

Classify Encrypted Data in Wireless Sensor Networks

Yongdong Wu, Di Ma, Tiejian Li and Robert H. Deng
Institute for Infocomm Research
21, Heng Mui Keng Terrace,
Singapore, 119613
Email: {wydong, madi, litieyan, deng}@i2r.a-star.edu.sg

Abstract—End-to-end security mechanisms like SSL [1], which are popular on Internet, may seriously limit the capability of In-network processing that is the most critical function in sensor network. Since supporting In-network processing can significantly improve the performance of extremely resource-constraint sensor networks featuring many-to-one traffic pattern. It is an open problem of how to protect the traffics and to support In-network processing at the same time. This paper tackles the problem by proposing a model of categorizing encrypted messages in wireless sensor networks. A classifier, an intermediate sensor node in our setting, is embedded with a set of searching keywords in encrypted format. Upon receiving an encrypted message, it matches the message with the keywords and then processes the message based on certain policies such as forwarding the original message to the next hop, updating it and forwarding or simply dropping it on detecting duplicates. The messages are encrypted before being sent out and decrypted only at its destination. Although the intermediate classifiers can categorize the messages, they learn nothing about the encrypted messages except several encrypted keywords, even the statistic information. The presented scheme is efficient, flexible and resource saving. The performance analysis shows that the computational cost and communication cost are minimized. Furthermore, it is resilient to node capture attack and many other kinds of attacks. We are prototyping the model on our MICA2 [2] mote testbed.

I. INTRODUCTION

Wireless appliances will be dominant in the near future. Many kinds of wireless technique have been used or proposed, such as GSM, GPRS, 3G, Bluetooth and 802.11b. Of the whole wireless family, one emerging wireless standard, 802.15.4 [5], is currently being proposed, which is foreseen as the new specification for wireless sensor network. Wireless sensor network distinguishes itself from other traditional wireless networks by relying on extremely constrained resources like energy, bandwidth and capabilities of processing and storing data. Among them, power efficiency is the most critical consideration since sensors are typically deployed in remote area for a long time. For instance, sensors are installed on the endangered animals so that they can be tracked in a large reserved area.

While sensors are scattered around some area, collecting useful information from those sensors becomes difficult since there is generally no fixed plotted network covering the area. Data collected by individual sensor has to be passed to other sensors. And one by one, until finally it is passed down to its destination. In actual deployment, there are also some more

powerful sensors that can act as the fusion points to aggregate the collected data. The In-network processing [6], [8], [9] can effectively reduce large amount of duplicated data in sensor network. The processed data are then sent out to the base station where events can be properly processed. Fig. 1 depicts a typical sensor network.

However, In-network processing, on the other hand, can not protect the data well in a hostile environment where a sensor can be captured by adversaries. How to protect the data and at the same time to support In-network processing, is an open problem. One immediate reference is to provide end-to-end security like SSL [1] on Internet. Unfortunately, this solution is not applicable due to two reasons: firstly, the *many-to-one* traffic pattern of sensor network makes it difficult to pre-load one-to-one keys between two sensors, and to refresh the keys, *etc.*. Without a scalable keying mechanism, we can not encrypt the message at one end and decrypt it at the other end. Secondly, the method simply blocks any In-network processing and may cause the network inefficient with heavy traffic, which is actually a kind of DDoS attacks. Another possible way to protect the message is to enable *point-to-point* security like the link layer encryption scheme [3]. This method is not efficient by decrypting and re-encrypting the message at every sensor. It also suffers from node capture attack where a single compromised node can pollute the whole network.

We tackle the problem with a classifier model inspired by a class of “Searching on encrypted data” methods [10], [11], [12], [15]. In our model, different classifiers are installed on deploying sensors taking different roles. The classifier can be considered as a search engine that can match certain encrypted keywords to the encrypted messages. Messages are therefore categorized and processed based on some pre-defined policies. We design a secure scheme that makes the classifier, while categorizing the messages, obtains neither the knowledge of messages nor the statistical information on keywords. The present classifier is very efficient in terms of sensor’s computational and communication costs. Moreover, it enables error detection. The model is being tested on MICA2 [2].

Paper organization: Section 2 introduces the sensor network model. We elaborate our classifying scheme in section 3. Then in section 4, we analyze the performance. Section 5 lists the related works. At last, we summarize the paper.

II. SENSOR NETWORK MODEL

In Fig. 1, we depict a typical sensor network scenario. The hierarchical architecture has three levels where the base station, fusion point and sensor node stay respectively. From the full powered base station to ultimately limited sensor nodes, the levels are separated depending on the energy, radio bandwidth, processing power, and so on. Being queried, the sensor will act on collecting raw data and generate reading reports. The sensor reading is then sent out toward the fusion point. The fusion point can aggregation the readings and send the data upward to some base station. The path of data flowing relies further on the routing protocol and the network topology.

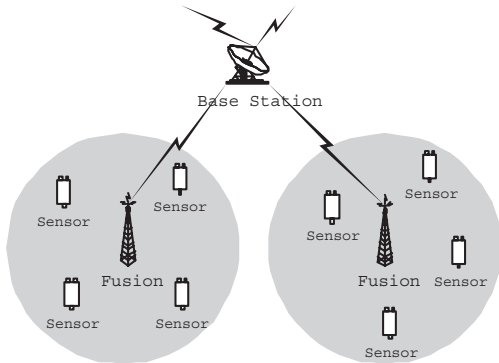


Fig. 1. A typical sensor network scenario. A base station, located centrally, is able to contact many fusion points. Each fusion point can interact locally with several neighboring sensors. The radio range of a sensor is very limited.

A. Sensor constraints and security assumption

A wireless sensor is extremely resource constrained, e.g. a Berkeley mote, MICA2 [2], has an 8MHz 8-bits Atmel AT-MEGA128L CPU with 128kB of memory, 4kB of RAM and 512kB flash memory. It features 19.2kbps wireless bandwidth on a single shared channel and with up to hundred meters range. With this setting, public key cryptography is generally considered inappropriate, e.g. RSA, with 1024-bit keys, takes 43ms to sign and 0.6ms to verify on a 200MHz Intel Pentium Pro, and takes much longer time (i.e. seconds on verifying) on an 8MHz 8-bits processor. However, software based symmetric ciphers are properly used in sensor nodes [3].

Without loss of generality, in our model we assume that a sensor, pre-loaded with certain keys, can process symmetric cryptography (either software or hardware implemented) with acceptable overhead. Any public key algorithm is not considered applicable on sensor nodes. Although most of the fusion points are more powerful sensor nodes, we assume generally that they do not perform any public key operation. Base stations are more powerful as they have both wired and wireless interface and can connect with each other via public key based secure end-to-end communication. But they can not communicate with sensors by this way. Wired connection and public key cryptography are out of the scope of this paper. Therefore, the sensor network is assumed to perform symmetric encryption/decryption only.

B. Threat model

Wireless sensor networks are more vulnerable to attacks that are more difficult from being launched in wired network [7]. Using open medium, an adversary can easily eavesdrop on, intercept, inject and alter the data transmitted. All these attacks can be dealt with by encrypting and authenticating the messages. Our scheme ensures the message authenticity, integrity and confidentiality by default.

The adversary is assumed to own more resources such as powerful processors and expensive radio bandwidth than sensors. Equipped with rich resources, the adversary can launch even more serious attacks. For example, the resource consumption attack: the adversary can jam a sensor node by repeatedly sending packets to it, the sensor will soon run out of battery and can not use its radio bandwidth. Moreover, the node compromise attack: when sensor nodes are plotted in remote area, the adversary can take control of the sensor, recover its secrets and send fake messages to the network. We do not address these threats and upper layer attacks [4] in our model.

III. CLASSIFIER STRUCTURE

The classifier in our scheme is actually a searching engine embedded with certain encrypted keywords. It can be considered as a software function. On input an encrypted message, the classifier matches it with those keywords sequentially, and outputs (*yes, no*) individually. Based on the pre-defined policy, post-processing follows. In Fig. 2, the classifiers are installed onto sensors at different levels of sensor network. Since base stations are responsible for higher layer applications such as various events, the classifiers are deployed at the fusion and sensor level.

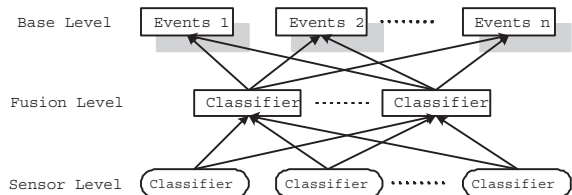


Fig. 2. Hierarchical model for classifying encrypted messages. Data are collected at the sensors and passed upward to the classifiers at either sensor level or fusion level, until gathered at base station. Thus, various events can be processed.

A. Classifying method

Suppose a sensor is about to send an encrypted message M to the base station with keywords W_1, W_2, \dots, W_n . The classifier, knowing a label L_i for a message folder, will collect the messages containing a keyword W_i , but learns nothing else. With respect to Figure 2, the sensors send the messages to the classifier, the classifier will categorize the messages based on the keywords and send the messages to next hops or other classifiers for further processing.

1) *Preparing messages on sensors*: In case a sensor wants to send a message M to the center, it calculates $c = E_K(M)$, where K is an encryption key known to the sensor. Then it encapsulates the message as

$$M' = c \parallel H(c, H(W_1)) \parallel \cdots \parallel H(c, H(W_n)). \quad (1)$$

where $H(\cdot)$ is a one-way function (e.g. MD5 [16]), and $E_K(M)$ is the encrypted message which can be decrypted by the intended recipient. The encryption function $E(\cdot)$ is a symmetry cipher (e.g. AES).

2) *Classifying messages on classifier*: Upon receiving the encapsulated message M' , the classifier will match the keywords sequentially. Specifically, if the classifier has the label $L_i = H(W_i)$, it categories the encrypted message according to $H(W_i)$.

B. Full classifying scheme

Above we describe the basic scheme in which only sensors and classifiers are involved. Remind in Fig. 2, we know that base station is the real event generator and processor. We now describe the full scheme which is a dynamic event-driven scheme.

An event can be any function that is performed in the sensor network. For example, the sensor application is to find those temperature higher than certain threshold. An event in this application needs to broadcast a request to the network to ask for sensor readings. The network then responses with those interested reading. In our scheme, an event sends out the request like this: $Req[s, H(s, W_1), \cdots, H(s, W_n)]$, where s is a unique sequence number, W_1, W_2, \cdots, W_n are the interested attributes (keywords). s is used to ensure that a random set of encrypted searching keywords are generated from the same set of keywords. The request is broadcasted to the whole network. Those sensors acting as a classifier store the keywords that present the classification rule for this event. They then collect and filter messages. The sensors that are not interested to be a classifier act as ordinary sensors. This scheme makes a sensor select its role dynamically. Moreover, even a classifier is dynamically driven by events and can filter messages according to different classification rules.

Following the basic scheme, if a sensor wants to send a message M responding to the request, it generates $c = E_K(M)$, where K is an individual encryption key shared by the sensor and the base station. The encapsulated message is generated by equation(2).

$$M' = c \parallel H(c, H(s, W_1)) \parallel \cdots \parallel H(c, H(s, W_n)). \quad (2)$$

Upon receiving the encapsulated message M' , the classifier will match the event's keywords sequentially as above. Suppose M' meets the required attributes, the classifier may response with $Rep[M']$.

C. Security analysis

First of all, without the key K , the encrypted messages can not be decrypted. The scheme thus protect the message confidentiality. Additionally, in Equation(1, 2), $c = E_K(M)$ is incorporated into the hash values. The benefit is that the hash values are different even when the message M is invariable. Thus, an attacker can not obtain the statistics of the messages from the sensors. Similarly, the classifier does not know the statistics. On the other hand, the classifier can not obtain any information on the keywords from the hash values. In case of requiring stronger security, $H(r, W_i)$ can be replaced with $H(K, r, W_i)$. However, this scheme can only be used in one-to-one communication which is unpopular in sensor network.

IV. PERFORMANCE ANALYSIS

A. Communication overhead

If there are n interested keywords, the communication overhead is nh with the present scheme, while it's nH with SPKE [11] (refer to section V), where h (or H) is the size of hash value (digital signature resp.) and $h \ll H$. Because the overhead of SSKE [10] is linear with the number of the words, and SSKE can not cooperate with the compression technology (e.g. the English language is estimated to be 75% redundant [17]), the communication traffic of SSKE is much heavier than that of the present scheme.

B. Computational cost

If there are n interested keywords, the computational cost overhead is nt with the present scheme, nT with SPKE, where t (or T) is the time for computing hash value (digital signature resp.). Usually, $T > 10^4t$ given the same security strength. The computational cost of SSKE is roughly the same as the present scheme.

C. Error detection

Because the wireless network is error-prone, the encapsulated message may be tampered in the transmission path. In this case, no keywords will be found, and the classifier will block the message or send a request for re-transmission. However, the schemes in [11] [12] did not care about message integrity and overlooked the errors.

V. RELATED WORKS

Our scheme is inspired by a class of approaches on "searching on encrypted data". Here, we give a review on existing searching methods.

A first work, by Song *et al.* [10], studied the problem of searching on data encrypted using SSKE (Searchable Symmetric Key Encryption) scheme. In the basic scheme of SSKE, if a ciphertext is decrypted into a plaintext which meets a standard format, the keyword is found. However the basic scheme is vulnerable to known-plaintext attack. Another weakness of SSKE is that any word is padded to be of fixed length such that the length of the message is increased apparently. To reduce the size of the encrypted messages, SSKE stores the length field before each word in the file, and glue the length field

and word together as one word. However, the searching time of this method is linear with the number of message bits, other than the number of words. Therefore, SSKE increases both the communication overhead and processing time.

Boneh *et al.* [11] proposed a SPKE (Searchable Public Key Encryption) scheme using a public key system. SPKE enables the classifier to test whether a keyword exists in an encrypted message. Waters *et al.* [12] used SPKE to build an encrypted and searchable audit log. As mentioned above, because public key system requires high resource consumption, both schemes are not suitable for wireless sensor networks.

One privacy-preserving scheme [13] studied how to build a decision tree classifier from two separated private databases. The proposed classifier does not compromise owner's privacy with the help of an untrusted third-party server. It requires the cooperation between two parties, as well as involving a third party.

Bellovin and Cheswick [14] proposed a searching scheme based on Bloom filters and Pohlig-Hellman encryption. A semi-trusted third party can transform one party's search queries to a form suitable for querying the other party's database, in such a way that neither the third party nor the database owner can see the original query. The computation cost of this third-party scheme is heavy too.

Recently, Goh [15] proposed secure index for searching encrypted data. The scheme builds a bloom filter based on different words for each document. To ensure the privacy of the document, the owner employed a keyed hash function to fill the bloom filter. Without the key known only to the owner, it can not be searched by any other parties including the database.

Other approaches [18], [19], [20] also studied related issues. [18] extend the searching scheme over conjunctive Keywords. [19] proposed a Database-Service-Provider Model where SQL queries can be executed over encrypted Database. Similarly, [20] proposed anti-tamper databases.

VI. CONCLUSION AND ONGOING WORKS

This paper proposes a classifier for categorizing encrypted messages which are attached with a serial of searching keywords. The classifier can classify messages correctly but gain no information from the messages. The present classifier is very efficient in terms of computational cost and communication overhead, and effective in error detection. However, we take as default the key shared by two parties. In fact, an efficient keying mechanisms is equally important and needs to be studied further to fit into our scheme. Furthermore, the keyword based searching scheme does not support well on searching numeric data, especially for calculating the numbers. We are looking into the approaches such as [21] for suitable solutions.

REFERENCES

- [1] OpenSSL <http://www.openssl.org>
- [2] Jason Hill, et al. "System architecture directions from networked sensors". In proceedings of ACM ASPLOS IX, pps. 93-104. Nov. 2000.

- [3] Chris Karlof, Naveen Sastry, David Wagner. "TinySec: A link layer security architecture for wireless sensor network". Unpublished Manuscript.
- [4] Chris Karlof and David Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", First IEEE International-Workshop on Sensor Network Protocols and Applications, May 2003
- [5] Wireless medium access control and physical layer specifications for low-rate wireless personal area networks. IEEE Standard, 802.15.4-2003, May 2003.
- [6] Alec Woo, Terence Tong, and David Culler "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks", The First ACM Conference on Embedded Networked Sensor Systems (SenSys2003).
- [7] Adrian Perrig, John Stankovic and David Wagner "Security in wireless sensor networks". Communications of the ACM, 47(6), June 2004, Special Issue on Wireless sensor networks, pp.53-57.
- [8] L. Hu and D. Evans. "Secure aggregation for wireless networks". In Workshop on Security and Assurance in Ad hoc Networks. Jan. 2003.
- [9] B. Przydatek, D. Song, and A. Perrig. "SIA: Secure Information Aggregation in Sensor Networks". In Proc. of ACM SenSys 2003.
- [10] D. Song, D. Wagner, and A. Perrig. "Practical Techniques for Searches on Encrypted Data," IEEE symposium on Security and Privacy, pp. 44-55, 2000
- [11] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," to appear in Eurocrypt 2004. <http://eprint.iacr.org/2003/195>
- [12] B. Waters, D. Balfanz, G. Durfee, D. Smetters, "Building an Encrypted and Searchable Audit Log," The 11th Annual Network and Distributed System Security Symposium (NDSS) 2004
- [13] Wenliang Du, Zhijun Zhan, "Building Decision Tree Classifier on Private Data," IEEE International Conference on Data Mining, 2002, <http://www.cis.syr.edu/~wedu/Research/publication.html>
- [14] Steven M. Bellovin and William R. Cheswick, "Privacy-Enhanced Searches Using Encrypted Bloom Filters," <http://eprint.iacr.org/2004/022/>
- [15] Eu-Jin Goh, "Building Secure Indexes for Searching Efficiently on Encrypted Compressed Data," <http://eprint.iacr.org/2003/216>.
- [16] R. Rivest, "The MD5 Message Digest Algorithm," RFC 1321, 1992
- [17] Doug Stinson, "Cryptography Theory and Practice", CRC Press, pp.61-62, 1995
- [18] Philippe Golle and Jessica Staddon and Brent Waters, "Secure Conjunctive Keyword Search over Encrypted Data," Applied Cryptography and Network Security (ACNS), 2004
- [19] Hakan Hacigümüş, Bala Iyer, Chen Li, Sharad Mehrotra, "Executing SQL over Encrypted Data in the Database-Service-Provider Model," ACM SIGMOD'2002, pp.216-227
- [20] S. C. Gultekin Ozsoyoglu, David Singer, "Anti-tamper databases: Querying encrypted databases," Annual IFIP WG 11.3 Working Conference on Database and Applications Security, 2003.
- [21] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, Yirong Xu, "Order Preserving Encryption for Numeric Data," ACM SIGMOD 2004.