Assessing Security-Critical Energy-Efficient Sensor Networks

Y. W. Law, S. Dulman, S. Etalle, P. Havinga
Department of Computer Science, University of Twente
Postbox 217, 7500 AE Enschede, The Netherlands
{ywlaw, dulman, etalle, havinga}@cs.utwente.nl

June 10, 2002

Abstract

In the EYES project (http://eyes.eu.org), we are investigating self-organizing, collaborative, energy-efficient sensor networks. This study is devoted to the security aspects of the project. Our contribution is three-fold: firstly, we present a survey, where we discuss the dominant issues of energy-security trade-off in the network protocol and key management design space. From there we set out future research directions for our security framework. Secondly, we propose an assessment framework based on *system profile*, with which we have managed to carve out manageable design spaces from the seemingly infinite possibilities of ad hoc mobile wireless networks. Finally, we have benchmarked some well-known cryptographic algorithms in search for the best compromise in security and energy-efficiency, on a typical sensor node. Our preliminary investigations also cover an important parameter in the design space: the resource requirements of the symmetric key algorithms RC5 and TEA.

1 Introduction

The vision of ubiquitous computing requires the development of devices and technologies, which can be pervasive without being intrusive. The basic components of such a smart environment will be small nodes with sensing and wireless communications capabilities, able to organize flexibly into a network for data collection and delivery. Within this framework, the concept of sensor networks was born. It is useful to think of such networks as *sensor-based ad hoc mobile wireless networks*, which combine the characteristic of ad hoc mobile wireless networks (ad hoc networks in short) on the system level, with the characteristics of sensors on the component level. Instead of giving a precise definition of ad hoc networks [47], we believe it suffices to list three pivotal properties:

1. **Ad hoc**: The network set-up is *possibly* short-lived.
2. **Mobile**: The nodes are not attached to any fixed communications infrastructure *as well as* fixed energy supply.
3. **Wireless**: The nodes communicate wirelessly.

These three properties imply a series of constraints, and together with the constraints imposed by sensors, among which energy being the predominant, they form the basis of our security research. It is counter-productive to enforce the definition that sensor networks only consist of sensor nodes, because there is no reason why other types of devices should be disallowed from becoming a part of the network to communicate with the sensors.

In the literature, we find many proposals concerning the security requirements of ad hoc networks [15, 22, 23, 33, 61, 62]. In standard security, we are concerned with confidentiality, authentication, integrity, nonrepudiation, access control, availability, but in the context of ad hoc networks, satisfying all these requirements does not ensure the security of the system as a whole [5, 8, 33, 35, 36, 37]. The hardware and energy constraints of the sensors add to the difficulty [10, 40]. On the current research lanscape, there is not yet a clear unifying pattern among most of the research results so far. We are motivated to put in order the field of security of ad hoc networks by surveying of the available results. Actually our contribution is three-fold:

1. **Survey**: We have done a broad survey of existing proposals in the area of communications protocols and key management architectures. We believe that targeting these two areas are sufficient to cover the security of the system as a whole, on both the low-level communications part and the high-level application part.
2. **System profiles**: We introduce *system profiles* as an effective means for assessing an application and categorizing the application according to its actual specification and requirements. Under this framework, we make it possible for architectural designs to relate to a set of fine-grained properties they should comply with, instead of a hypothetical and often arbitrary set of assumptions. By keeping one system profile in perspective at a time, this helps prevent oversight and underestimation.
3. **Benchmarks**: Our investigation extends to the benchmarking of some well-known cryptographic algorithms in search for the best compromise in security and energy-efficiency, on a typical sensor node.

The significance of our benchmarks is not so much theoretical, but practical such that it allows us to evaluate the applicability of current technologies and explore the subtleties of the resource limitation.

In Section 2, we present a survey of the current proposals in the design space of security protocols. Section 3 is devoted to our system profile proposal, and Section 4 describes how this proposal is used to derive a preliminary conceptual design of the

EYES prototype. Section 4 also has the details of our benchmarks. Finally Section 5 gives the conclusion.

2 Security Protocol Design Space

In this section, we present a survey of the state-of-the-art in network protocols and key management, in the context of security and energy-efficiency. As a sidenote, the level of security achieved and the level of energy conserved are by nature contradictory. Therefore, while we do not always bring energy into the picture during the course of our discussion, we assume the fact that the more overhead a security protocol introduces, the more energy consuming the protocol is. We also assume that the conventional threat model in Figure 1 applies [52, p. 8].

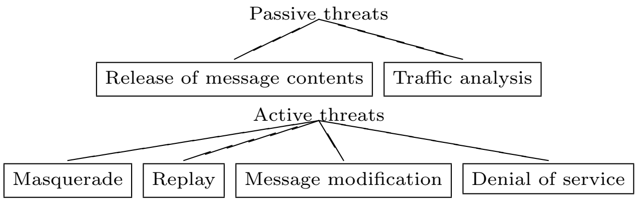

Figure 1: Threat model

2.1 Network Protocols

Our starting point is the Open System Interconnect (OSI) model. However we only concentrate on two layers: the data link layer and the network layer. We are not interested in the physical layer. It suffices to say that on one hand there exists the possibility of denial-of-service attacks by signal jamming, on the other there are such well-known counter-measures as spread-spectrum and frequency-hopping [29]. We also do not consider the transport layer. The reasoning goes like this: if the data link layer and network layer are secure, then the transport layer can be sure that the packets it receives from the network layer are confidential, authenticated and original. What is left for the transport layer to do is the usual grunt work of flow control, packets reordering, error recovery, connection states management etc. The application, presentation and session layers are only abstractions of an user of the underlying layers – they do not contribute to the machinery of networking.

Data Link Layer In the wireless world, data link security is more critical than its wired counterpart, as data is transmitted in an open insecure medium. In the classical example of "wardriving", hackers driving through the parking lots of offices are able to capture raw packets in the clear [34]. The competing technologies include Bluetooth (http://www.bluetooth.com), IEEE 802.11 [24], HomeRF (http://www.homerf.org) and HiperLAN2 (http://www.hiperlan2.com). Bluetooth is primarily designed for wire replacement applications. Although Bluetooth does provide link-level encryption and entity authentication using a challenge-response scheme [17], it does not protect the network layers, nor does it cater for intermittent group connectivity, multi-hop routing and unattended operations [9, 56]. IEEE 802.11 has the advantage of wide market acceptance, otherwise its infamous Wired Equivalent Privacy (WEP) has been irreparably broken [2, 53]. HomeRF has a better security architecture in comparison [21]. HiperLAN2 is widely regarded as superior to IEEE 802.11, but it is evolving rapidly, so it is still early to comment. This much said, all these protocols actually operate at frequency bands that are entirely unsuitable for the low-power transceivers used in sensor networks (cf. Section 4.2): Bluetooth uses 2.45GHz, IEEE 802.11 uses 2.4GHz, HomeRF uses 2.4GHz and HiperLAN2 uses 5GHz. In terms of energy conservation [27], PAMAS [48] seems to be a viable alternative, but its security framework has yet to be developed. It is thus seen that no protocol to-date actually stands out as the ideal candidate for the data link layer protocol of sensor networks.

Network Layer Routing protocols operate at the network layer. Since controlling how information flows from a source to a destination underlies the very basis of networking, it is the most important layer in consideration here. An entire network can be compromised by disrupting the routing alone.

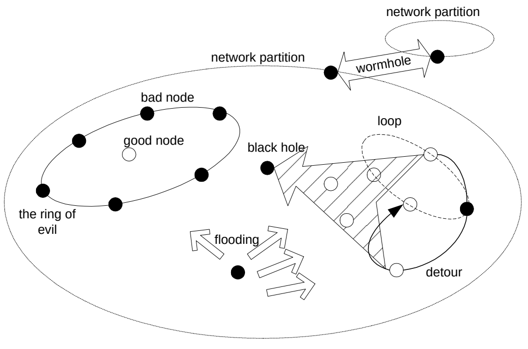

Figure 2: Attacks on routing

We define *malicious behaviors* as any form of behaviors, whether intentional or unintentional, beneficial or ambiguously beneficial to the perpetrator, that results in the disruption of the normal operation of the network. With this definition, malicious behaviors may thus seem arbitrary and often manifest as denial-of-service attacks, to which sensor networks are particularly vulnerable. Of particular interest from an energy perspective is the battery exhaustion or *sleep deprivation torture attack* [51]. It is easy to achieve this kind of attacks by disrupting the routing fabric, because a disrupted routing fabric causes energy to be wasted on erratic routing. Examples (Figure 2) [22]: (1) without authentication, a node can easily create *black holes*, or network endpoints that aggressively sinks and drops messages that are routed through them; (2) similarly, a node can *flood* the network with illegitimate routing messages; (3) *detours* and *loops* can be introduced to misdirect traffic, possibly through congested or energy-depleted routes; (4) a *wormhole* is a covert channel between a pair of attacker nodes that creates a virtual vertex cut; (5) a group of colluding malicious nodes may gang up and *hijack* a group of good nodes by refusing to route their packets, dropping their packets silently, or injecting bogus packets; (6) *blackmail*: in some scheme (discussed below) where the reliability of a node is rated by the neighbours of the node, neighbouring malicious nodes may collude and "spread bad words" about an innocent node. Looking at these scenarios, it is obvious why authentication is fundamental to the well-being of routing protocols. However as shall be shown, this alone is not enough.

Before looking at the protocols proposed for ad hoc networks, the first point we want to address is what renders existing technologies such as Mobile IP inapplicable. Although Mobile IP can easily be extended to support ad hoc networking [31], the security mechanism of Mobile IP has to be supplemented by IPSec [13]. Unfortunately the effective implementation of the Public Key Infrastructure (PKI) required by IPSec, especially the distribution of public keys and certificates, still poses a major unsolved problem in sensor networks. Hansen [18] mentions the difficulty in interoperating Mobile IP, IPSec and firewalls. However the biggest reason actually lies in the fact that Mobile IP is a means for preserving the IP address of a mobile node in foreign networks, through the interplay between the *foreign agents on the foreign networks* and the *home agent on its home network*. Sensor networks in general do not impose such a requirement and therefore do not deserve such infrastructure and overhead.

Therefore there is a need to introduce new routing protocols. A lot of protocols have been proposed [15], among which the Ad hoc On-demand Distance Vector (AODV) protocol [57] and the Dynamic Source Routing (DSR) protocol [43] have recorded very good performance [39]. Unfortunately security issues arise with these protocols, because security features are not designed built-in. A number of "rescue efforts" have emerged as a result:

- Most notably, Marti et al. [33] pioneer the idea of *watchdog* and *pathrater*. Although the solution is far from perfect – its weaknesses being explained in the paper itself, it is nevertheless the first attempt at using collective evaluation at curbing the misbehavior of non-colluding nodes: every node implements a watchdog that, operating in *promiscuous mode* (which consumes a great amount of energy), constantly monitors the packet forwarding activities of its neighbours, and a pathrater that rates the transmission reliability of all alternative routes to a particular destination node, according to the reports of the watchdog. Although proposed as a general mechanism for fortifying any general routing protocol, it is essentially only practical for source routing protocols. Collusion between malicious nodes remains an unsolved problem.

- This line of investigation has been followed up. Michiardi et al. [35, 36, 37] go further by generalizing the rating mechanism. Now the neighbors of any single node collaborate in rating the node, according to how well the node execute the functions requested from it. The difficulty of this scheme lies in how an evaluating node is able to evaluate the result of a function executed by the evaluated node. Depending on the function executed, the evaluated node may be able to cheat easily. Or the result of the function may require significant overhead to be communicated to the evaluating node. If the requested function is simply forwarding packets, then the scheme faces similar difficulties faced by the watchdog mechanism. The problem with colluding nodes raises the usual concern. Despite the inadequacies, Michiardi [35] does strike a resonant chord on the importance of making "selfishness" pay. Selfishness is different from maliciousness in the sense that selfishness only aims at saving resources for the node itself by refusing to perform any function requested by the others, such as packet forwarding, and not at disrupting the flow of information in the network by intention. From this we can see that there are in fact two kinds of "badness": maliciousness and selfishness. We agree with Michiardi that selfishness is not solvable by virtue of classical security alone.

- Still along the same line of investigation are Buttyan et al. [8] and Blazevic et al. [5] who conceptualize the motivation for nodes not to be selfish as *nuglets*, a sort of virtual currency. To insulate a node's nuglets from illegal manipulation, a tamper-resistant *security module* [3, p. 280] storing all the relevant IDs, nuglet counter and cryptographic materials (but not the code) is compulsory. The cross-certification architecture calls for public key cryptography, which exerts a high demand on computing resources. The amount of overhead is also a concern.

- Another line of thought can be traced to Yi [59], who proposes levels of protection as a negotiable metric in route discovery. In this way, a pair of nodes establishes a certain application-specific level of protection before any security-sensitive traffic begins. As mentioned, the simple availability of cryptographic protection does not solve the basis of security problems. Security-awareness does not equal security-sufficiency.

SPINS (Security Protocols for Sensor Networks) is one exception where routing is an application of a security framework [40]. The main idea of SPINS is to demonstrate the feasibility of security with very limited computing resources, by using symmetric cryptography alone, without emphasis on general applicability. The target wireless network is homogeneous and static. A central base station acts as the only point of trust, i.e. all nodes only trust the base station and themselves. As a result, the routing model that can be facilitated by SPINS is fairly limited: route discovery depends solely on the detection of authenticated beacons broadcast by the base station. Node-to-node communication necessitates authentication via the base station. Hu et al. [22] adsorb the ideas of SPINS and came out with a hardened version of DSR called Ariadne. One of the requirements is that every node has to be able to generate an one-way key chain. Since the memory of a sensor node is limited, it cannot afford to generate a long key chain, and so has to spend a lot of time generating keys. By enforcing authenticity alone, Ariadne does not guard against attacks by multiple colluding nodes.

We conclude that there is currently no best candidate for the selection of routing protocols so far by the following observation. In terms of performance, there is no all-rounder. Broch et al. [6] discover that DSR outperforms AODV except in overhead byte count. Johanson et al. [26], using scenario-based analysis, and Das et al. [12] both conclude that AODV performs better than DSR at high load and high mobility, but poorer otherwise. The reasons for these results are: (1) the source routes of DSR, although enables routes to be discovered more efficiently, also incurs a higher cost in terms of overhead byte count; (2) AODV, with the use of routing tables, has an upper hand in response time though at the expense of overhead packet count. It is worth noting that the throughout of AODV and DSR are comparable in most cases. One attempt at trying to improve the performance of AODV at low load, by implementing *path collection*, failed [28]. The fact that DSR does not support multicast at the time of writing, while AODV does, compounds the difficulty further in choosing the better protocol. The good news is that the power-saving technique of Singh et al. [49] can be incorporated in any existing routing protocol. In sensor networks, energy is a more important metric than throughput.

2.2 Key Management

Leaving the domain of network protocols, the bulk of the remaining literature falls on key establishment. Carman et al. [10] has performed a broad survey of some important key establishment protocols from the perspective of energy efficiency. Through simulations and measurements, the survey concludes that there is currently no protocol that combines the best energy efficiency and security features. In other words, an optimal strategy would have to be a hybrid one that dynamically selects the appropriate sub-protocol, i.e. arbitrated, or pairwise, or group keying, depending on such factors as the availability of multicast, the distribution of remaining energy, the group size, the node density and so on.

Basagni et al. [4] reason that since these sensor nodes, or *pebbles*, are so resource-constrained that only symmetric key cryptography is feasible, it is inevitable that clusters of nodes, or *pebblenets*, would have to share a symmetric key, and on a network-wide level, all *pebblenets* would share a *traffic encryption key.* Like SPINS, pebblenets use only symmetric cryptography. The disadvantage is that once a node is compromised, forward secrecy is broken, therefore tamper-resistance becomes crucial. Thus resurfaces our observation that cryptography alone is not enough. Coupling cryptography with a collaborative monitoring and evaluation scheme facilitates a second line of defense.

The proposal of Zhou et al. [62] is actually intended to secure routing, but its key management service is of interest here. For authenticating routing messages, every packet is signed. The verification process depends on the key management service that is distributed over $t + 1$ servers among n nodes, where $n \geq 3t + 1$, so that at most t may be compromised, by the principle of *threshold cryptography* [45, p. 71]. A key management server not only has to store its own key pair, but also the public keys of all the nodes in the network. The difficulty includes the storage requirement exerted on the servers which must potentially be specialized nodes in the network, and the overhead in signing and verifying routing message both in terms of computation and of communication.

Hubaux et al. [23] go a step further than Zhou, by requiring each node to maintain its own *certificate repository.* These repositories store the public certificates the node themselves issue, and a selected set of certificates issued by the others. The *performance* is defined by the probability that any node can obtain and verify the public key of any other user, using only the local certificate repositories of the two users. The dilemma is: too many certificates in a sensor node would easily exceed their capacity, yet too few might greatly impact the performance (as previously defined) of the entire network. For example: although the size of a certificate depends on many parameters, it is typically around 1KB. In the context of SPINS, 1KB is already more than 20% of the available 4500-byte code space. Depending on the size of the trusted graph size, the required number of stored certificates for acceptable performance easily exceeds 10.

Lastly, Zhang et al. [61] argue on the importance of intrusion detection for ad hoc networks, and propose an extension of existing techniques to the ad hoc wireless case. We believe that the time is still too early to fingerprint anomalies in ad hoc networks when it is even unclear how an efficient protocol should behave. And since the scheme entails intrusion detection agents to be installed on every node, it is a resource expenditure that our current envisioned networks cannot afford.

3 Profiling Application Patterns

In the previous section, we reviewed a wide range of proposals in the design space of network protocols and key management. We observe that the results are loosely knitted. We also observe that it is impossible to formulate a single framework that fits all ad hoc networks. For example, while some architectures stipulate every node to be tamper-resistant (e.g. Terminodes [5]), some do not (e.g. SPINS). Some explicitly call for PKI (e.g. Terminodes), yet some explicitly avoid it (e.g. SPINS). Therefore, instead of attempting to provide a one-size-fits-all solution, we propose a *unified assessment framework* based on system profiles. In the next section, we will discuss how we use this framework as our starting point in the EYES project.

The inspiration of system profiles comes from Sun's Java$^{\text{TM}}$ 2 Platform, Micro Edition (J2ME) (http://java.sun.com/j2me). J2ME is the Java platform targeted at devices such as PDAs, cellphones, pagers and so on. The challenge is to put a Java Virtual Machine (JVM) and its associated libraries in every one of them. One way is to pick the lowest common denominator. However then a PDA would become only as powerful as the lowest common denominator, from the point of view of Java. As a solution, Sun introduced the concept of *configurations* and *profiles* [54]. Currently there are two configurations: Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). CLDC targets resource-constrained devices with typically a 16/32-bit processor, and 512 Kb or less memory available for the Java platform and applications; whereas CDC is for more powerful devices. The strategy of J2ME attests to the fact that one size does not fit all. We are adopting a similar profiling strategy. The only difference is what we are profiling is not the nodes themselves, but the applications. We believe such an approach is consistent with real-world industrial experience.

We categorize different applications of ad hoc networks into different system profiles, each of which is defined by a set of boolean *critical system parameters.* The following critical system parameters have been defined:

1. **Data Confidentiality (DC)** specifies the requirement for data confidentiality. Rationale: not all types of application require data confidentiality as part of their security requirements. A brute force approach of encrypting all data irregardless of necessity does not necessarily provide the highest level of security, nor does it conserve energy.

2. **Tamper Resistance (TR)** speficies whether there is a need for tamper resistant hardware for protecting every node in the network. Rationale: to entrust a node with a key, we have to make sure the node itself does not betray us by divulging its key to unauthorized parties upon tampering. If not all nodes in the network can be made tamper-resistant, it is insufficient to rely on cryptography alone to ensure the integrity of any node, since any node can be tampered, with its keys compromised and its program modified. For such networks, cryptographic material cannot be kept at any node for any extended period, and supplementary security means are necessary. A way of protecting the keys embedded in the nodes is to implement Stajano's *reverse metempsychosis* [51], i.e. when the node (duckling) is not in use, it is put to rest, with its keys reset (soul taken away), until the next time it is put to new use with another set of keys imprinted (resurrected). Another way is exploiting what Stajano [50] calls the "fraternal love among sibling ducklings": when a node fails to detect the presence of at least n siblings for some time, it would refuse to work. In our case, if it fails to detect any sibling at all, it shall reset its keys.

3. **Public Key Cryptographic Capability (PKCC)** refers to the capability of any node in the network to perform public key cryptography. In general, processor speed is not always an issue, the deciding factor is the availability of sufficient RAM. Rationale: this parameter determines whether public key cryptographic technologies can be applied. Note however that the fact that this parameter is true does not guarantee that public key cryptography can be used extensively. It only indicates that the technology *can* be used. The degree of usage depends on the architectural design.

4. **Rich Uncles (RU)** refers to the availability of Rich Uncle nodes, which are resourceful nodes, both in terms of computing resources and energy, that are suitable for the role of certification authorities, for example in the Rich Uncle Protocols [10]. These nodes might be floating or might be gateways to some external wired networks. If all nodes are equally "rich" (i.e. the network is homogeneous), we may assume that either every node can be a Rich Uncle or no node can be a Rich Uncle to avoid an unbalanced distribution of energy. Rationale: their existence confirms the possibilities of relegating resouce-intensive tasks and assigning important security roles to them, thereby facilitating the use of certain hierarchical architectures and public key cryptography.

It does no harm to stress that mobility is *not* a parameter because it is obviously always true. This selection of parameters is not meant to be exhaustive or definitive and yet we find it an unambiguous way of categorizing the types of system we know so far.

Therefore in our definition, there are 16 kinds of system profiles. We give a system profile an ID, called SPID, according to the value of: $(DC) \times 2^0 + (TR) \times 2^1 + (PKCC) \times 2^2 + (RU) \times 2^3$, where a parameter takes on a value of 1 if it is true, or 0 if it is false (recall that all parameters are boolean). Below is a few examples of how we can classify some typical ad hoc network systems (we do not want to limit our perspective to sensor networks just yet):

- **Battlefield interpersonal communication** is as illustrated in [56, p. 270] a scenario where telecommunication devices, carried by vehicles and soldiers, communicate in an ad hoc fashion without the need as well as danger for using a base station. The requirement for Data Confidentiality and Tamper Resistance is obvious. The assumption for Public Key Cryptographic Capability can also be justified, even though Rich Uncles may not be as readily assumed.

Parameter	DC	TR	PKCC	RU
Value	T	T	T	F
SPID	7			

Table 1: Parameters for battlefield interpersonal communication

- **Battlefield sensor surveillance** is the class of applications in which minute wireless sensors are dispatched in military zones for critical surveillance [56, p. 270]. [30] cites the use of chemical sensors, broad-spectrum acoustic sensors, seismic sensors, video sensors, imaging sensors etc. Signals Intelligence data are meant to be gathered from Unmanned Aerial Vehicles (UAVs) and relayed to the forward operating base for analysis and correlation. Because of the low cost and disposable design of sensor nodes, Tamper Resistance, Public Key Cryptographic Capability and Rich Uncles cannot be assumed.

Parameter	DC	TR	PKCC	RU
Value	T	F	F	F
SPID	1			

Table 2: Parameters for battlefield sensor surveillance

- **Spontaneous networking** is as described in [16] a technology that allows people to meet and use their laptops, PDAs, tablets etc. to start collaborating on some tasks through wireless networking, i.e. in the absence of a fixed infrastructure. For the same reason why IPsec is invented, Data Confidentiality is important. Tamper Resistance, as applied to consumer hardware, cannot be assumed. Public Key Cryptographic Capability is generally available although the performance varies widely across the classes of device. By the psychological reasoning that nobody wants to spend more energy than the others, we can assume that nobody wants to be a Rich Uncle.

Parameter	DC	TR	PKCC	RU
Value	T	F	T	F
SPID	5			

Table 3: Parameters for spontaneous networking

- **SPINS-type sensor networks** refer to networks of sensors connected to a single base station. The presence of the base station immediately guarantees the existence of a Rich Uncle. The rest of the parameters go without saying.

Of course, we lack experience in military applications to be able to profile them accurately, but this is meant to demonstrate

Parameter	DC	TR	PKCC	RU
Value	T	F	F	T
SPID	9			

Table 4: Parameters for SPINS-type sensor networks

how widely different applications can be profiled – in fact they all have different SPID's. System profile is a means of classification and assessment, it does not dictate what architecture should be adopted for which particular profile.

4 Security in EYES

In this section, we demonstrate where the EYES architecture fits in the assessment framework of system profiles. We start by profiling the EYES project prototype, citing possible usage scenarios. We then explore some design issues and explain how the associated design decisions are made. We conclude this section with some benchmark results that we think are crucial to an energy-efficient implementation strategy.

4.1 System Outline

In the EYES project, we envision our prototype to be an application of sensor networks for intelligent buildings. The context in discussion consists of an office building, the employees working in the building and the extensive network of sensors that runs through the building. We call the device that allows an employee to interact with the sensor network a *devEYES*. The form of a devEYES is not limited: it can be a laptop with a wireless card, a wireless PDA, a mobile phone, an electronic badge with RF capability, or even some fancy pendant with an embedded wireless chip.

The sensors are meant to collaborate to achieve some desired functions. Often in the course of performing such functions, privacy and security-sensitive data are transmitted, so Data Confidentiality is to be upheld. For the reason of cost, Tamper Resistance is not assumed. We expect our node to have Public Key Cryptographic Capability because each sensor will carry a 1MB serial RAM, which is large enough for the purpose. We also do not want to rule out the possibilities of Rich Uncles, since the office environment is largely under our control.

Parameter	DC	TR	PKCC	RU
Value	T	F	T	T
SPID	13			

Table 5: Parameters for EYES prototype

A non-exhaustive list of usage scenarios might be: (1) **remote control**: on entering the office building, Alice the Manager would have her office computer automatically switched on, through Intel's Wired for Management (WfM) Preboot eXecution Environment (PXE) [25]; (2) **access control**: standing in front of the door alone, and possibly with the help of biometric authentication [19], Alice is able to authenticate herself to the door and be allowed into her office; (3) **crime detection**: devEYES'es in the form of dongles (assuming they are hard to be removed) can be attached to expensive office equipments, so that when the equipments trespass their respective allowed perimeter, alarms would be activated, and their location pinpointed; (4) **suspicious behavior detection**: sensors detecting suspicious behaviors [46] may alert the security personnels; (5) **comfort adjustment**: sensors in a room may detect the amount of lighting, airflow, temperature, or even monitor anxiety level and heartbeat of the occupants, in order to adjust the comfort level of the room in collaboration, possibly with a say from the occupants' devEYES; (6) **disaster alarm**: fire, earthquake, chemical sensors distributed throughput the building can detect anomalies and warn occupants through alarms and their devEYES'es.

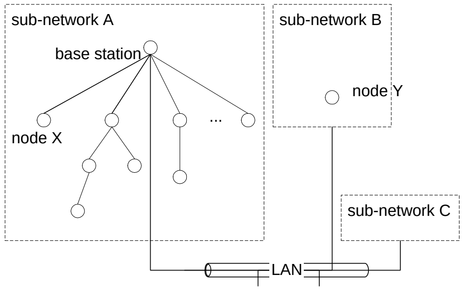

Figure 3: Prototype architecture: First attempt

The EYES prototype seems deceptively easy to implement at first sight (Figure 3). One might be tempted to classify this as a SPINS-type network, but such an assumption is flawed, because then a node X in a sub-network A can only communicate with the other node Y in another sub-network B via both servers of A and B, *even when* X and Y are actually within radio range – an unnecessary overhead. Therefore sub-network boundaries should not exist.

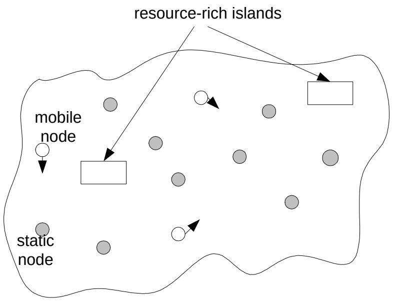

Figure 4: Prototype architecture: The refined model

Thus a more proper model should allow paths to be formed dynamically depending on circumstantial priorities and contraints (especially energy). To be specific, we imagine a sea of sensors, some of which mobile and some static, peppered with tiny islands of relatively resource-rich devices (Figure 4). Pebblenets provide a hint on how to lay down a structure on such sensor sea, but if the size of clusters is too small, cluster-join and cluster-leave messages would overwhelm the traffic, assuming some nodes are mobile for a long period of time. So far an efficient architecture has yet to be worked out.

A final note about a *non-assumption*: bidirectional connectivity. Researchers tend to argue that since IEEE 802.11 requires links to be bidirectional, they have "no choice" but to go along with the assumption. In fact, links become unidirectional due to differing propagation patterns, or sources of interference [39, p. 144]. In other circumstances, e.g. military applications, unidirectional connectivity can be due to necessity, environmental jamming, or adversarial jamming [10, p. 35]. The reason we mention bidirectional connectivity in the context of security is that some security schemes (e.g. Ariadne) require bidirectional connectivity. We then know what to rule out when we design our security architecture.

4.2 Benchmarks of Cryptographic Algorithms

Cryptographic algorithms are the bedrock of security services but they put a toll on computing resources, therefore it is important to select the algorithms which can provice the best level of security through the most energy-efficient means. For this matter, we have assessed the deployability of a few symmetric key algorithms through benchmarking. In particular, we have tested RC5 [44] and Tiny Encryption Algorithm (TEA) [58] on our targeted platform. They are chosen because, unlike heavy-weight protocols such as AES [11], KASUMI [1] etc., they do not use multiplication and large tables. Below, we give the details of our hardware platform and development environment.

Our targeted processor is a Texas Instruments MSP430x149 [55], mounted on a board custom built by Nedap (http://www.nedap.nl). The processor can operate in 4 low-power modes and 1 active mode. In the active mode, the current is 420μA when the voltage is 3V at a frequency of 1MHz, which means that the energy per instruction cycle is 0.5292 pJ. The transceiver is a RF Monolithics' 868.35MHz TR1001 [42]. If energy is provided by Energizer's lithium/manganese dioxide battery CR2450 [14], (average service capacity of 575mAh), the sensor would last for at most 2.7 days if the microcontroller constantly operates in active mode, and the transceiver operates in transmit mode for half of the time, receive mode for the other half of the time. The available energy is indeed very limitied. The following is a comparison of the sensor node used in EYES with that is used in SPINS:

	EYES node	Smart Dust
CPU	16-bit, 8 MHz	8-bit, 4 MHz
Flash memory	60 KB	8 KB
RAM	2 KB	512 B
Frequency	868.35 MHz	916 MHz
Bandwidth	115.2 kbps	10 kbps

Table 6: Comparison of EYES node with Smart Dust

The compiler is IAR Systems' (http://www.iar.com) MSP430 C-Compiler V1.26A/W32. The debugger is C-Spy V2.3, also from the same company. For maximum size optimization during compilation, the switch "-z9" is used. Similarly the switch "-s9" is used for maximum speed optimization. For obtaining the code size and RAM size, the implementation is compiled without any debug information. However for obtaining the instruction cycles, it is. This is in order to facilitate profiling by the debugger.

For simplicity, all algorithms are written in C, and measured in the electronic codebook (ECB) mode instead of the counter (CTR) mode as suggested for SPINS. To measure just the core of the algorithm, the source code is devoid of error checking.

For RC5, both the reference implementation [44] and Schneier's implementation [45, p. 659] are used. The difference between the two implementations is primarily a matter of coding style. The RC5 parameters are chosen as: word size = 16 bits, number of rounds = 12 (nominal), key length = 16 bytes. In other words, the block size is 32 bits.

For TEA, the extended version is used [38]. The key length and block size are the same as that of RC5.

Implementation	Optimization	Code size (bytes)
RC5 (reference)	No	746
	Size	646
	Speed	688
RC5 (Schneier)	No	682
	Size	612
	Speed	640
TEA	No	868
	Size	798
	Speed	838

Table 7: Measurements of code size

It is interesting to note that the effect of the compiler is when using maximum speed optimization, the code size obtained is even smaller than that of the unoptimized (Table 7). Secondly, different coding styles result in different code size. Lastly, both implementaitons of RC5 clearly have a smaller code size than TEA. These results of RC5 here are smaller than SPINS' because, as mentioned, CTR mode is not used.

Implementation	Static		Automatic	
		Key setup	Encryption	Decryption
RC5 (reference)	52	32	16	16
RC5 (Schneier)	52	28	12	12
TEA	16	2	28	28

Table 8: Evaluation of static and automatic variable size (in bytes)

For RAM size, we consider data memory that is allocated statically (for static variables) as well as alocated on the stack (for automatic variables). Currently we do not use dynamic memory allocation and deallocation. We do not consider the memory for the plaintext and ciphertext. The sizes are evaluated by inspecting the source code. Actually TEA does not require any key setup, however to present a uniform API to every implementation, we copy the key from the function argument into a static storage in the "artificial" key setup phase of TEA. Finally it can observed that TEA uses less memory (static and automatic) than RC5 does because RC5 uses an *expanded key table* while TEA does not (Table 8).

Assuming key setup is done very infrequently, the energy cost is dominated by the cost of encryption and decryption. Table 9 shows that TEA consumes the lowest amount of energy, followed by the reference implementation of RC5 and Schneier's implementation of RC5 (in the same class of optimization), exactly the opposite order of code sizes (Table 7), thus presenting a typical size versus energy cost trade-off.

To minimize both code size and energy, we seek the implementation that gives the lowest value in the energy-code-size product. Note that the energy value used in this product is the energy for encryption and decryption *and* key setup, and the code size is the total code size. This turns out to be the

Impl.	Opt.	Energy	Energy-Code-Size
		(nJ)	Product (uJ Bytes)
RC5	No	6.4403	17.9737
(reference)	Size	6.1779	23.7811
	Speed	6.1779	4.585
RC5	No	6.8553	14.0359
(Schneier)	Size	6.4985	16.6073
	Speed	6.4985	4.2581
TEA	No	5.2042	14.6360
	Size	5.2725	17.0766
	Speed	4.2903	3.6485

Table 9: Measurements of energy for encryption and decryption, and the associated energy-code-size product (Impl.=Implementation, Opt.=Optimization)

speed-optimized version of TEA (Table 9). One final remark: although we do acknowledge that fact that due to their simplicity, the algorithms can be coded in assembly language for better results, we do not want to overdo it at this stage.

Apart from RC5 and TEA, we are interested in benchmarking more symmetric key algorithms, especially TREYFER [60] and VINO [41]. Apart from symmetric key algorithms, we also plan to investigate some public key alrogithms: ECC (http://www.certicom.com), XTR [32] and NTRU [20]. RSA is ruled out because of its poor performance on low-end devices [7].

5 Conclusion

From our survey, we have identified the need for a secure energy-efficient data link layer protocol, and a secure energy-efficient routing protocol. The prevalent approach is to patch existing protocols with security features, but we suspect that an integrated design would be more robust, as suggested by [9]. Also, by virtue of selfishness, a brute force cryptographic framework is incapable of solving the whole problem. A framework in which multiple nodes can collaborate to evaluate the reliability of a node offers a resilient approach to isolating misconducting members.

Realizing that one size does not fit all, we have introduced a unified assessment framework based on the notion of system profiles, not only to remind ourselves of the valid set of assumptions and requirements, but also to allow ourselves to concentrate on one profile at a time. Our research exercise has testified it to be a useful tool in assessing ad hoc networks.

Lastly, it has to be emphasized that security architecture alone is not a panacea to all possible means of subversion. For example, if a set of nodes is entirely surrounded by misbehaving nodes within their transmission range, whether colluding or not, the only possible means for them to come out of the siege is to roam to a friendlier and safer neighbourhood. If the besieged nodes are static, the territory is lost.

6 Acknowledgements

We would like to express our appreciation of Pieter Hartel for reviewing the text, Jaap-Henk Hoepman for his indispensable advice, and Cheun Ngen Chong for his information sources.

References

[1] 3rd Generation Partnership Project. 3GPP KASUMI Evaluation Report. Security Algorithms Group of Experts (SAGE) Report on the Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms (SAGE version 2.0), 2001.

[2] B. Aboba. The Unofficial 802.11 Security Web Page. Web page. `http://www.drizzle.com/~aboba/IEEE`.

[3] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., 2001.

[4] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti. Secure pebblenets. In *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 177–228, October 2001.

[5] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec. Self-organization in mobile ad hoc networks: the approach of terminodes. *IEEE Communications Magazine*, 39(6):164–174, June 2001.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[7] M. Brown, D. Cheung, D. Hankerson, J. L. Hernandez, M. Kirkup, and A. Menezes. PGP in Constrained Wireless Devices. In *9th USENIX Security Symposium*, Aug 2000.

[8] L. Buttyán and J.-P. Hubaux. Nuglets: A Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks. Technical Report DSC/2001/001, Department of Communication Systems, Swiss Federal Institute of Technology, 2001.

[9] D. Carman, B. Matt, D. Balenson, and P. Kruus. A communications security architecture and cryptographic mechanisms for distributed sensor networks. In *DARPA SensIT Workshop*. NAI Labs, The Security Research Division Network Associates, Inc., 1999.

[10] D. W. Carman, P. S. Kruus, and B. J. Matt. Constraints and approaches for distributed sensor network security. Technical Report #00-010, NAI Labs, 2000.

[11] J. Daemen and V. Rijmen. AES Proposal: Rijndael. specification version 2, 1999. `http://www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaeldocV2.zip`.

[12] S. R. Das, C. E. Perkins, and E. E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *IEEE INFOCOM*, pages 3–12, 2000.

[13] N. Dooraswamy and D. Harkins. *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks*. Internet Infrastructure Series. Prentice Hall, 1999.

[14] Eveready Battery Company, Inc. ENERGIZER NO.CR2450. Engineering data. http://data.energizer.com.

[15] L. Feeney. A taxonomy for routing protocols in mobile ad hoc networks. Technical Report T99/07, Swedish Institute of Computer Science, October 1999.

[16] L. Feeney, B. Ahlgren, and A. Westerlund. Spontaneous networking: An application-oriented approach to ad hoc networking. *IEEE Communications Magazine*, 39(6):176–181, Jun 2001.

[17] C. Gehrmann. Bluetooth^{TM} Security White Paper. White paper, Bluetooth SIG Security Expert Group, Apr 2002.

[18] H. Hansén. IPsec and Mobile IP in Mobile Ad Hoc Networking. Article, Department of Computer Science and Engineering, Helsinki University of Technology, Apr 2000. http://www.hut.fi/~hansen/papers/adhoc/.

[19] N. J. Henderson and P. H. Hartel. Pressure sequence - a novel method of protecting smart cards. In In J. Domingo-Ferrer, D. Chan, and A. Watson, editors, *4th Int. IFIP wg 8.8 Conf. Smart card research and advanced application (CARDIS)*, pages 241–256. Kluwer Academic Publishers, Sep 2000.

[20] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A Ring Based Public Key Cryptosystem. In J.P. Buhler, editor, *Algorithmic Number Theory (ANTS III)*, number 1423 in Lecture Notes in Computer Science 1423, pages 267–288. Springer-Verlag, Berlin, Jun 1998.

[21] HomeRF Working Group. A comparison of security in homerf versus ieee802.11b. White paper, 2001.

[22] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. Technical Report TR01-383, Department of Computer Science, Rice University, 2001.

[23] J. P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Long Beach, CA, USA*, October 2001.

[24] IEEE. IEEE Std 802.11-1997 Information Technology-telecommunications And Information exchange Between Systems-Local And Metropolitan Area Networks-specific Requirements-part 11: Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications, Nov 1997.

[25] Intel Corporation. *Preboot Execution Environment (PXE) Specification Version 2.1*, 1999.

[26] P. Johanson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of ACM/IEEE MOBICOM'99, Seattle, WA*, pages 195–206, 1999.

[27] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.

[28] J. Kjensli and G. Zaera. Adding Path Collection to AODV. Student project. http://www.zaeratech.com/projects/cs276_aodv-pc/docs/aodv_pc-results.ps.

[29] A. Lapidoth and P. Narayan. Reliable communication under channel uncertainty. *IEEE Transactions on Information Theory*, 44(6), 1998.

[30] Large Scale Networking (LSN) Coordinating Group Of the Interagency Working Group (IWG) for Information Technology Research and Development (IT R&D). Workshop on New Visions for Large-Scale Networks: Research and Applications . workshop paper, 2001. http://www.hpcc.gov/iwg/pca/lsn/lsn-workshop-12mar01/workshop-12mar01.pdf.

[31] H. Lei and C. E. Perkins. Ad hoc networking with mobile ip. In *The Second European Personal Mobile Communications Conference (EPMCC), September 30 - October 2, Bonn, Germany*, 1997.

[32] A. K. Lenstra and E. R. Verheul. The XTR Public Key System. In *Advances in Cryptology – Crypto 2000*, Lecture Nodes in Computer Science, pages 1–19. Springer-Verlag, Berlin, 2000.

[33] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 255–265, 2000.

[34] S. McClure, J. Scambray, and G. Kurtz. *Hacking Exposed: Network Security Secrets and Solutions*. Osborne McGraw-Hill, 3rd edition, Sep 2001.

[35] P. Michiardi and R. Molva. Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. In *Communications and Multimedia Security Conference*, 2002.

[36] P. Michiardi and R. Molva. Prevention of denial of service attacks and selfishness in mobile ad hoc networks. Research Report RR-02-063, Institut Eurécom, France, 2002.

[37] P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. In *European Wireless 2002: Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, February 25-28, 2002, Florence, Italy*, 2002.

[38] R. M. Needham and D. J. Wheeler. TEA extensions, 1997. http://www.cl.cam.ac.uk/ftp/users/djw3/xtea.ps.

[39] C. E. Perkins, editor. *Ad Hoc Networking*. Addison Wesley, 2001.

[40] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS:Security Protocols for Sensor Networks. In *The 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy*, pages 189–199, 2001.

[41] A. Di Porto and W. Wolfowicz. VINO: a block cipher including variable permutations. In *Cambridge Security Workshop*, Lecture Notes in Computer Science. Springer Verlag, 1993.

[42] RF Monolithics, Inc. TR1001: 868.35 MHz Transceiver. Datasheet. `http://www.rfm.com/products/data/tr1001.pdf`.

[43] Rice University. Rice University Monarch Project: Mobile Networking Architectures. Home page. `http://www.monarch.cs.rice.edu`.

[44] R. L. Rivest. The RC5 Encryption Algorithm. In *Proceedings of the 1994 Leuven Workshop on Fast Software Encryption*, pages 86–96. Springer-Verlag, 1995.

[45] B. Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C.* John Wiley & Sons, Inc., 2nd edition, 1996.

[46] H. Shao, L. Li, P. Xiao, and M. K. H. Leung. ELEVIEW: An Active Elevator Video Surveillance System. In *Workshop on Human Motion*, pages 67–72. IEEE Computer Society, 2000.

[47] B. Shrader. A proposed definition of 'Ad hoc network'. Course project report, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2002. `http://www.s3.kth.se/~brooke/Reports/pscience.pdf`.

[48] S. Singh and C. S. Raghavendra. PAMAS - Power Aware Multi-Access Protocol with Singalling for Ad Hoc Networks. *ACM Computer Communication Review*, Jul 1998. http://citeseer.nj.nec.com/460902.html.

[49] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *ACM/IEEE MOBICOM*, 1998.

[50] F. Stajano. The resurrecting duckling - what next? In *Security Protocols Workshop*, pages 204–214, 2000.

[51] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In B. Christianson, B. Crispo, and M. Roe, editors, *Security Protocols, 7th International Workshop Proceedings*, Lecture Notes in Computer Science, pages 172–182. Springer-Verlag Berlin Heidelberg, 1999.

[52] W. Stallings. *Network and Internetwork Security : Principles and Practice.* Prentice Hall, 2nd edition, 1995.

[53] A. Stubblefield, J. Ioannidis, and A. D. Rubin. Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. In *Network and Distributed System Security Symposium Conference Proceedings: 2002.* Internet Society, 2002.

[54] Sun Microsystems, Inc. JAVA^{TM} 2 PLATFORM, MICRO EDITION. Datasheet, 2001. `http://java.sun.com/j2me/j2me-ds-0201.pdf`.

[55] Texas Instruments, Inc. MSP430x13x, MSP430x14x Mixed Signal Microcontroller. Datasheet, 2001. `http://www-s.ti.com/sc/ds/msp430f149.pdf`.

[56] C.-K. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems.* Prentice Hall, 2002.

[57] University of California, Santa Barbara. Ad hoc On-Demand Distance Vector Routing. Home page. `http://moment.cs.ucsb.edu/AODV/aodv.html`.

[58] D. J. Wheeler and R. M. Needham. TEA, a tiny encryption algorithm. *Lecture Notes in Computer Science*, 1008:363–366, 1995.

[59] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 299–302. ACM Press, 2001.

[60] G. Yuval. Reinventing the travois: Encryption/MAC in 30 ROM bytes. In E. Biham, editor, *Fast Software Encryption: 4th International Workshop*, volume 1267 of *Lecture Notes in Computer Science*, pages 205–209. Springer-Verlag, 1997.

[61] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *Proc. 6th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'00)*, pages 275–283, 2000.

[62] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.