Forward Secure Communication in Wireless Sensor Networks

Sjouke Mauw[1], Ivo van Vessem[1], and Bert Bos[2]

[1] Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[2] Chess Information Technology BV, De Ronde 15B, P.O. Box 273, 5680 AG Best, The Netherlands

Abstract We propose a set of security provisions for node to base station communication in wireless sensor networks. It supports standard security requirements, viz. authentication of the origin of data and confidentiality of data. Additionally we use key evolution to achieve forward security which is of particular importance in the face of node capture attacks. As a bonus we obtain implicit weak freshness without message expansion. We take the typical resource constraints of wireless sensor networks into account. The security provisions can be superimposed on several communication models, such as the epidemic communication model.

1 Introduction

The main application of Wireless Sensor Networks is in monitoring the condition of a large area by distributing a collection of communicating sensors. Such networks are envisaged to consist of thousands of sensor nodes. In this scenario the budget per sensor is severely limited, which translates to a limited amount of chip area and a low capacity energy supply. This in turn puts tight restrictions on the amount of available storage, the affordable computational complexity, the amount of data that can be transmitted, and the transmission range.

Yet, there is an obvious need for authenticating sensor readings and some scenarios call for confidentiality as well. Consider, for example, the case of smoke detection sensors where an attacker could trigger false fire alarms if there were no means to establish the origin of a message. Sensors used for military purposes are also a prime target for adversarial manipulation. In a scenario where sensors monitor the presence of persons or perhaps their health condition, a confidential information exchange is required for privacy protection.

Our problem setting is characterised as follows. An area has to be monitored with respect to certain environment variables. For this purpose, a collection of sensors is distributed over the area. The readings of these sensors must be transmitted to a base station for further processing. We assume wireless communication and only a limited number of sensors in the range of the base station. Sensors have no location awareness and the network topology is unknown to both sensors and base station. We rely on an underlying communications scheme that

forwards messages such that with sufficiently high probability at some point the message reaches the base station.

We were originally inspired by a particular epidemic communication model [4] that uses a randomised swapping scheme in which neighbouring nodes engage in an exchange of messages held in a local cache. The scheme offers a high robustness but at the price of a considerable increase in the amount of data sent. We will not go into further details here since our proposed security provisions do not rely on specific properties of the underlying communication model and have relevance beyond the limited context of epidemic communication.

We do not place any unreasonable restrictions on the attacker. In particular, since sensors are deployed in a potentially hostile environment and the limited budget does not allow for tamper proof sensor nodes, we have to assume that the attacker is able to capture one or more sensors and extract key material. In such a scenario the best we can accomplish at the cryptographic level is a forward secure scheme such that a node capture does not compromise the security of prior messages. At another level the situation in which not all sensor readings can be trusted can perhaps be remedied by placing redundant sensors and assuming the number of captured sensors has a sufficiently low upper bound.

In this paper we strive to strike a balance between the resource constraints and the strong security requirements. We propose a scheme that achieves node to base station authentication with weak freshness and message confidentiality, yet has a modest computational complexity, minimal storage requirements, and quite acceptable message expansion. We do assume an invulnerable and reasonably powerful base station[3]. Our scheme was inspired by a scheme [14] that offers forward secure anonymity through key evolution in the context of Radio Frequency Identification. We have modified that scheme such that it can be used for message authentication, freshness, and confidentiality. Because we do not require anonymity of the nodes, the associated scalability problem disappears. We refer the reader to [2] for an extensive discussion of using key evolution to obtain forward security.

We remark that our confinement to (one-way) node to base station communication enables a solution that is economical in terms of the mentioned resource criteria. For example since we do not require in-network verification or processing of messages from other nodes, we do not need an expensive asymmetric scheme. Note also that a simple single shared network key would not provide much protection in the face of node captures and extraction of key material.

The remainder of this paper is organised as follows. Section 2 gives an overview of relevant existing work. Section 3 introduces our security provisions. Section 4 provides proofs for our claims of authenticity, confidentiality, freshness and forward security. Section 5 investigates implementation options. Section 6 analyses achievements and discusses remaining open problems.

[3] Note that we refer to the central trusted authority as *the* base station but of course it could equally well be distributed over multiple physical base stations provided they each satisfy the given requirements (i.e. reasonable resources, invulnerability).

Acknowledgements. Thanks are due to Berry Schoenmakers for his helpful comments on the verification and to Maarten van Steen for introducing us to the Gossip communication model.

2 Related Work

General sensor network security surveys include [15,6,11]. Closest to our work is [16], in particular SNEP, which is the part based around a single block cipher that is used for authentication, confidentiality, and random number generation. The key difference to note is that SNEP does not provide forward security. Although some empirical results [19] suggest that there is a notable performance penalty for using SHA-1 as our basic building block as opposed to RC5, the latter has the disadvantage that it is covered by a patent. Data from [18] indicates that the widely adopted AES block cipher, which would be a natural replacement for RC5, consumes at least 60% more energy per byte than SHA-1.

TinySec [9] is targeted at the security of node to node communication. The ability to detect in an early stage malicious messages and avoid committing resources to deliver these messages to the destination is used as motivation. We recognise the desirability of early detection possibilities, but failure to address replay attacks —which is justifiably omitted because the required maintenance of state for each communications partner is infeasible for memory limited sensor nodes— raises doubts on the usefulness of investing in authentication of node to node communication. Moreover, a key deployment mechanism is assumed of which the mentioned examples are either not robust (i.e. a single captured node compromises the entire security of the system) or require knowledge of the network topology.

We mention some studies that make further assumptions that are incompatible with the scenario we consider. Secure pebblenets [1] relies on a tamper resistance assumption to protect a shared secret key. SEKEN [7] requires the establishment and maintenance of the network topology by the base station. Sophisticated schemes such as those proposed in [17] require a bidirectional communication channel between node and base station and are clearly beyond what we consider feasible. Security provisions often require that nodes share cryptographic keys with each other. Key establishment mechanisms in sensor networks are considered in [21,12,8,3].

3 Specification

In our discussion we ignore the specifics of the underlying communication model. We use a very general message format as our starting point and define extensions that achieve node to base station authentication, confidentiality and weak freshness. We delegate the actual message sending and forwarding to the underlying communication model.

The constructions we suggest for authentication and confidentiality are a straightforward application of the well known symmetric key scheme. Our main

contribution lies in the introduction of a continuously evolving key that achieves forward security and, additionally, provides implicit freshness without the need to expand messages with e.g. counter values.

3.1 Definitions

Let N be the set of nodes in the network. Node $n \in N$ generates messages of the form (n, d) where d is an element of some unspecified data domain.

We assume the possibility of secure node initialisation, that is, the base station is able to securely distribute an initial key to the sensor nodes. More precisely, for each node $n \in N$ the base station chooses an initial key x_n^0 uniformly at random: $x_n^0 \in_R \{0,1\}^k$ for security parameter k and sends x_n^0 to node n such that x_n^0 is a shared secret between the base station and node n. As explained below, this key will be updated after each transmission of node n yielding a series of keys $x_n^0, x_n^1, x_n^2, \ldots$.

Let $\mathcal{H}$ be a non-invertible collision resistant hash function with domain restricted to k-bit strings: $\mathcal{H} : \{0,1\}^k \to \{0,1\}^k$.

We define $h : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^t$ to be a MAC function that on input of a k-bit key x and arbitrary d outputs a t-bit signature $h_x(d)$ on d. We require h to resist adaptive chosen-message attacks.

3.2 Authentication

We redefine the i-th message of node n to include a signature under ephemeral key x_n^i:

$$(n, d, h_{x_n^i}(d)) \tag{1}$$

Informally we argue that under the assumption that x_n^0 is unknown to the attacker and the fact that h offers key non-recoverability, an attacker is unable to learn any key $x_n^j, j \geq 0$ without physically tampering with the sensor node. Therefore an attacker will have a negligible probability of success in creating a valid signature for data other than data previously signed by n. We provide a more thorough security analysis in Section 4.

3.3 Key Evolution

We define for any $n \in N$ the i-th key x_n^i to be simply the image under $\mathcal{H}$ of the previous key x_n^{i-1}:

$$x_n^i = \mathcal{H}(x_n^{i-1}) \,,\; i > 0 \tag{2}$$

We assume nodes erase key x_n^{i-1} once the new key x_n^i has been determined and the erasure is such that we can assume that physical inspection cannot reveal erased keys.

3.4 Confidentiality

For confidentiality we introduce encryption function $E : \{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ that on input of a k-bit key x and arbitrary plain text d of length l outputs cipher text $E_x(d)$ also of length l. In our scenario it is particularly important that encryption does not result in data expansion as energy costs are dominated by transmission rather than computation [9,16,5]. For the moment we require E to resist adaptive chosen plain text attacks, implementation options and specific security properties will be explored in Section 5.

Messages that offer confidentiality in addition to authentication are now defined by:

$$(n, E_{x_n^i}(d), h_{x_n^i}(E_{x_n^i}(d))) \tag{3}$$

3.5 Freshness

Since our scenario does not exclude the possibility of message loss or delay in message delivery, the base station should anticipate non-sequential message receipt. This is typical behaviour in the epidemic communication model from [4].

We introduce an acceptance window of length $2w+1, w \geq 0$. Let X_n be the element of the hash chain $x_n^i, i \geq 0$ stored by the base station. Initially X_n is set to x_n^0. Upon receipt of a message (n, d, s) with $n \in N$ the base station searches for a $j, 0 \leq j \leq 2w$ such that $h_{\mathcal{H}^j(X_n)}(d) = s$. If it finds such a j it accepts d as authentic and fresh originating from node n.

The stored element of the hash chain is updated as follows.

$$X_n := \mathcal{H}^{j-w}(X_n) \quad \text{iff} \quad j > w \tag{4}$$

Hence apart from a short initialisation phase where the backward window is empty, $\mathcal{H}^w(X_n)$ is the maximal element of the chain $x_n^i, i \geq 0$ for which a message has been received. Once a message signed with an element further down the chain is received, the acceptance window is shifted forwards such that messages with an offset of at most $\pm w$ are accepted as fresh. Obviously the non-invertibility of $\mathcal{H}$ prevents us from storing $\mathcal{H}^j(X_n)$ rather than $\mathcal{H}^{j-w}(X_n)$.

Of course there is a trade-off to be made here, the base station can reserve $2wk$ additional bits per node to avoid having to compute up to $2w$ elements of the hash chain for each received message (n, d, s).

We will ignore the handling of duplicate messages here. It is trivial to record for which elements of the hash chain a message has been received such that duplicates can be ignored if so desired.

We remark that there is ample opportunity for refinement of the base station's policy for accepting messages. The base station could, for instance, dynamically change the value of w perhaps even for each node individually. It could use previous communication patterns and the current network traffic to determine an appropriate value. Should the base station detect prolonged disruption of communication it could temporarily increase the length of the forward window

or update the stored hash chain elements using previous patterns to avoid loss of synchronisation. These refinements are context dependent and will further be ignored here.

4 Verification

4.1 Signature Forgery

We define a signature forgery attack to be successful if the attacker without knowledge of key x succeeds in constructing some message (n, d, s) that is accepted by the base station as authentic and fresh originating from node n using key x, when no such signature under key x has been generated by node n.

We show that the probability of success of such a forgery is negligible in security parameters k and t. In our analysis we consider h to be a black box for which the probability of constructing a valid signature $s = h_x(d)$ for arbitrary data d and unknown but fixed key x is at most 2^{-t}, assuming $t \leq k$.

Note that for specific node $n \in N$ the set of acceptable keys is determined by $\{\mathcal{H}^i(X_n) \,|\, 0 \leq i \leq 2w\}$. It now follows easily that the probability of successful forgery is at most $2^{-t+\log(2w+1)}$.

As expected, the width of the acceptance window influences the probability of successful forgery. When determining an appropriate value for t also an upper bound on the value of w should be established such that the probability of successful forgery as determined above is sufficiently low. We refer to Section 5 for practical values for both t and w.

A notable difference with [14] is that an attacker is forced to choose a particular node $n \in N$ whereas [14] excludes node identifiers in order to preserve anonymity. This increases the probability of successful forgery by a factor $|N|$.

4.2 Confidentiality Violation

We define oracle $\mathcal{O}_C$ that on input of arbitrary plaintext d, node $n \in N$, and iteration number $i \geq 0$ outputs the cipher text under key x_n^i:

$$\mathcal{O}_C(d, n, i) = E_{x_n^i}(d) \tag{5}$$

We consider an attacker that has queried $\mathcal{O}_C$ at a number of inputs (d_j, n_j, i_j) that is polynomial in security parameter k but not at input (d, n, i). We define an attack on the confidentiality of $E_{x_n^i}(d)$ to be successful if the attacker is able to determine any information on d other than its length $|d|$.

First, we show by contradiction that the key that is input to E is unique. Assume $x_n^i = x_{n'}^j$ with $i < j$. Note that we do not require the keys to be element of the same hash chain. Then we have established x_n^{i-1} and $x_{n'}^{j-1}$ as colliding inputs to $\mathcal{H}$, contradicting its collision resistance.

If we consider E to be a Random Oracle and the input to E is unique, we have that $E_{x_n^i}(d)$ cannot be distinguished from a truly random $r \in_R \{0,1\}^{|d|}$ without knowledge of key x_n^i. Hence the probability distribution is uniform over all possible plaintexts from $\{0,1\}^{|d|}$.

4.3 Forward security

Suppose an attacker is able to physically extract or otherwise obtain at some point key x_n^j. We argue that knowledge of this key does not lead to a compromise of any previous messages.

We define oracle $\mathcal{O}_F$ that provides for given $n \in N$ and $i \geq 0$ key x_n^i:

$$\mathcal{O}_F(n, i) = x_n^i \tag{6}$$

We consider an attacker that has queried $\mathcal{O}_F$ at input (n, j) but not at any of the inputs $(n, i)\,, 0 \leq i < j$. We show that knowledge of the obtained key x_n^j does not help the attacker in forging signatures or deciphering messages under any key $x_n^i, 0 \leq i < j$.

Under the assumed non-invertibility of $\mathcal{H}$ key $x_n^{j-1} = \mathcal{H}^{-1}(x_n^j)$ cannot be determined from x_n^j in less than 2^k operations. Obviously we then have for any $x_n^i, 0 \leq i < j$ that it cannot be determined from x_n^j in less than 2^k operations since any such x_n^i would lead to $x_n^{j-1} = \mathcal{H}^{j-1-i}(x_n^i)$.

5 Implementation

So far our discussion has been on an abstract level and mainly focused on the security properties. In order to analyse the resource requirements we will now consider implementation options for abstract functions $\mathcal{H}$, h, and E as defined earlier.

5.1 Hash Function

Although recently it has been shown [20] that finding collisions for SHA-1 requires far less effort than the ideal strength of 2^{80} operations, we propose to take SHA-1 to implement $\mathcal{H}$. We note that the non-invertibility of SHA-1 is not affected and we have only used its collision resistance property in support of our confidentiality proof of Section 4.2. Even there, the inputs to $\mathcal{H}$ cannot be influenced by the attacker which makes the attack inapplicable.

The energy required to establish a new key as in equation (2) is about 15 μJ according to experimental results from [18] and assuming $k = 160$. The code size for a software implementation on various embedded processors is around 2000 byte based on data from [19] which is less than half the amount required for MD5. Presumably, in a hardware implementation several optimisations can be applied to reduce the required chip area, however the SHA-1 implementation is likely to be significant compared to the simple sensor node logic required for e.g. a smoke detecting sensor. It is our objective to reuse SHA-1 for functions h and E in order not to further increase code size/chip area.

5.2 MAC Function

We choose to use HMAC-SHA1-t as our implementation of MAC function h. Here parameter t is used as described in [10] to denote that only the first t output bits are used as the MAC.

We remark that the HMAC construction involves basically two applications of the hash function and some simple padding operations. The added complexity in terms of code size/chip area is therefore minimal. The amount of energy required to generate a signature is estimated by [18] to be around 1 μJ/byte.

5.3 Encryption Function

Similar to [14] we define $\mathcal{G}$ to be a non-invertible collision resistant hash function independent from $\mathcal{H}$ with domain restricted to k-bit strings: $\mathcal{G} : \{0,1\}^k \rightarrow \{0,1\}^k$. We use the hash values generated by $\mathcal{G}$ as a keystream. E can now be defined in terms of $\mathcal{G}$:

$$E_{x_n^i}(d) = d \oplus \mathcal{G}(x_n^i) \tag{7}$$

Where $\oplus$ denotes the bitwise exclusive-or operation that is understood to discard the remaining suffix of $\mathcal{G}(x_n^i)$ of length $k - |d|$ bits and operate on the first $|d|$ bits of $\mathcal{G}(x_n^i)$ only.

As a first observation note that the length of the plain text is now limited to k bits. We remark that for the scenarios we consider the value of k (to be determined shortly) is more than sufficient to accommodate typical message sizes. We could generate a longer keystream, for instance, by applying $\mathcal{G}$ to simple variations of x_n^i.

Earlier we showed uniqueness of key x_n^i. If we consider $\mathcal{G}$ to be a Random Oracle, we can view $\mathcal{G}(x_n^i)$ as a One Time Pad offering perfect secrecy. Non-invertibility of $\mathcal{G}$ also guarantees that a known plaintext attack in which an attacker obtains $\mathcal{G}(x_n^i)$ does not lead to a compromise of x_n^i.

Notice that the construction we suggest here relies rather heavily on the assumptions that we make on the properties of $\mathcal{G}$. Once we choose a concrete hash function to implement $\mathcal{G}$ that has some predictable bias, this bias would directly reveal (partial) information on the plain text.

An attacker knowing some plaintext can determine (a prefix of) keystream $\mathcal{G}(x_n^i)$ resulting in the compromise of subsequent keys if $\mathcal{G} = \mathcal{H}$. However from the point of chip area/code size it would be desirable if $\mathcal{G}$ and $\mathcal{H}$ could use the same concrete hash function. Therefore, we define $\mathcal{G}(x_n^i) = \mathcal{H}(x_n^i \oplus msk)$ for fixed and public $msk \neq 0$ of length k bits. Knowledge of $\mathcal{G}(x_n^i)$ does not allow an attacker to determine subsequent key $\mathcal{H}(x_n^i)$.

5.4 Choice of Security Parameter Sizes

Throughout we have used k to denote the size of the key and t to denote the size of the signature. As mentioned before the amount of message expansion is of particular importance in our scenario.

We adopt a keylength of $k = 160$ bit as suggested for a collision resistant hash function in [13]. Choosing a lower value does not seem to offer significant savings as the key is expanded to full block length when used as input to SHA-1.

The suggested lowerbound of $t = 64$ bit for the size of the MAC could add significant overhead for typical message sizes. As mentioned in [13] for specific applications this value could be reduced if the chance of successful forgery $2^{-t+\log(2w+1)}$ is acceptable, taking into account the total number of forgery attempts an attacker is able to perform during the lifetime of the sensor network and the implications of a single successful forgery.

6 Conclusions

We developed a set of security provisions for communication in wireless sensor networks which establishes authentication of the origin of data, confidentiality of data, forward security (implying a weak form of tamper resistance), and freshness (to mitigate the effect of maliciously delayed data).

Our security provisions do not protect against attacks at the physical level where an attacker directly manipulates sensors, for instance, by placing a heat source in close proximity of a temperature sensor or shielding a sensor from its environment. We are also not concerned with availability and do not prevent the attacker to learn information from traffic analysis.

We strove to minimise the resource requirements for the sensor nodes. The computational complexity is low due to our choice of using a hash function. In order to minimise the communication overhead, we chose for encryption algorithms without data expansion. Finally, by reusing already available logic (the hash function) chip area can be reduced.

Rather than a communication protocol, we developed a set of security provisions that can be superimposed on several underlying communication models for sensor networks. This made it possible to broaden the scope of our work which was initially targeted at adding security provisions to the epidemic model as described in [4].

The decision to secure the data transfer at the level of node to base station communication has several consequences. It allowed us to minimise the resource requirements for the nodes and to look for solutions without PKI or keys shared between nodes. Of course this comes at the price of an increased effort at the base station. As a drawback, we have that messages will only be verified at the base station and not during their transmission through the network. This could make a denial of service attack by inserting false messages more effective. In order to assure freshness, we have introduced an acceptance window at the base station. A drawback is that this will imply that in some cases benign messages will be ignored. In practice the size of the window will have to be tuned to the actual latency of the network.

There are several interesting options for future research. First of all, a simple protection against DoS attacks would be useful to strengthen our scheme. Next,

practical experiments are needed to validate that the degradation of the performance of the epidemic communication model stays within reasonable bounds when adding the security provisions. Finally, it would be interesting to study secure bi-directional communication between the nodes and the base station, making it possible to dynamically instruct the sensors.

References

1. Stefano Basagni, Kris Herrin, Danilo Bruschi, and Emilia Rosti. Secure pebblenets. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 156–163, New York, NY, USA, 2001. ACM Press.
2. Mihir Bellare and Bennet Yee. Forward-security in private-key cryptography. Cryptology ePrint Archive, Report 2001/035, 2001.
3. Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002. ACM Press.
4. Daniela Gavidia, Spyros Voulgaris, and Maarten van Steen. Epidemic-style monitoring in large-scale sensor networks. Technical Report IR-CS-012.05, Vrije Universiteit Amsterdam, March 2005. `http://www.cs.vu.nl/pub/papers/globe/IR-CS-012.05.pdf`.
5. Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *ASPLOS-IX: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*, pages 93–104, New York, NY, USA, 2000. ACM Press.
6. F. Hu and N. Sharma. Security considerations in ad hoc sensor networks. *Ad Hoc Networks*, 3(1):69–89, 2005.
7. Kamran Jamshaid and Loren Schwiebert. Seken (secure and efficient key exchange for sensor networks). In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pages 415–422, 2004.
8. K. Jones, A. Wadaa, S. Olariu, L. Wilson, and M. Eltoweissy. Towards a new paradigm for securing wireless sensor networks. In *NSPW '03: Proceedings of the 2003 workshop on New security paradigms*, pages 115–121, New York, NY, USA, 2003. ACM Press.
9. Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175, New York, NY, USA, 2004. ACM Press.
10. H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication, February 1997. RFC 2104.
11. Y. W. Law, S. Dulman, S. Etalle, and P. Havinga. Assessing security-critical energy-efficient sensor networks. Technical Report TR-CTIT-02-18, University of Twente, The Netherlands, July 2002. http://purl.org/utwente//38381.
12. Donggang Liu, Peng Ning, and Rongfang Li. Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(1):41–77, 2005.
13. Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.

14. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to "privacy-friendly" tags. In *RFID Privacy Workshop*, MIT, MA, USA, November 2003.
15. Adrian Perrig, John Stankovic, and David Wagner. Security in wireless sensor networks. *Commun. ACM*, 47(6):53–57, 2004.
16. Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. Spins: security protocols for sensor networks. *Wirel. Netw.*, 8(5):521–534, 2002.
17. Asad Amir Pirzada and Chris McDonald. Kerberos assisted authentication in mobile ad-hoc networks. In *CRPIT '04: Proceedings of the 27th conference on Australasian computer science*, pages 41–46, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
18. Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan, and Niraj K. Jha. Analyzing the energy consumption of security protocols. In *ISLPED '03: Proceedings of the 2003 international symposium on Low power electronics and design*, pages 30–35, New York, NY, USA, 2003. ACM Press.
19. Ramnath Venugopalan, Prasanth Ganesan, Pushkin Peddabachagari, Alexander Dean, Frank Mueller, and Mihail Sichitiu. Encryption overhead in embedded systems and sensor network nodes: modeling and analysis. In *CASES '03: Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems*, pages 188–197, New York, NY, USA, 2003. ACM Press.
20. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. Technical report, Shandong University, Shandong, China, June 2005.
21. Yongge Wang. Robust key establishment in sensor networks. *SIGMOD Rec.*, 33(1):14–19, 2004.