Preserving Source Privacy and Hop-by-Hop Message Authentication in Wireless Sensor Networks

Prathibha M Chandrakanth
Department of computer science
T. John Institute of Technology

Bindu Madavi P
Assistant Professor
Department of computer science
T. John Institute of Technology

***Abstract*— Authentication of message is the most effective way to prevent unauthorized and corrupted messages from being forwarded in wireless sensor networks (WSNs). There are many message authentication schemes developed are based on either symmetric-key cryptosystems or public-key cryptosystems. Most of these schemes have the limitations of high computational and communication overhead in addition to lack of scalability and resilience to node compromise attacks. A scheme was introduced based on degree of polynomial to overcome these issues. The weakness of this scheme is built-in threshold determined by the degree of the polynomial: when the number of messages transmitted is larger than this threshold, the degree of polynomial is recovered by the opponent. A scalable authentication scheme is proposed based on elliptic curve cryptography (ECC). While enabling intermediate nodes authentication, any node is allowed to transmit an unlimited number of messages without suffering the threshold problem in proposed scheme. The scheme can also provide message source privacy. The results are analyzed both theoretically and is simulated to demonstrate that our proposed scheme is more efficient than the polynomial-based approach in terms of computational and communication overhead under comparable security levels while providing message source privacy.**

Keywords*— *Hop-by-hop authentication, symmetric-key cryptosystem, public-key cryptosystem, source privacy, simulation, wireless sensor networks (WSNs)

I. INTRODUCTION

Authenticating a message plays a key role in preventing unauthorized and corrupted messages from being forwarded in networks to save sensor energy. Therefore, for this reason, many authentication schemes have been proposed to provide message authenticity and integrity verification for wireless sensor networks (WSNs) [1]–[5]. The schemes can be divided into two categories: public-key based approaches and symmetric-key based approaches.

The symmetric-key based approach requires complex key management, lacks of scalability, resiliency to large numbers of node compromise attacks since the message sender and the receiver have to share a secret key. The sender uses shared key to generate a message authentication code (MAC) for each transmitted message. The authenticity and integrity of the message is verified by the node with the shared key, which is shared by group of sensor nodes. An intruder can use the shared key by capturing a single node.

A secret polynomial based message authentication scheme was introduced in [3] to solve the scalability problem, where the threshold here is determined by the degree of the polynomial. When the messages transmitted is below the threshold, the nodes verify the authenticity of the message through a polynomial evaluation. When the messages transmitted is above the threshold, the polynomial is recovered and the system is completely broken.

The public-key based approach, every message is transmitted along with the digital signature of the message generated using the sender’s private key. Each intermediate forwarder and the final receiver can authenticate the message using the sender’s public key [7], [8]. The limitation of the publickey based approach is high computational overhead.

In this paper, we propose a secure and efficient source anonymous message authentication (SAMA) scheme which is based on modified ElGamal signature (MES) scheme on elliptic curves. The scheme enables the intermediate nodes to authenticate the message so that all corrupted message can be detected and dropped to conserve the sensor power. To achieve resiliency, flexibility in authentication and source identity protection, our scheme does not face the threshold problem. Our proposed scheme is more efficient than the polynomial-based algorithms under comparable security levels.

This is the first scheme that provides hop-by-hop node authentication without the threshold limitation, and has perform better than the symmetric-key based schemes. The nature of our algorithm makes the scheme suitable for decentralized networks.

II. ASSUMPTIONS AND GOALS

A. *Threat Model and Assumptions*

The wireless sensor networks consists of large number of sensor nodes. Each sensor node in sensor networks knows its relative location in the sensor domain and is capable of communicating with its neighboring nodes directly using geographic routing. The network is connected through multi-hop communications. There will be a security server (SS) which stores and distributes the security parameters among the network. This server will never be compromised. After deployment, the sensor nodes may be captured and compromised by attackers. After compromising, all information stored in the sensor nodes can be accessed by the attackers. The compromised nodes can be reprogrammed and fully controlled by the attackers. The compromised nodes captured by the attackers will not be able to create new public keys that can be accepted by the SS and other nodes.

The two types of attacks launched by the attackers are:

• *Passive attacks*: In passive attacks, the adversaries could eavesdrop on messages transmitted in the network and perform traffic analysis.
• *Active attacks*: Active attacks can only be launched from the compromised sensor nodes. Once the sensor nodes are compromised, the adversaries will obtain the information stored in the compromised nodes, including security parameters of the compromised nodes. The adversaries can then modify the contents of the messages, and inject their messages.

B. *Goals*

The proposed scheme aims at achieving following goals:

• *Message authentication:* The message receiver should be able to verify whether a received message is sent by the node that is claimed, or by a node in particular group. In other words, the attackers cannot pretend to be an innocent node and inject fake messages into the network without being detected.
• *Message integrity:* The message receiver should be able to verify whether the message has been modified en-route by the adversaries. In other words, the attackers cannot modify the message content without being detected.
• *Hop-by-hop message authentication:* Each forwarder on the routing path should be able to verify the authenticity and integrity of the messages upon reception.
• *Identity and location privacy:* The attackers cannot determine the message sender's ID and location by analyzing the message contents or the local traffic.
• *Node compromise resilience:* The proposed scheme should be resilient to node compromise attacks. Even though many nodes are compromised, the remaining nodes are still secure.

III. TERMINOLOGY AND PRELIMINARY

A. *Terminology*

Privacy is sometimes referred to as anonymity. Communication anonymity generally refers to the state of being unidentifiable within a set of subjects. This set is called the ambiguity set (AS). Sender anonymity means that a particular message is not linkable to any sender, and none of the message is linkable to a particular sender. Let us start with the definition of the unconditionally secure source anonymous message authentication scheme (SAMA).

Definition 1 (SAMA): The SAMA consists of the following two algorithms:
• Generate $(m,Q_1,Q_2, \ldots,Q_n)$: Given a message m and the public keys $Q_1,Q_2,\ldots,Q_n$ of the AS $S = \{A_1,A_2,\ldots,A_n\}$, the actual message sender A_t, $1 \leq t \leq n$, produces an anonymous message S(m) using its own private key d_t.

• Verify S(m): Given a message m and one anonymous message S(m), which includes the public keys of all members in the AS, a verifier can determine whether S(m) is generated by a member in the AS.

The security requirements of SAMA include:
• Sender ambiguity: The probability that verifier successfully determines the real sender of the anonymous message is exactly 1/n, where n is the total number of members in the AS.

•Unforgeability: An anonymous message is unforgeable if no adversary, given the public keys of all members of the AS and the anonymous messages $m_1,m_2,\ldots,m_n$ adaptively chosen by the adversary, can produce in polynomial time a new valid anonymous message with non-negligible probability.

Definition 2 (MES): A modified ElGamal signature scheme [17] consists of the following three algorithms:
Key generation algorithm: Let p be a large prime and g be agenerator of Z_p Both p and g are made public. For a random private key x € Z_p, the public key y is computed from $y = g^x \bmod p$.

Signature algorithm: The MES can also have many variants [18], [19]. For the purpose of efficiency, we will describe a variant, called optimal scheme. To sign a message m, one chooses random k € Z^*_{p-1}, then computes the exponentiation $r = g^k \bmod p$ and solves s from:

$$s = rxh(m, r) + k \bmod (p - 1), \qquad (1)$$

where h is a one-way hash function. The signature of this message m is defined as the pair (r, s).
Verification algorithm: The verifier checks whether the signature equation $g^s = ry^{rh(m,r)} \bmod p$. If the equality holds true, then the verifier accepts signature, and will reject otherwise.

B. *Proposed MES Scheme on Elliptic Curves*

Let p > 3 be an odd prime. An elliptic curve E is defined by an equation of the form:

$$E : y^2 = x^3 + ax + b \bmod p,$$

where a, b € Fp, and $4a^3 + 27b^2 \neq 0$ mod p. The set E(Fp) consists of all points (x, y) € Fp on the curve, together with a special point O, which is called point at infinity.

Let G = (x_G, y_G) be a base point on E(Fp) whose order is a very large value N. User A selects a random integer d_A € [1,N− 1] as his private key. Then, user can compute his public key Q_A from $Q_A = d_A \times G$.

Signature generation algorithm: For Alice to sign message m, she follows these steps:

1) Select a random integer k_A, $1 \leq k_A \leq N - 1$.
2) Calculate $r = x_A \bmod N$, where $(x_A, y_A) = k_AG$. If r = 0, go back to step 1.
3) Calculate $h_A \leftarrow h(m, r)$, where h is a cryptographic hash function, such as SHA-1, and ← denotes the l leftmost bits of the hash.
4) Calculate $s = rd_A\, h_A + k_A \bmod N$. If s = 0, go back to step 2.
5) The signature is the pair (r, s).

Signature verification algorithm: For Bob to authenticate Alice's signature, he must have a copy of public key Q_A, then he will:

1) Checks that $Q_A \neq O$, otherwise invalid
2) Checks that Q_A lies on the curve
3) Checks that $nQ_A = O$

After that, Bob follows these steps to verify the signature:

1) Verify that r and s are integers in [1,N − 1]. If not, the signature is invalid.
2) Calculate $h_A \leftarrow h(m, r)$, where h is the same function used in the signature generation.
3) Calculate $(x_1, x_2) = sG - rh_A\, Q_A \bmod N$.
4) The signature is valid if $r = x_1 \bmod N$, invalid otherwise.

C. *Proposed SAMA on Elliptic Curves*

Suppose that the message sender (say Alice) wishes to transmit a message m anonymously from her network node to any other nodes. The AS include n members, $A_1, A_2, \ldots, An$, e.g., $S = \{A_1, A_2, \ldots, An\}$, where the actual message sender Alice is A_t, for some value t, $1 \leq t \leq n$. In this paper, we will not distinguish between node Ai and its public key Q_i. Therefore, we also have $S = \{Q_1, Q_2, \ldots, Q_n\}$.

Authentication generation algorithm: Suppose m is a message to be transmitted. The private key of message sender Alice is d_t, $1 \leq t \leq N$. To generate an efficient SAMA for message m, Alice performs following three steps:

1) Select random and pairwise different k_i for each $1 \leq i \leq n-1$, $i \neq t$ and compute r_i from $(r_i, y_i) = k_iG$.
2) Choose a random k_i € Z_p and compute r_t from $(r_t, y_t) = k_tG - \sum_{i=t} r_ih_iQ_i$ such that $r_t \neq 0$ and $r_t \neq r_i$ for any $i \neq t$, where
$h_i \leftarrow h(m,r_i)$
3) Compute $s = k_t + \sum_{i \neq t} k_i + r_td_th_t \bmod N$

The SAMA of the message m is defined as follows:

$S(m) = (m, S, r_1, y_1, \ldots, r_n, y_n, s)$.

Verification algorithm: For Bob to verify an alleged SAMA $(m, S, r_1, y_1, \ldots, r_n, y_n, s)$, he must have a copy of the public keys $Q_1, \ldots, Q_n$. Then he will:

1) Checks that $Q_i \neq O$, i = 1, …., n, otherwise invalid
2) Checks that Q_i, i = 1, ….., n lies on the curve
3) Checks that $nQ_i = O$, i = 1,….. , n

After that, Bob follows these steps:

1) Verify that r_i, y_i, i = 1, …. , n and s are integers in [1,N − 1]. If not, the signature is invalid.
2) Calculate $h_i \leftarrow h(m, r_i)$, where h is the same function used in the signature generation.
3) Calculate $(x_0, y_0) = sG - \sum r_ih_iQ_i$
4) The signature is valid if first coordinate of $\sum_i (r_i,y_i)$ equals x_0, invalid otherwise.

In fact, if the SAMA has been correctly generated without being modified, then we compute the following:

$$(x_0, y_0) = sG - \sum_{i=1} r_ih_iQ_i$$

$$= (k_t + \sum_{i \neq t} k_i + r_td_th_t) - \sum_{i} r_ih_iQi$$

$$= \sum_{i \neq t} k_iG + (k_tG - \sum_{i \neq t} r_ih_iQ_i)$$

$$= \sum_{i \neq t} (r_i,y_i) + (r_t,y_t)$$

$$= \sum (r_i,y_i)$$

Therefore, the verifier should always accept SA.

IV. SOURCE PRIVACY AND AUTHENTICATION SERVER SELECTION

The message source node selects an AS from the public key list in the SS as its choice before a message is transmitted. This set should includes itself with some other nodes. When an attacker receives a message, he may possibly find the direction of the previous hop, or even the real node of previous hop. However, the attacker will not be able to distinguish whether the previous node is the actual source node or simply a forwarder node if the adversary is unable to monitor the traffic of the previous hop. Therefore, the selection of AS should create sufficient diversity so that it is infeasible for the adversary to find the message source based on the selection of the AS itself.

Some criteria for the selection of the Ambiguity Set can be described as follows:

• To provide message source privacy, the message source needs to select AS to include nodes from all directions of the source node. In particular, AS should include nodes from the opposite direction of the successor node. In this way, even an immediate successor node will not be able to distinguish the message source node from the forwarder based on the message that it receives.

• Though the message source node can select any node in the AS, some nodes in AS may not be able to add any ambiguity to the message source node. For instance, nodes that are apparently impossible or very unlikely to be included in the AS based on the geographic routing. Therefore, these nodes will not be appropriate candidates for the AS. They should be excluded from AS for energy efficiency.

• To balance the source privacy and efficiency, we should try to select nodes to be within a predefined distance range from the routing path. We recommend selecting an AS from the nodes in a band that covers the active routing path. However, the AS may not have to include all the nodes in the routing path.

• The AS may not have to include all nodes in that range, nor does it have to include all nodes in the active routing path. In fact, if all nodes are included in AS, then this may help the adversary to identity the possible routing path and find the source node.

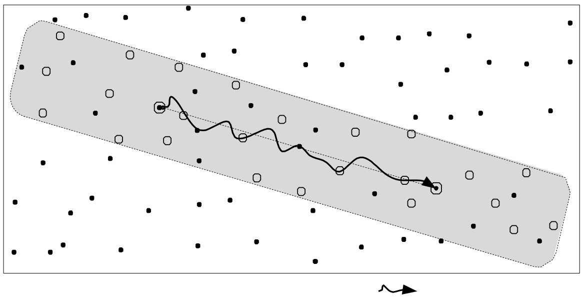


Fig. 1. Anonymous set selection in active routing

As an example, suppose we want to transmit a packet from source node S to destination node D in Fig. 1. We select AS to include only nodes marked with o , while nodes marked as • will not be included in the AS. Of all these nodes, some nodes are on the active routing path, while others are not. All these nodes will be located within the shaded band area surrounding the active routing path. Assume node A is compromised, unless node A collaborates with other nodes and can fully monitor the traffic of the source node S, then it will not be able to determine whether S is the source node, or simply a forwarder. This analysis is also true for other nodes.

Any node in the active routing path can verify the contents' authenticity and integrity. However, anyone who receives a packet in transmission can possibly exclude some of the nodes in the WSNs as the possible source node. Inclusion of these nodes in AS will not increase the source privacy. Nevertheless, more the nodes included in the AS are, higher the energy cost will be. Therefore, the selection of the AS must be done with care so that the energy cost and the source privacy can both be optimized.

In addition, balancing power consumption between authenticity and integrity verification, and the possibility that corrupted messages are being forwarded, the verification ser- vice may not have to take place in every hop; instead, it may be configured to take place in every other hop, for instance.

V. KEY MANAGEMENT AND COMPROMISED NODE DETECTION

In our scheme, we assume that there is an SS whose responsibilities include public-key storage and distribution in the WSNs. We assume that SS will never be compromised. After deployment, the sensor node may be captured and compromised by the attackers. Once compromised, all information stored in the sensor node will be accessible to the attackers. We further assume that the compromised node will not be able to create new public keys that can be accepted by the SS.

As a special scenario, we assume that all sensor information will be delivered to a sink node, which can be co-located with SS. As described in Section V, when a message is received by the sink node, the message source is hidden in the AS. Since the SAMA scheme guarantees that message integrity is untampered, when a bad or meaningless message is received by the sink node, the source node is assumed as compromised. If compromised source node only transmits one message, it would be very difficult for node to be identified without additional network traffic information. However, when a compromised node transmits more than one message, the sink node can narrow possible compromised nodes down to a very small set.

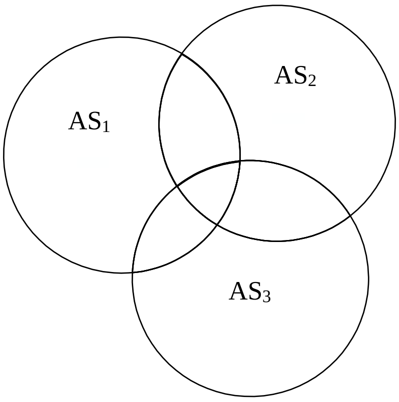


Fig. 2. Compromised node detection

As shown in Fig. 2, we use the circle to represent an AS. When one message is transmitted, the sink node can only obtain information that the source node will be in a set, say AS1. Therefore, if the sink node keeps tracking the compromised message, there is high probability that the compromised node can be isolated.

If the compromised nodes repeatedly use the same AS, it makes traffic analysis of compromised nodes feasible, which will increase the likelihood for the compromised nodes to be identified and captured. When any node identified as compromised, the SS can then remove its public key from its public key list. It can also broadcast node's short identity to the entire sensor domain so that any sensor node that uses the stored public key for an AS selection can update its key list. Once the public key of a node has been removed from the public key list, and/or broadcasted, any message with AS containing the compromised node should be dropped without any process in order to save the precious sensor power.

VI. PERFORMANCE ANALYSIS

Our proposed authentication scheme is evaluated through both theoretical analysis and simulation demonstrations. We will compare the proposed scheme with the bivariate polynomial-based symmetric-key scheme described in [3], [4]. A fair comparison between our proposed scheme and the scheme proposed in [4] should be performed with n = 1.

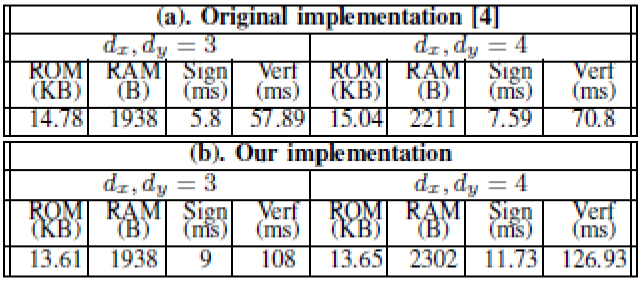
Most of the authentication schemes are based on symmetric key schemes, including the polynomial evaluation based threshold authentication scheme. The secret bivariate polynomial is defined as [3]:

$$f(x, y) = \sum_{i=0}^{d_x} \sum_{j=0}^{d_y} A_{i,j}\, x^{i} y^{j}$$

where each coefficient $A_{x,y}$ is an element of a finite field F_p , and d_x and d_y are the degrees of this polynomial. d_x and d_y are also related to message length and the computational complexity of this scheme.

Each SAMA contains an AS of n randomly selected nodes that dynamically changes for each message. For n = 1, our scheme can provide at least the same security as the bivariate polynomial-based scheme. For $n > 1$, we can provide extra source privacy benefits. Even if one message is corrupted, other messages transmitted in the network can still be secure. Therefore, n can be much smaller than parameters d_x and d_y.

TABLE 1
PERFORMANCE COMPARISON OF THE BIVARIATE POLYNOMIAL BASED SCHEME IN TWO DIFFERENT SCENARIOS: (a) THE ORIGINAL IMPLEMENTATION UNDER 8MHZ, AND (b) OUR IMPLEMENTATION UNDER 4MHZ



(a). Original implementation [4]							
$d_x, d_y = 3$				$d_x, d_y = 4$			
ROM (KB)	RAM (B)	Sign (ms)	Verf (ms)	ROM (KB)	RAM (B)	Sign (ms)	Verf (ms)
14.78	1938	5.8	57.89	15.04	2211	7.59	70.8

(b). Our implementation							
$d_x, d_y = 3$				$d_x, d_y = 4$			
ROM (KB)	RAM (B)	Sign (ms)	Verf (ms)	ROM (KB)	RAM (B)	Sign (ms)	Verf (ms)
13.61	1938	9	108	13.65	2302	11.73	126.93

From the table, we have the following findings:

- For bivariate polynomial-based scheme, the authentication generation time is much longer than the verifying time; while for our proposed scheme, the verifying time is about half of the authentication generation time, except when n = 1, the generation time is shorter than the verification time.

- Comparing bivariate polynomial-based scheme with our proposed scheme for n = 1, we find that the generation time of our scheme is less than 5% of the bivariate polynomial-based scheme for all dx, dy, but verifying time is slightly longer when dx, dy is less than 100. When dx and dy is longer than 150, the verifying times of both schemes are comparable.

- The memory consumption of proposed scheme is slightly less than the bivariate polynomial-based scheme in all scenarios.

- For proposed scheme, to provide source privacy, the cost of generation time and time required for verifying increase linearly with n.

VII. CONCLUSION

In this paper, we first proposed a novel and efficient source anonymous message authentication scheme (SAMA) based on elliptic curve cryptography (ECC). While ensuring message sender privacy, SAMA can be applied on any message to provide message content authenticity. To provide every hop message authentication without the weakness of the built-in threshold of the polynomial-based scheme, we then propose a node by node message authentication scheme based on the SAMA. When applied to Wireless Sensor Networks with fixed sink nodes, we discussed possible techniques for compromised node identification.

REFERENCES

[1] F. Ye, H. Lou, S. Lu, and L. Zhang, “Statistical en-route filtering of injected false data in sensor networks,” in IEEE INFOCOM, March 2004.

[2] S. Zhu, S. Setia, S. Jajodia, and P. Ning, “An interleaved hop-by-hop authentication scheme for filtering false data in sensor networks,” in IEEE Symposium on Security and Privacy, 2004.

[3] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly-secure key distribution for dynamic conferences,”

in Advances in Cryptology - Crypto’92, ser. Lecture Notes in Computer Science volume 740,1992, pp. 471-486.

[4] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Communications. of the Assoc. of Comp. Mach., vol. 21, no. 2, pp. 120–126, 1978.

[5] T. A. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms,” IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469–472, 1985.

[6] H. Wang, S. Sheng, C. Tan, and Q. Li, “Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control,” in IEEE ICDCS, Beijing, China, 2008, pp. 11–18.

[7] D. Pointcheval and J. Stern, “Security proofs for signature schemes,” in Advances in Cryptology - EUROCRYPT, ser. Lecture Notes in Computer Science Volume 1070, 1996, pp. 387–398.

[8] D. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” Communications of the ACM, vol. 24, no. 2, pp. 84–88, February 1981.

[9] “The dinning cryptographer problem: Unconditional sender and recipient untraceability,” Journal of Cryptology, vol. 1, no. 1, pp. 65–75,1988.

[10] A. Pfitzmann and M. Hansen, “Anonymity, unlinkability, unobservabil- ity, pseudonymity, and identity management a proposal for terminol- ogy,” http://dud.inf.tu-dresden.de/literatur/Anon Terminology v0.31.pdf, Feb. 15 2008.

[11] A. Pfitzmann and M. Waidner, “Networks without user observability–design options.” in Advances in Cryptology - EUROCRYPT, ser. Lecture Notes in Computer Science Volume 219, 1985, pp. 245–253.

[12] M. Reiter and A. Rubin, “Crowds: anonymity for web transaction,” ACM Transactions on Information and System Security, vol. 1, no. 1, pp. 66–92,1998.

[13] M. Waidner, “Unconditional sender and recipient untraceability in spite of active attacks,” in Advances in Cryptology - EUROCRYPT, ser. Lecture Notes in Computer Science Volume 434, 1989, pp. 302–319.

[14] D. Pointcheval and J. Stern, “Security arguments for digital signatures and blind signatures,” Journal of Cryptology, vol. 13, no. 3, pp. 361–396, 2000.

[15] L. Harn and Y. Xu, “Design of generalized ElGamal type digital signature schemes based on discret logarithm,” Electronics Letters, vol. 30, no. 24, pp. 2025–2026, 1994.

[16] K. Nyberg and R. A. Rueppel, “Message recovery for signature schemes based on the discrete logarithm problem,” in Advances in Cryptology - EUROCRYPT, ser. Lecture Notes in Computer Science Volume 950, 1995, pp. 182–193.

[17] R. Rivest, A. Shamir, and Y. Tauman, “How to leak a secret,” in Advances in Cryptology–ASIACRYPT, ser. Lecture Notes in Computer Science, vol 2248/2001. Springer Berlin / Heidelberg, 2001.

[18] M. Bellare and P. Rogaway, “Random oracles are practical: A paradigm for designing efficient protocols,” in CCS’93, 1993, pp. 62–73.

[19] BlueKrypt, “Cryptographic key length recommendation,” http://www.keylength.com/en/3/.

[20] W. Zhang, N. Subramanian, and G. Wang, “Lightweight and compromise resilient message authentication in sensor networks,” in IEEE INFOCOM, Phoenix, AZ., April 15-17 2008.

[21] A. Perrig, R. Canetti, J. Tygar, and D. Song, “Efficient authentication and signing of multicast streams over lossy channels,” in IEEE Symposium on Security and Privacy, May 2000.

[5] A. Perrig, R. Canetti, J. Tygar, and D. Song, “Efficient authentication and signing of multicast streams over lossy channels,” in IEEE Symposium on Security and Privacy, May 2000.

[6] M. Albrecht, C. Gentry, S. Halevi, and J. Katz, “Attacking crypto-