set of policies stored in the cache
S_{O_x}	set of system states which are incomplete when using the set of stored policies O_x
M_{O_x}	absorbing Markov chain obtained from M based on O_x
T_{O_x}	matrix of the transition probability of M_{O_x}
AB_{O_x}	characteristic vector of the absorbing states of M_{O_x}
NH	number of timeslots before the next harvesting event will occur
$Pr_{miss}(NH)$	probability that a cache miss will occur within the N timeslots before the next harvesting event

of policies that may be stored in the cache depends on the available cache size, R, and on the size of the pre-computed policies, $l(p_i^k)$. For simplicity, we assume that in each state S_i there is enough cache memory, R, to store all the policies needed in the current state[3] , i.e.,

$$R \geq \sum_{p_i^k \in P_i} l(p_i^k), \qquad \forall S_i \in S.$$

However, in general the cache memory would not be large enough to store the full set of access control policies associated to all possible application states, thus leading to potential cache misses when the state of the system changes, i.e., when there is a transition from state S_i to state S_j.

In fact, a cache miss occurs if a policy p_j^k is not available in the cache when needed and must be computed on-the-fly. Computing a given policy has an energy cost that depends on its size (Table 2).

The goal of our optimization is to minimize the number of cache misses, by wisely selecting in each state the policies that should be pre-computed.

[3]This is indeed the case in realistic scenarios given that only a few policies are expected per application state, each combining few different attributes.

Our strategy is based on the knowledge of the application dynamics, which can be obtained at design time. The optimization is performed by the sink and then disseminated to the nodes in the network. To reduce memory and communication overhead, caching strategies are stored in a compressed form, by using a bitmap representation of the policies that should be precomputed in each application state. The caching strategy is updated by the sink and retransmitted to the node whenever a major change in either the application or in the energy source dynamics occurs.

We formalize this problem using a discrete-time Markov chain, M, with $|S|$ states. The associated transition matrix is T. We recall that the set S models the application states, as defined during the design phase, while T indicates the probability t_{ij} to transit to a state S_j, given that the system is currently in the state S_i. To minimize the number of cache misses, we employ the following approach:

- We explore all the possible combinations of policies stored in the cache. Each of such combinations is a set O_x. We define a state S_i as *incomplete* if there is at least a policy, p_i^k, which is needed in state S_i but is not included in the set O_x of stored policies. Given the set of policies O_x, we define by $S_{O_x} = \{S_i, \ldots, S_j\}$ the set of incomplete states:

$$S_{O_x} = \{S_i \text{ s.t. } \exists p_i^k \in P_i \text{ and } p_i^k \notin O_x\}$$

- For each set of policies O_x, we define a new absorbing Markov chain, M_{O_x}, as the Markov chain obtained from M by setting to zero the output transition probability of each incomplete state, i.e., by making each incomplete state in S_{O_x} an absorbing state. We name the new transition probability matrix obtained in this way as T_{O_x}. The characteristic vector A_{O_x} is also defined. It associates a value equal to 1 to each absorbing state and a value of 0 to each non-absorbing state, i.e., $AB_{O_x}(i) = 1$ if $S_i \in S_{O_x}$, $AB_{O_x}(i) = 0$ otherwise.

- Time is discretized into timeslots of equal length. We denote with NH the number of timeslots before the next harvesting event will occur. We then compute the state probability as $pr(NH) = T_{O_x}^N H \times [0, 0, \ldots, 1, 0, 0, \ldots]$. The second factor in the matrix multiplication is a unitary vector of size S that indicates that the system starts in a given state S_k. The probability that a cache miss will occur within the NH timeslots before the next harvesting event can thus be computed as:

$$Pr_{miss}(NH) = pr(NH) \times AB_{O_x}$$

Such formalization allows to minimize the cache miss probability by selecting the best set O_x of policies that must be stored in the memory. If two or more sets of policies, $O_1, O_2, \ldots O_k$, have the same, minimum cache miss probability, we take into account the fact that, whenever a cache miss occurs, computing the missing policy has an energetic cost proportional to the size of the policy. Thus, in such case, we select the set O_i which minimizes both the cache miss probability and the energetic cost of computing the missing policies whenever a cache miss occurs.

It is worth noting that the number of possible Markov chains M_{O_x} is significantly smaller than the number of possible policies combinations, since different policies set may generate the same absorbing chain. For this reason, it is feasible to explore the possible combinations of policies stored in the cache and defining the associated chain. Additionally, some optimizations may be employed when computing each O_x sets. For instance, it is reasonable to consider only sets of policies that are maximal, i.e., such that no policy may be added to the set without exceeding the memory constraint R. Moreover, if the system starts in the state S_i, the O_x sets which do not contain all the policies in P_i (i.e., those needed in state S_i) can be pruned from the exploration.

5. Performance evaluation

5.1. Experimental results: energy cost of CP-ABE encryption

In our implementation, we have specifically focused on two families of nodes: the Telos B [27] and Mica2 [28] motes. Telos B features an 8MHz MSP430 micro-controller, a 16b RISC processor, 10 kB of RAM, 48 kB of program memory (ROM), 1024 kB of external flash, and the Chipcon CC2420 IEEE 802.15.4 compliant transceiver. The Mica2 motes are equipped with the 4MHz Atmel ATmega128L 8b micro-controller, 4 kB of RAM, 128 kB of ROM, 512 kB of external flash and the Chipcon CC1000 low-power wireless transceiver. We implemented a nesC library supporting CP-ABE in TinyOS 2.x for both Telos B and Mica2 motes. Our library is based on Relic[4], an open source cryptographic meta-toolkit with emphasis on efficiency and flexibility. As recommended by NIST [29], we adopt a security level of 80-bit using a binary field $\mathbb{F}_{2^{271}}$. To encrypt data we use AES encryption, performed in hardware on the Telos B mote, which provides AES in Counter mode with CBC-MAC (CCM) within the CC2420 chip.

[4] http://code.google.com/p/relic-toolkit

Table 2: Energy consumption for policy computation (mJ), for both Telos B and Mica2 platforms. The energy cost of an encryption operation depends on the number of attributes of the access policy and on the number of bytes sent to transmit the encrypted key.

Attributes	**Policy Length [bytes]**	**Tx [bytes]**	**Scalar Multi.**	**Energy (Telos B)**	**Energy (Mica2)**
2	242	$242 + \|key\|$	7	80.7	451.6
3	349	$349 + \|key\|$	10	115.3	645.1
⋮	⋮	⋮	⋮	⋮	⋮
10	1154	$1154 + \|key\|$	31	357.6	1999.9
11	1277	$1277 + \|key\|$	34	392.2	2193.4

Such combined mode supports integrity, authentication and confidentiality. For Mica2 motes, we use a software implementation of AES[5] in CBC mode.

Pairing and scalar multiplication are the most expensive operations among the ones performed, so we focus on them in our evaluation. While encrypting the data, one pairing operation is performed to calculate $e(g, g)$. For each attribute a, three scalar multiplications must be performed to compute $C_{1,x}, C_{2,x}$ and $C_{3,x}$. Since decryption operations are not performed by sensor nodes, but only by the final users who receive the encrypted data, we do not account for them in our evaluation. Finally, the communication overhead of transmitting an encrypted key is $|SSK| + a^2 + log|G_T| + a(log|G_T| + 2log|G|)$ bytes, where $|SSK|$ is the size of the secret session key used to encrypt the data with a symmetric encryption algorithm. At most a^2 bytes are needed for the matrix LS, and $a(|G_T| + 2|G|) + |G_T|$ bytes to transfer C_0, $C_{1,x}$, $C_{2,x}$ and $C_{3,x}$. Table 2 shows the energy cost of an encryption operation, depending on the number of attributes of the access policy and the amount of information to be sent to transmit the encrypted key, for both Telos B and Mica2 platforms. Performing one pairing operation and one scalar multiplication takes 1.29s and 1.73s for the Telos B mote and 1.9s and 2.24s for the Mica2 mote. Supposing an access policy is composed of 5 attributes, 16 scalar multiplication operations have to be performed. Assuming, for Telos B motes, an operating current of 1.8mA and an operating voltage of 3V, those operations cost 184.6mJ. Mica2 motes work with the same voltage, but their operating current is 8mA, so for the same operations the energy cost is 1032mJ. For a key size of 128 bit, the size of the information to be

[5] http://byte-oriented-aes.googlecode.com

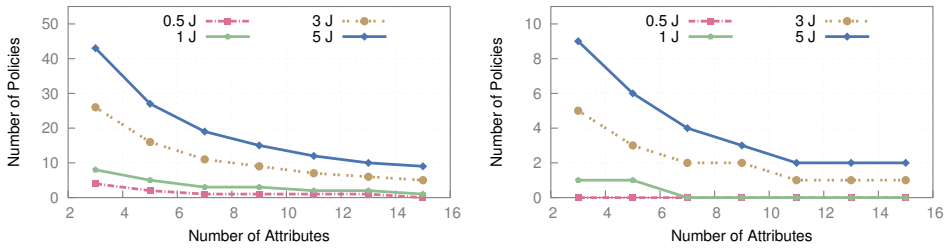


Figure 3: Number of policies which can be pre-computed per day, as a function of the number of attributes and the average energy surplus per day for (a) Telos B and (b) Mica2 motes.

sent is $569 + 16$ byte. Figure 3 shows the number of policies that can be pre-computed per day by using excess harvested energy, depending on their number of attributes and on the average energy surplus experienced per day, for both Telos B and Mica2 motes.

5.2. Energy model and real-life energy harvesting traces

In our experiments, we consider a Telos B mote powered by a hybrid harvesting systems consisting of a photovoltaic (PV) cell, a supercapacitor and a non rechargeable battery, which provides a backup energy source. PV cells are common sources of energy harvesting and, being unobtrusive, have been considered before in wearable systems. For example, in [3] Leonov et al. demonstrated an electroencephalography system and an electrocardiography system in a shirt, powered by photovoltaic cells and a thermoelectric generator. The supercapacitor we used for energy storage is a $25F$ Panasonic Gold capacitor [30], which can nominally hold around 90J of charge and has a round-trip (charging and discharging) efficiency of $90 + \%$ [31]. The leakage experienced by the supercapacitor is modeled as in [32], i.e., by using a piecewise linear approximation of the empirical leakage pattern experienced by the supercapacitor we have experimentally validated. We focus on a remote vital signs monitoring scenario, in which the heart rate and the blood oxygenation level of a patient are monitored through a wearable pulse oximeter, such as [33]. Pulse and oxygenation values are measured at 60 second intervals and such measurements requires up to 8 seconds [34], thus leading to a duty cycle of $\approx 13\%$. Measured data are delivered immediately after reading and a low-power, state-of-the-art communication protocol, such as [35, 36], is used for data delivery. The rest of the time the MCU is in idle (sleep) mode, thus leading to an average power consumption

of 2.02 mW. In such setting a node can run for almost three hours using only the energy stored in its supercapacitor (assuming it is full).

For our first set of experiments, a dataset of real-life indoor light traces has been used. The dataset was obtained through a testbed of ten Telos B motes equipped with a 0.5W PV cell, deployed indoors for a week in the student office of the CS Department building of Sapienza University of Rome. Nodes were able to harvest energy from artificial light generated by ceiling and table lamps and from solar light entering the room from the windows. A dedicated TinyOS application was developed to periodically track the amount of energy harvested by the cell.

The second type of energy traces we used is a database of indoor radiant light measurements collected in several office buildings in New York City within the EnHANTs (Energy Harvesting Active Networked Tags) project of Columbia University [37]. In particular, we used data from Setup C (departmental conference room) and Setup F (student office), obtaining from the radiant energy measurements the corresponding power harvestable by a photovoltaic cell of size of $7x5$ cm^2 with efficiency of 15%.

5.3. AGREE simulation results

In this section we describe the results of a simulation-based performance evaluation of AGREE, conducted by using a custom-built simulator developed in C and Python.

Our setup is as follows: We consider a system with a number of application states, $|S|$, ranging from 3 to 9. The transition probability between each couple of states, denoted by t_{ij}, is randomly generated as to represents different application scenarios. The corresponding Markov's chain M is defined based on the set of states S and the transition matrix T. To each state of the chain, a random number of policies (between 1 and 6) is associated. The size of such policies varies from 242 to 1277 bytes, depending on the number of their attributes (Table 2). In our experiments, we model a scenario where 7kB of the Telos B mote RAM are allocated to storing cache policies, while the rest of the memory is reserved for the application.

To validate our proposed caching strategy, we compared three different versions of the protocol: AGREE, which is the complete solution described in Section 4.4, and *Current* and *Current + Random*, two variants of AGREE that do not leverage information about the harvesting process and the dynamics of the application. More in details, in each system state S_i, nodes using the *Current* strategy pre-compute and cache only the policies P_i needed in the current state, while nodes using the *Current + Random* strategy pre-compute and cache, in addition to the policies needed in the current

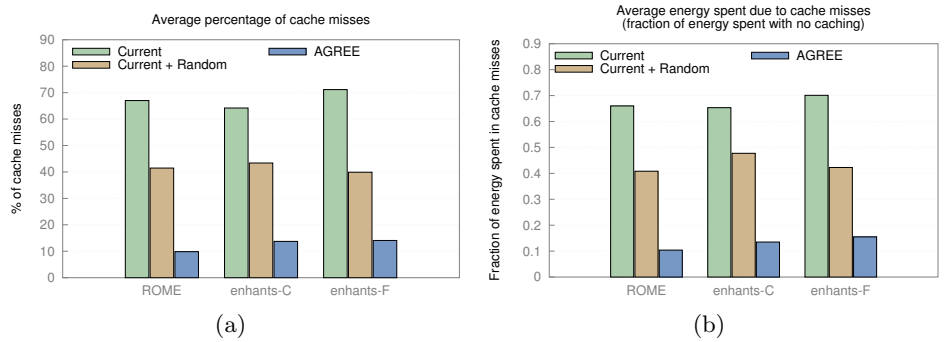


Figure 4: Performance of caching strategies *Current*, *Current + Random* and AGREE in term of (a) average percentage of cache misses and (b) average energy spent to compute policies on-the-fly due to cache misses, as a fraction of the energy spent when no caching is performed.

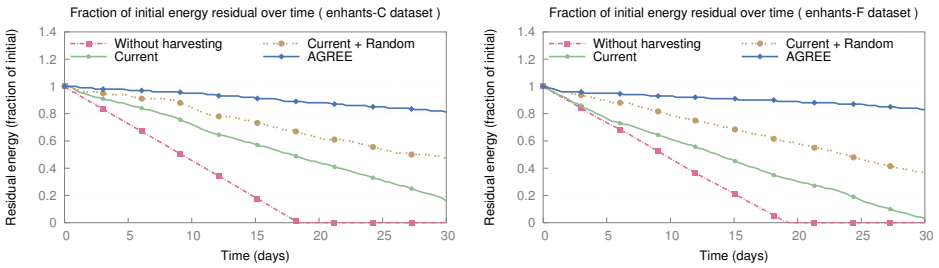
state, also as many other policies as possible, randomly chosen among those needed in neighbors states, until the RAM memory of the node is full.

To perform a comparative performance evaluation of the three approaches, we randomly choose a state S_i as the initial state of the system and follow the evolution of the chain for a week. We evaluate the performance of the different strategies with respect to the following metrics:

1. average percentage of cache misses (Figure 4(a));
2. energy spent by the nodes due to cache misses (Figure 4(b)).

Figure 4 shows the results of such performance evaluation for the three different light energy datasets we consider. Each data point is obtained by averaging the results over 10 runs. The fact that our approach is able to successfully pre-compute the most likely set of policies is confirmed by Fig. 4(a), which shows that AGREE leads to the smallest number of cache misses with respect to the other strategies. Specifically, the first comparison strategy, *Current*, obtains a number of cache misses that, depending on the considered energy harvesting dataset, is between 4.7 and 6.8 times higher than that of AGREE, while the strategy *Current + Random* obtains a number of cache misses that is between 2.8 and 4.2 higher than that of AGREE.

Fig. 4(b) shows the total energy spent by nodes due to cache misses over a week of simulation, as a fraction of the total energy spent over the same time period when no caching is performed. Results in Fig. 4(b) confirm that taking into account the dynamics of both the harvesting source and the system works well, as the total energy spent by AGREE due to cache



(a) (b)

Figure 5: Fraction of initial energy residual over time for a system without energy capabilities and performing no caching and for caching strategies *Current*, *Current + Random* and AGREE computed over one month using: (a) enhants-C and (b) enhants-F indoor light traces datasets.

misses is approximately one fifth of the first comparison strategy, *Current*, and less than one third of *Current + Random*.

Figure 5 confirms that such energy saving has a significant impact on the overall lifetime of the system. To study the long-term behavior of our application, we used the same setup as before, but followed the evolution of the system for a longer period (a month instead of a week) using enhants-C and enhants-F indoor light traces datasets to simulate the harvesting process. Fig.5(a) and Fig.5(b) show the average residual energy over time, as a fraction of the initial battery energy of the nodes, for the caching strategies *Current*, *Current + Random* and AGREE, and for a system without energy capabilities and performing no caching at all.

The fraction of initial energy remaining over time is a significant metric because, although nodes may recharge through energy harvesting, a mote with no residual battery will suffer from fluctuations of the environmental source. In fact, energy stored in the supercapacitor would allow the node to run in normal operation (i.e., monitoring data but not refreshing policies) for less than three hours. Such energy reservoir would not be enough to continue the data collection during a whole night, thus leading to periods of inactivity that significantly degrade the end user perceived performance.

The fact that AGREE allows to spend significant less battery energy with respect to other caching strategies is confirmed by Fig.5(a), showing simulation results for the enhants-C dataset. As can be seen from the figure, after one month of operation the average residual battery energy of nodes running AGREE is still 82% of their initial energy, while motes running *Current* and *Current + Random* retain only 16% and 47% of their battery energy, respectively. Results are similar for the second harvesting dataset

we considered, enhants-F: The average residual battery after one months is 85% of the initial energy when nodes run AGREE, while only 3% (37%) of the nodes initial energy is available when nodes run *Current* and *Current + Random*, respectively. Finally, for both datasets, a system without energy harvesting capabilities and performing no caching at all depletes its battery in less than 20 days.

6. Conclusion

In this paper, we presented AGREE, a context-aware decentralized data access control for EH-WSNs. Our scheme is based on CP-ABE, supports multi-authority and allows to dynamically change access policies based on context dependent user settings. AGREE is developed for WSN scenarios. We have proposed several optimizations for dealing with resource and energy constrained embedded systems. We have implemented the basic schema on Telos B and Mica2 motes and experimentally evaluated our proposed solution. A simulation-based performance evaluation of AGREE confirmed that our caching mechanism is able to efficiently operate based on the excess harvested energy and that it significantly outperforms other caching strategies which do not leverage information about the harvesting process and the dynamics of the application. Our evaluation show that, in spite of literature trends, complex cryptographic primitives are feasible in realistic EH-WSNs scenarios.

Acknowledgment

Dora Spenza is a recipient of the Google Europe Fellowship in Wireless Networking, and this research is supported in part by this Google Fellowship. This research was also sponsored in part by the FP7 project GENESI (GrEen sensor NEtworks for Structural monItoring), by the ARTEMIS project #1000128 CHIRON (Cyclic and person-centric Health management: Integrated appRoach for hOme, mobile and clinical eNvironments) and by the PRIN project TENACE.

References

[1] S. Basagni, M. Y. Naderi, C. Petrioli, D. Spenza, Wireless sensor networks with energy harvesting, in: Mobile Ad Hoc Networking: The Cutting Edge Directions, John Wiley and Sons, Inc., Hoboken, NJ, 2013.

[2] S. Sudevalayam, P. Kulkarni, Energy harvesting sensor nodes: Survey and implications, IEEE Communications Surveys & Tutorials, vol. 13 (3), 2011, pp. 443–461.

[3] V. Leonov, T. Torfs, R. Vullers, C. Van Hoof, Hybrid thermoelectric-photovoltaic generators in wireless electroencephalography diadem and electrocardiography shirt, Journal of Electronic Materials, vol. 39 (9), 2010, pp. 1674–1680.

[4] T. Torfs, V. Leonov, C. Van Hoof, B. Gyselinckx, Body-heat powered autonomous pulse oximeter, in: Proceedings of IEEE Sensors 2006, Daegu, Korea, 2006, pp. 427–430.

[5] B. O'Flynn, E. Popovici, D. Boyle, M. Magno, D. Brunelli, C. Petrioli, GENESI: Wireless sensor networks for structural monitoring, in: Proceedings of IEEE IMAPS - CPMT, Gdansk-Sobieszewo, Poland, 2011.

[6] D. Dondi, A. Di Pompeo, C. Tenti, T. Rosing, Shimmer: A wireless harvesting embedded system for active ultrasonic structural health monitoring, in: Proceedings of IEEE Sensors 2010, Waikoloa, HI, USA, 2010, pp. 2325–2328.

[7] P. Corke, P. Valencia, P. Sikka, T. Wark, L. Overs, Long-duration solar-powered wireless sensor networks, in: Proceedings of ACM EmNets '07, Cork, Ireland, 2007, pp. 33–37.

[8] C. Alippi, R. Camplani, C. Galperti, M. Roveri, A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring, IEEE Sensors Journal, vol. 11 (1), 2011, pp. 45–55.

[9] A. Sahai, B. Waters, Fuzzy identity based encryption, in: Proceedings of IACR EUROCRYPT 2005, Aarhus, Denmark, 2005, pp. 457–473.

[10] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of ACM CCS 2006, Alexandria, Virginia, USA, 2006, pp. 89–98.

[11] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: Proceedings of IEEE SP 2007, Oakland, California, USA, 2007, pp. 321–334.

[12] A. Lewko, B. Waters, Decentralizing attribute-based encryption, in: Proceedings of ACR EUROCRYPT 2011, Tallinn, Estonia, 2011, pp. 568–588.

[13] S. Yu, K. Ren, W. Lou, Fdac: Toward fine-grained distributed data access control in wireless sensor networks, IEEE Transactions on Parallel

and Distributed Systems, vol. 22 (4), 2011, pp. 673–686.

[14] S. Ruj, A. Nayak, I. Stojmenovic, Distributed fine-grained access control in wireless sensor networks, in: Proceedings of IEEE IPDPS 2011, Anchorage, Alaska, USA, 2011, pp. 352–362.

[15] A. Cammarano, C. Petrioli, D. Spenza, Pro-Energy: a novel energy prediction model for solar and wind energy harvesting Wireless Sensor Networks, in: Proceedings of the 9th IEEE International Conference on Mobile Ad hoc and Sensor Systems, IEEE MASS 2012, Las Vegas, Nevada, 2012, pp. 75 – 83.

[16] J. Piorno, C. Bergonzini, D. Atienza, T. Rosing, Prediction and management in energy harvested wireless sensor nodes, in: Proceedings of Wireless VITAE 2009, Aalborg, Denmark, 2009, pp. 6–10.

[17] A. Kansal, J. Hsu, S. Zahedi, M. B. Srivastava, Power management in energy harvesting sensor networks, ACM Transaction Embedded Computer System, vol. 6 (4), 2007, pp. 32–es.

[18] Y. Ren, V. Oleshchuk, F. Li, X. Ge, Security in mobile wireless sensor networks – a survey, Journal of Communications, vol. 6 (2), 2011, pp. 128–142.

[19] Y. Zhou, Y. Fang, Y. Zhang, Securing wireless sensor networks: a survey, IEEE Communications Surveys & Tutorials, vol. 10 (3), 2008, pp. 6–28.

[20] A. V. Taddeo, M. Mura, A. Ferrante, Qos and security in energy-harvesting wireless sensor networks, in: Proceedings of ICETE SECRYPT 2010, Athens, Greece, 2010, pp. 1–10.

[21] S. Pelissier, T. Prabhakar, H. Jamadagni, R. Venkatesha Prasad, I. Niemegeers, Providing security in energy harvesting sensor networks, in: Proceedings of IEEE CCNC 2011, Las Vegas, Nevada, USA, 2011, pp. 452–456.

[22] J. Goodman, A. P. Chandrakasan, A. P. Ch, An energy-efficient reconfigurable public-key cryptography processor, IEEE Journal of Solid-State Circuits, vol. 36, 2001, pp. 1808–1820.

[23] A. Beimel, Secure Schemes for Secret Sharing and Key Distribution, Ph.D. thesis, Israel Institute of Technology, 1996.

[24] N. FIPS, 197: Announcing the Advanced Encryption Standard (AES), Information Technology Laboratory, National Institute of Standards and Technology, vol. 5 (4), 2001.

[25] D. Page, N. P. Smart, F. Vercauteren, A comparison of mnt curves and supersingular curves, Applicable Algebra in Engineering, Communication and Computing, vol. 17 (5), 2006, pp. 379–392.

[26] P. S. L. M. Barreto, S. D. Galbraith, C. O'Eigeartaigh, M. Scott, Efficient pairing computation on supersingular abelian varieties, Designs, Codes and Cryptography, vol. 42 (3), 2007, pp. 239–271.

[27] Crossbow Technology, TelosB mote platform datasheet, 2004. Document Part Number: 6020-0094-01 Rev B.

[28] Crossbow Technology, MICA2 mote platform datasheet, 2003. Document Part Number: 6020-0042-04.

[29] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, Recommendation for Key Management - Part 1: General (Revised), Technical Report 800-57, NIST Special Publication, 2007.

[30] Panasonic Corporation, Electric Double Layer Capacitors (Gold Capacitor)/ HW Series, 2008.

[31] P. Barker, Ultracapacitors for use in power quality and distributed resource applications, in: Proceedings of the 2002 IEEE Power Engineering Society Summer Meeting, Chicago, IL USA, 2002, pp. 316–320.

[32] T. Zhu, Z. Zhong, Y. Gu, T. He, Z.-L. Zhang, Leakage-aware energy synchronization for wireless sensor networks, in: Proceedings of ACM MobiSys 2009, Kraków, Poland, 2009, pp. 319–332.

[33] M. Tavakoli, L. Turicchia, R. Sarpeshkar, An ultra-low-power pulse oximeter implemented with an energy-efficient transimpedance amplifier, IEEE Transactions on Biomedical Circuits and Systems, vol. 4 (1), 2010, pp. 27–38.

[34] O. Chipara, C. Lu, T. C. Bailey, G.-C. Roman, Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit, in: Proceedings of ACM SenSys '10, Zurich, Switzerland, 2010, pp. 155–168.

[35] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, A. Terzis, Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless, in: Proceedings of ACM SenSys '10, Zurich, Switzerland, 2010, pp. 1–14.

[36] U. M. Colesanti, S. Santini, A. Vitaletti, Dissense: An adaptive ultralow-power communication protocol for wireless sensor networks, in: Proceedings of IEEE DCOSS 2011, Barcelona, Spain, 2011, pp.

1–10.

[37] M. Gorlatova, A. Wallwater, G. Zussman, Networking low-power energy harvesting devices: Measurements and algorithms, in: Proceedings of IEEE INFOCOM 2011, Shanghai, China, 2011, pp. 1602–1610.