In-Network Data & Secure Communication Processing In Sensor Nodes

Kawsar Mohiudin [1], Dr. Bilal A. Lone [2]
[1] Research Scholar, Department of Computer Science, OPJS University, Churu Rajasthan, India.
[2] Department of Electronics & IT, Sopore Infotech NIELIT Accredited Hardware Establishment J&K, India

Abstract - In-network data processing in sensor nodes is a fastly emerging research topic. The distributed processing could have several advantages for sensor nodes. This computation is typically much less energy consuming than communication and in-network processing enables WSN to provide more complex services to application layer, and not only data gathering functionality. However, in addition to computational overhead, in-network data processing introduces also many challenging security issues. Most of them are still open and require development of innovative security systems. In this article we survey the present research related to security of in-network data processing in sensor nodes and highlight the directions, which are most promising in our opinion. In this work we present a way that can be applied in sensor nodes in order to provide secure and authenticated in-network processing. It utilizes nodes which are responsible for data assembling and command diffusion. The proposed model is simple and scalable and exhibits resiliency against node capture and replication as compromised nodes cannot be used to calculate and eventually take over the network. It also allows for dynamic addition of new nodes and eviction of compromised ones by requiring minimum involvement of the base station. Finally, it is designed so that the majority of the sensors have to store only two keys plus a hash value, minimizing the storage capacity needed for the method to operate.

Index Terms - Encryption, Sensor Nodes, Assembler, Distribution Keys, Layers, Diffusion, In-Networks.

I Introduction

Sensor nodes introduce several technical challenges. Some of the most important issues are related to extracting useful, reliable and timely information from the deployed sensor network, and include distributed information processing, data fusion and security. In-network data processing is called as a processing the network nodes carry out on data in a distributed fashion. The process is performed while data is exchanged, i.e. before it is acquired and used by the higher layers. Such distributed processing could have several advantages for sensor nodes. First of all, in WSN, computation is typically much less energy consuming than communication. In-network processing enables sensor networks to provide services, and not just data, and to fully exploit its multitude and redundancy properties [1]. Therefore in-network data processing, such as data fusion and assembling, has emerged in the recent years as an active research area in WSNs. One of the important issues related to in-network data processing is to look for a realistic balance between computational overhead, delay, data resolution, and data trustworthiness. Therefore, in-network data processing in sensor nodes requires development of new secure and energy-efficient methods enabling fusion of relative large amount of data. In our survey we focus on security and trust aspects of in-network data processing. Sensor networks have attracted much scientific interest during the past few years. These networks use hundreds to thousands of inexpensive devices over an area for the pur-pose of monitoring certain phenomena and capture distinct measurements over a long period of time. The large scale of sensor networks and the limited resources

of sensor nodes in terms of computation power, memory, communication, and most importantly, energy make it an important challenge to design and develop efficient information processing and ag-gregation techniques.

II Authentic Processing of Data by Sensor Nodes

Most of the ongoing research activities in WSN focus on the lesser layers, namely radio communication, routing and self-organization. But first on top of these layers the real capabilities of WSN are unleashed. Indeed, WSN can be seen not just as data source, but rather as a provider of services tightly related to data assembling and processing. It is clear that WSNs must assemble data, but the real added value lays in the fact that the multitude of nodes can also process this data [2]. This processing can range from mixing, assembling, and simple mathematical operations such as averaging to more complex reasoning on data. The in-network processing layer represents one of the real strengths of WSNs it can deliver not just raw data but also process it in real-time with no intervention of the infrastructure and leverage on redundancy to cope with data loss and corruption. In-network data processing can also be used to reduce the amount of information transmitted, as data coming from multiple sensors is usually highly correlated, and thus to achieve a longer network lifetime. However appealing and powerful, in-network processing requires a high level of security tampering with data at this level can indeed introduce risks that range from simple unauthorized data access to malicious data modification. The first line of defense against these risks are cryptographic systems integrity and confidentiality can be achieved using cryptographic schemes. Nodes can be captured and their secret material can be disclosed to an attacker. To overcome this problem, techniques are proposed, that exploit end-to-end encryption in conjunction with particular key distribution systems, homomorphic encryption schemes or public cryptographic schemes. Nevertheless, in order to make sensor networks economically viable, sensor devices are limited in their energy, computation and communication capabilities; hence these schemes have to take into account the technical and economic constraints [3]. Once data has been sensed and possibly processed by nodes, it must be delivered to data sinks, i.e. nodes that are responsible for gathering data and passing it to application gateways. Application gateways are nodes that are responsible to deliver data to the real point of exploitation. The sink nodes and gateways can super constitute single point of failure, given that they are normally much less numerous than wireless nodes. Sinks and gateways, being the eventual destination of data, are also the perfect point of attack in order to get access to data, modify it or supply false data. Countermeasures to such attacks include mutual authentication of the gateway and the nodes, ensuring that both the gateway is entitled to receive data and that the data is sent by legitimate nodes [4]. Higher layers of a WSN deployment include a middleware layer and an application layer. In addition to providing important functional service, WSN can be seen also as a source of external risks to the middleware and applications, as network layer can be used as means to attack them. Since the sensors are now communicating data over smaller distances in the clustered environment, the energy spent in the network is much lesser than the energy spent when every sensor communicates directly to the information processing center. Further benefits of assembling processing include scalability as assembling nodes can form multi-level hierarchies, and increased life-time as assembling processing reduces the volume of data exchanged and hence the overall energy spent for communication. A closely related form of in-network processing is data diffusion, in which the network hierarchy is used in the reverse direction in order to diffuse control messages from the central server towards the assemblers and eventually towards the sensor nodes. For example, in tracking applications the sensor network must be used in both modes first to assemble sensed data about the movement of the tracked object and then to diffuse commands to nearby sensors to enable further tracking.

III Ensuring Security in the Sensor System

All security methods in sensor networks must satisfy certain requirements in order for sensor nodes to be able to exchange data securely. The bare minimum consists of pro-viding confidentiality, authentication, integrity and fresh-ness. However, making secure communications between sensor nodes becomes a challenging task. We present new security systems that can be used to provide secure in-network processing in sensor nodes. In particular, this means that we design the security systems with both assembling and diffusion in mind. Secure assembling implies that data is forwarded from the sensors in a secure and authenticated way. Thus an adversary cannot issue false data into the network unless of course a particular sensor node has been compromised. Secure diffusion requires that lesser level nodes are able to authenticate commands is-sued by their parents in the hierarchy. For both directions, protection is also provided against eavesdropping and tampering of data [5]. Our method is simple and scalable and most importantly offers resiliency against node capture and replication as compromised nodes cannot be used to calculate and eventually take over the network. In this model, a sensor network consists of a large num-ber of sensors distributed over an area of interest. There is a base station in charge of the network's mission. The network also consists of super-nodes, called as assemblers in addition to the sensor nodes [6]. We assume that the network is partitioned into distinct clusters and that each cluster is composed of an assembler and a set of sensor nodes (distinct from other sets), which gather information and transmit it to the assembler of their cluster. The assembler fuses the data from the different sen-sors, performs mission-related data processing, and sends it to the base station.

IV. Securing the In-Network Data Processing Model

In this section we present a security method for secure in-network processing. We break this discussion into multi-ple subsections to ease the readability and understanding of the method. Although most of the research in the security of in-network data processing is quite recent, it has produced many promising results. As we already discussed, a topic that has attracted particular attention is homomorphism of cryptographic functions such as encryption or signatures. These PH techniques provide foundations for adding security to in-network data processing. The security goals of in-network data processing are mainly confidentiality, integrity and authentication of data origin [7]. The operations involved in in-network data processing range from concatenation of data and mathematical operations (mainly addition and multiplication) to operations on set. Hidden data assembling techniques aim at processing sensor data while protecting confidentiality of both raw data and intermediate results. Once keys are shared between nodes, data is decrypted, processed, encrypted and sent on toward a sink. In the end-to-end scenario, a network wise key needs to be established between the sink and all the sensor nodes. This can be achieved using a master key or a public-key based solution. The additional security with respect to the hop-by-hop scenario is that all the sensors have the network key, but assemblers do not. Once the key is established, one of the privacy homomorphism can be used to assemble the data. The network is modeled as a tree rooted in a sink. Nodes are organized in layers. The advantage of this scheme is that if nodes are compromised, it is still impossible to disclose partial information because there are always k layers of encryption to protect the data. The drawback of this approach is that it heavily relies on the key pre distribution phase and a very rigid network topology. In another end-to-end technique for secure data assembling is presented [8]. This scheme is similar to the one presented before the homomorphic function used to perform the computation is again a simple variation of a one-time pad scheme and the network model is a tree rooted at the sink. In the previous scheme, a group of nodes belonging to a given layer shared a set of keys; in turn, in this scheme, each node shares with the sink a single, distinct, long-term key; with this key, each node generates the key stream used to encrypt its data. Every node can then assemble encrypted data thanks to the homomorphic properties of the encryption scheme. Even if a node is compromised and its secret material revealed, the secrecy of other nodes' data and of temporary results is preserved, since it is still protected by the encryption performed by other nodes.

V. Removing the Compromised Nodes from Sensor Network

Before we discuss a system for dynamically insert-ing new nodes into the network, we need a scheme to evict compromised nodes and revoke their corresponding keys. Since we are not dealing with intrusion detection in this work, we assume the existence of a detection system that informs the base station or the assemblers about compromised nodes [9]. We further assume that a sensor node cannot be compromised during the setup time of our system, either during the initial setup phase, or dur-ing the new node addition phase. This is a valid assumption, since the time needed for the setup phase is small in com-parison to the time needed for the node to be captured. If a node is compromised, the attacker cannot insert duplicates of that node in groups other than the group it origi-nated from. This is the case, since no other assembler will be able to compute which is essential for making the secure communication channel. Therefore, it suffices to provide a system for node revocations transmitted by the base station to be authenticated and eventually for aggregators to revoke nodes within their groups. As in the case of authenticating commands issued by the assemblers, we will base the revocation scheme on the use of one-way hash key chains. To use this scheme, we assume that sensor nodes are loosely synchronized and each aggre-gator is preloaded before deployment with the first key of the chain. The base station then discloses the keys in the key chain periodically in order reverse to the generation of these keys. Such use of the key chain allows the base station to broadcast authenticated messages to all assembler nodes. When assemblers receive such lists of compromised nodes, they verify the authenticity of the messages and then evict the nodes from the cluster by making a new group key with the rest of the nodes using the method. At the end of this procedure, only the nodes that are not in the revocation list will obtain the new group key while compromised nodes cannot be used to insert or retrieve information that is transferred within the group. In the case of compromised assemblers, we consider the entire cluster to be compromised as assemblers hold pair wise keys with all the cluster nodes. However, the damage is confined to the assembler's group of sensors as the compromised assembler cannot impersonate any other ones. In such a case, the parent nodes in the cluster hi-erarchy can establish new group keys with the rest of the sensors.

VI. The In-Network Data Assembling and Detection of Failure Nodes

The compromised nodes represent a big risk to the security of in-network data processing. The challenge arises from the fact that sensor nodes often need to be low-cost to justify their deployment, which makes it very hard to satisfy tamper-resistance requirements. An attacker could gain control over a sensor node in a stealthy way in order to generate faulty data or to alter the data processing. Thus, once a node is compromised, the secret material contained within is completely exposed and usable by the attacker. In order to cope with such risk, a few trust frameworks have been proposed in the literature to detect bogus sensor data. This implies a trust evaluation of sensor data at acquisition and assembling time trust refers to the reliability and accuracy of sensed information and it is related to the quality of the delivered sensor data. Within a WSN, sensor nodes are prone to different kind of failures, such as crash, omission, timing, value and arbitrary failures. Crash and omission imply no response from the sensor to the data query [10]. Timing refers to timeout during the processing a request. Value failure deals with delivering incorrect value due to malfunctioning or compromised sensor nodes. Finally, arbitrary failures include all the types of failures that cannot be classified in previously described categories. In sensor node failure detection, we identify self-diagnosis and group detection approaches. With self-diagnosis, each nodes detect its own failure, e.g., based on battery exhaustion. In group detection, each node in the same area is supposed to deliver a similar information. A good example is temperature measurement in a room. Let us assume a WSN application to measure the temperature in a room taking the average value provided by different thermometers in the same room makes it possible to resist attacks and to produce sensor data which is potentially more trustworthy as the number of contributors increases.

VII. Resource Requirements

In this section we analyze the computational, communi-cation and storage requirements for our key establishment method. The individual key of each sensor node is pre-computed and does not involve any processing or transmis-sion overhead. The cost for making a group key in our method is the same as updating the group key in the clus-ter, thus we only analyze the cost for making the group key. We must emhasize at this point that our method is independent of the underlying routing method used. Each sensor node has to store in memory its individual key, the present group key and the present value of the One Way Hash Chain. Assuming that the each of the above is 10 bytes long (80 bits), each sensor has to store only 30 bytes of information. The assembler node needs to store the key of each sensor node in its cluster (d in total), at most n values of the OWHC and its individual key, needed for communication with nodes higher in the hierarchy (for example with the Base Station).

VIII. Conclusions and Future Work

In-network data processing can potentially bring important benefits to sensor nodes. Computation is typically much less energy consuming than communication, so the additional computational overhead can be well justified by the reduced data transfer. The distributed processing, and possibility of assembling or even partial reasoning about sensed data, could also enable WSNs to provide more complex services to application layer, and not only data gathering functionality. However before the concept of such intelligent sensor network becomes the reality, many technical challenges have to be addressed. In particular, we need to design and implement secure, yet very efficient and cost-effective, data assembling systems. Very promising results have been recently achieved in this area. Our proposal accommodates both secure assembling of data so that they are forwarded to the base station, as well as secure dis-semination of commands issued by the base station to all sensors in the network. The systems presented in this work also support the addition of new nodes to the network, a critical requirement in sensor networks, as sensors have limited energy and thus limited life expectancy. Additionally, given the existence of an intrusion detection system, our method allows for the eviction of compromised nodes. We must also note that the successful application of our proposal neither de-pends on the existence of location information or on the underlying routing method, nor does its performance vary depending on the density of the network, like probabilis-tic schemes. The proposed method scales very efficiently as it is based on forming clusters in a hierarchical fashion, which is the most efficient architecture when networks be-come very large. Finally, the number of keys that each sen-sor node has to store is small, only three for regular sensors, and the method does not require any computations round the clock.

REFERENCES

[1] V. Przydatek, S. Song and V. Perrig, "Secure Information Assembling in Sensor Networks," in ACM SensSys, 2003.

[2] B. Basagni, H. Kim, D. Vomi, and S. Rila, "Se-cure pebblenet," in Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing, MobiHoc 2004, pp. 256–263, October 2004.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor Proceed-ings of the 18th ACM conference on Computer and communications security, pp. 243–297, 2008.

[4] H.Chan, A.Perrig, and D.Song, "Random key predistribution schemes for sensor networks," in IEEE Sym-posium on Security and Privacy, pp. 197–213, Can 2003.

[5] R. Liu and D. Ning, "Making pairwise keys in distributed sensor networks," in Proceedings of the 10th ACM Conference on Computer and Communication Security, pp. 252–261, October 2004.

[6] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in Proceedings of the 10th ACM con-ference on Computer and communication security, pp. 42–51, October 2003.

[7] S. Zhu, S. Setia, and S. Jajodia, "LEAP efficient security systems for large-scale distributed sensor networks," in Proceedings of the 10th ACM Conference on Computer and Communication Security, pp. 62–72, October 2003.

[8] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In Proceedings of the 7th International Conference on Information Security (ISC), pages 281-283.

[9] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks A survey. Communication Magazine, April 2009.

[10] B. Jung, and G. S. Sukhatme," Tracking targets using multiple robots The effect of environment occlusion", Autonomous Robots, Vol. 13, No. 3, pp 191- 205, 2002.