17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

Container Communities: Anti-tampering Wireless Sensor Network for Global Cargo Security

Jiang Bian, Remzi Seker and Srini Ramaswamy
Department of Computer Science
University of Arkansas at Little Rock
2801 S. University Avenue, Little Rock, Arkansas 72204
Email: {jxbian, rxseker, srini}@ualr.edu

Nuri Yilmazer
Department of Electrical Engineering and Computer Science
Texas A&M University - Kingsville
700 University Blvd. MSC 192, Room 305 Kingsville, TX 78363
Email: nuri.yilmazer@tamuk.edu

***Abstract*—U.S. Immigration and Customs Enforcement is overwhelmed with the number of containers entering U.S.A. on a regular basis. Although containers are pre-screened and inspected at the time of shipment, it does not necessarily address all security risks. Containers stay en-route for long enough time that their contents can be tempered with or altered according to the procedures/needs of a terrorist attack. Moreover, considering the huge amount of shipments entering U.S. board daily, it is not practical to inspect every container again upon arriving U.S. ports. There is a urgent need to develop a protection system, such that the integrity of the containers can be persevered or at least the intrusion can be detected and properly handled. This paper proposes a design of a comprehensive solution that would monitor containers' integrity from the originating port to the destination port and report any intrusion event if it has taken place so the intruded container can be handled in an appropriate way. More important, the system itself needs to be secure and intrusion resistant. In the proposed system, intrusion detection sensors are deployed on each container, and form a Wireless Sensor Network (WSN) to report intrusion incidents. A secure and reliable communication protocol has been developed to ensure not only the integrity but also the authenticity of the communication conducted among among sensor nodes. With in the proposed system, not only the intrusions can be identified and properly handled, but also the system itself is attack resistant by utilizing Wireless Sensor Networks in a smart fashion.**

I. Introduction

According to U.S. Customs and Border Protection (CBP), about 90% of the world's trade is transported in cargo containers. About half of the containers are carried by vessels. Right after 9/11, the CBP launched the Container Security Initiative (CSI) that intends to increase security for container cargoes shipped to the U.S.A. As described by CBP, it's goal is to "extend the zone of security outward so that American borders are the last line of defense, not the first." One of its task is to identify and locate vulnerable containers that pose a terrorist threat. One of its core element is to pre-screen the containers at the port of departure, which does a really good job of preventing high risk containers from boarding. However, there is no efficient method in place that can protect the containers from being tampered en-route. Ever increasing global trade and its reliance on security of shipping industry necessitates that security of containers is maintained throughout their transit. Repeated hijacking of ships off the coast of Somalia in 2008 has clearly shown that anything can happen to a container on a ship en-route to its destination. Even at a port, after the containers have been pre-screened, it is not certain that containers' contents have not been accessed. Goods can be stolen or in a worst case scenario, a weapon of some sort can be placed in any container. Screening the arriving containers at their destination port incurs asymmetric costs (both time and money) on the importing party. Also, it is impractical to inspect every container at the destination port. Therefore, the integrity of the containers should be monitored during transportation, after they leave their origin.

The goal of this paper is to propose a system that can be used to track and secure the contents of containers. Nevertheless, such a protection system should be attack-hardened. Containers should be smart enough to identify an intrusion event, generate alerts, and log the events. The warnings along with the location of the tampered container can then be sent to port and freight companies, so that the intrusion can be properly handled. Intrusion detection can be archived with a tamper sensor (e.g. detect light, opening door, etc). However, considering the environment around the containers and the ships (i.e. in the middle of the ocean), the radio communication may not operate well due to bad weather and can cause black-outs in communication. Moreover, a well-educated attacker can block the alerts and wipe the security log of the container(s) he broke into. In such a case, an attack is likely to go undetected.

We propose the Container Community Wireless Sensor Network (CC-WSN), where the smart containers on a vessel are wirelessly linked, so that they can cooperatively monitor one another's integrity. This is analogous to neighborhood watch. When an intrusion is detected, the tampered container can log and report this event to nearby containers. In a CC-WSN, containers exchange their status to increase the level of information redundancy to make it difficult for an attacker to cover tracks. The CC-WSN containers will actively query one another for their status. If no response is received from a particular container, C_i, then there is a good chance that C_i is compromised. Furthermore, communication among containers should utilize good authentication mechanisms as wireless signals can be easily captured by anyone with a receiver. By analyzing the communication among containers, it is possible for an attacker to impersonate a container and perform a

978-1-4244-4685-8/09/$25.00 ©2009 IEEE

man-in-the-middle attack. Particularly, in CC-WSN, when an intrusion is detected, the alert message is split into n segments using an Information Dispersal Algorithm (IDA), and delivered to different nearby nodes. To an outside observer, each message segment will appear to be random and not meaningful data, however, the original message can be reconstructed by an authorized personnel. Under such settings, impersonation attacks can be prevented. It is also worthwhile to note that the CC-WSN does not have as strict power limitations as classical WSNs as, since one can afford to equip a larger power source in a container due to its size. Moreover, in the scenario of preventing terrorist attacks, lives may depend on the timeliness and correctness of the alert messages obtained from the container sensors. As a result, the routing in CC-WSNs need to be secured. There are intrusion-tolerant routing mechanisms which can be applied to further secure the CC-WSN.

The rest of this paper is organized into four sections. Section II presents the concept of the WSNs, approaches currently used to address WSN security issues as well as the concept of the Information Dispersal Algorithm. In Section III, we introduce the overall design of the CC-WSN system as well as our concerns in terms of security and overall performance. We conclude our work with the importance of developing a CC-WSN system, where the integrity of the containers is of importance. Finally, additional improvements to improve the security of container communities are discussed in Section V.

II. Related Work

A. Wireless Sensor Network

Wireless Sensor Networks (WSN) [1] [2] is an emerging technology and its application areas have grown dramatically, ranging from military battlefields to home security applications [3] [4]. A WSN normally contains several distributed sensor nodes that can monitor the environmental conditions, such as temperature, sound, pressure, motion, etc., and a base station. Sensor nodes are spatially distributed in the region of interest to track and monitor the changes in the environment. Each sensor node in the Wireless Sensor Network is capable of sensing, processing of signals, and transmitting/receiving information. The information will eventually be routed to the base station where limited power and computational resources are not as scarce. Depending on the application, data needed to be processed is varied, and sometime it may be very large.

In the case of our CC-WSN, every container will be considered as one sensor node making up the CC-WSN, regardless of how many sensors a container may have internally. Moreover, an assumption has been made that containers are equipped with large enough power source (e.g. battery) and we will not have the power limitations similar to those of classical WSNs.

There are various ways to detect an intrusion to a container. For example, a motion sensor can detect the vibration when the container door opens; a sound sensor can identify abnormal noises when a container is being physically damaged, etc. However, the mechanics of what sensors and how many of them to use is beyond the scope of this paper. The goal of CC-WSN is to protect the data gathered when an intrusion is detected, logged in the CC-WSN, and if possible, deliver this warning quickly to the base station, the control room of the ship, so that the intrusion can be relayed to appropriate authorities via the communications facilities of the vessel and properly handled.

Because of the unique characteristics of the CC-WSN and the environment surrounding the cargo containers, design considerations include wireless coverage area, battery life, ability to cope with node failures, ability to withstand harsh environmental conditions, communication failures, etc. Standard shipping containers are 20, 40 or 45 feet long and made entirely of metal. Wireless signals reflect from metal surfaces, which will dictate placement of the antenna outside the container. The communication between the sensor(s) and the wireless unit are hard wired. Because the sensors are powered by batteries, battery life certainly needs to be a consideration in the design. A fundamental compromise between power consumption and functionality must be reached. Low power consumption allows for a less expensive implementation. However, increased wireless range, data transfer speed, and sensor capability demand more power. There must be enough power to help tracking the integrity of CC-WSN throughout its journey from the port of origin to the port of destination. This requires larger, more expensive batteries and higher maintenance costs involving more frequent recharges and replacement. However, in our case, cargo containers are fairly large and there is enough space to place a large battery inside a container.

B. Security in Wireless Sensor Network

Wireless communication among the sensor nodes face a wide spectrum of threats, including eavesdropping, spoofing, impersonation, and denial-of-service (DOS) attacks. Moreover, the sensor nodes are highly resource limited, which undermines use of strong encryption and authentication at each individual sensor node. Therefore, a sensor device usually cannot use public-key algorithms due to its insufficient resources. The symmetric key encryption and cryptographic hash functions are relatively cheaper and faster but standard commercial-strength algorithms of these types are not practical for typical WSNs.

Secure peer-to-peer communication can be useful for a lot of applications, such as the communication between two soldiers in a battlefield. However, as aforementioned, traditional pairwise key agreement such as Diffie-Hellman and other public-key schemes are not well-suited due to power and computational constraints. Therefore, some pairwise key agreements for WSN rely on pre-distribution [5] [6]. SPINS [7] contains two sets of protocols: SNEP for data confidentiality, two-way data authentication and data freshness; while μTESLA provides efficient broadcast authentication. However, the system assumes the presence of a base station to act as a gateway for all the other sensor nodes; this gateway is also the one point of failure for the whole system. Some other key management related research in WSN include [8] [9], etc.

In [10] [11], a one-way hash key chain is used to ensure the authenticity of the packets broadcast from the base station. First, the base station uses a one-way function $h()$ to generate a sequence of keys k_0, k_1,...,k_n, such that $k_i = h(k_{i+1})$. k_0 and hash function $h()$ are pre-distributed to every node. In the first broadcast round, the base station use k_1 to sign its packet, and the child nodes can verify the signature by comparing $h(k_1)$ with the known k_0. Since $h()$ is a one-way hash function, there is no way, an adversary can compute k_{i+1} from k_i. The authenticity is ensured, since the base station is the only one who knows k_{i+1} at the ith round of communication. Moreover, even if a node is compromised by an attacker, k_i is useless for next round of communication. However, based on number of communication rounds needed, this method may require the base station to compute a long chain of keys for pre-distribution, which will dramatically increase the setup time. Considering that an international cargo ship may travel for weeks, it is not a good idea to adopt such method for authentication, since the length of the key chain required is hard to be pre-determined.

C. Information Dispersal Algorithm

The Information Dispersal Algorithm (IDA) was first introduced by Michael O. Rabin [12] to design a fault-tolerant and transmission efficient information storage systems. The IDA is actually a special use of *erasure codes* a.k.a. *forward error correction* (FEC) codes. The most well-known erasure code is the one used in RAID level 5, known as the parity driver. In this system, there are at least 3 disk drives, the first two store different data but the third drive stores the XOR value of the data on the other two drives. Under this setting, given any two of the drives, one can recover all the data stored on the other drive. The basic idea of erasure codes is to add redundant data (error correction codes), in addition to the original data before transmission, and this extra information allows the receiver or reader to detect and correct data errors without the need to ask the sender to resend.

Reed-Solomon (R-S) codes [13] was introduced in 1960 as a class of error-correcting codes, which are mostly used in CD/DVD storage devices to recover the data after scratch as well as in communication protocols such as DSL and CDMA to reduce or eliminate the effect of packet loss. R-S codes have been suggested to be slow compared to some of its descendants, such as Gallager's LDPC-codes [14], Turbo-codes [15], Tornado-codes [16], LT-codes [17], Raptor-codes [18], etc. Although, both LDPC-codes and Turbo-codes are theoretically very fast, near Shannon limit [19], LDPCs have been neglected by researches because of its complexity (especially its software implementation), while Turbo-codes are avoided because of patent issues. Raptor-codes are closely related to LT-codes, since both of them are Fountain Codes. A Fountain Code produces a potentially limitless stream of output symbols for a given set of m input symbols. The decoder can recover the original m symbols from any set of k output symbols with high probability. Such codes are best suitable for data transmission over networks, where the receiver can cut off the communication as soon as it collects enough segments to reconstruct the original data.

III. Design of Container Communities

The concept of CC-WSN is to enable containers, each of which correspond to a node in an WSN, collaborate with each other to tackle the intrusion problem. When an intrusion occurs, if it is logged on the compromised node alone, the adversary may have the ability to modify the logs, even wipe off the memory, or use many other available techniques to cover his/her tracks and conceal the fact that intrusion has taken place. In such a case, the alert messages will never be viewed by the investigator so appropriate action can be taken. Therefore, in our CC-WSN, nodes cooperate to render such attacks ineffective.

A. IDA in CC-WSN

When an intrusion is detected, the intrusion event message will be split into n (i.e. based on the setting of the whole system) slices using an Information Dispersal Algorithm and delivered to n nearest neighbors along with the Message Identifier (MID) and Node Identifier (NID). When the ship arrives the destination port, an investigator can easily reconstruct the original message using the corresponding IDA decoder. Since the message slices are transmitted through wireless communications, some of the message slices may not be usable due to various reasons. For instance, the receiver node maybe facing a device failure during transmission, or simply, the packets can be lost due to a weak wireless signal, etc. However, the original message can be easily restored as long as there are k (i.e. based on the setting of the IDA) nodes still holding the message slices. For example, let us assume that n is 10 and k is 3. There is little chance that all 10 nearby nodes are failed to receive the message or compromised by the attacker all at the same time. As long as we have at least 3 nodes survives the attack, we can reconstruct the original message sent by the node when intrusion was underway.

Question may be raised that it may be best to just send out the original messages 10 times to 10 different nodes, so that the reconstruction phase associated with use of IDA can be eliminated. However, despite the fact that one can add more resources into a container, we think it is crucial to keep the message size small to increase its chance of delivery. Splitting the original message into 10 pieces, render the size of each slice smaller than the original message even considering the redundant content added by the IDA. The smaller the size, the higher the chance that a message can be transmitted successfully when an intrusion occurs. When one considers number of containers that may be placed on a large vessel, shorter message sizes will result in better overall efficiency in the CC-WSN's operation. Some diversity enhancing techniques can be incorporated into the CC-WSN to mitigate the fading effects that will occur in the wireless environment due to other objects and containers between the transmitter and the receiver specifically speaking between the two nodes. Since we do not have the strict battery problem

for this specific application, we can take advantage of using some of the traditional wireless communication techniques to provide a better and accurate performance.

Another advantage of using an IDA in such a case is that the IDA can be somehow seen as an encryption mechanism. Since the wireless communications are conducted through radio waves in open air, it is relatively easier to be interfered by the adversaries. An attacker can easily perform a man-in-the-middle attack, if the information is not encrypted. When using plain-text messages, the attacker may block the warning message and replace it with a message that indicates normal activity. However, when using an IDA, the resulting message slices are not necessarily meaningful to a human. Moreover, any change made to a message slice during transit or storage will cause it to fail the validation test performed by the IDA decoder.

The software implementation of an IDA is often considered to be computationally expensive. However, there are plenty of fast hardware implementations with very low cost, which can be added to the container node. The usage of an IDA in CC-WSN is first to introduce redundancy to prevent data loss from transmission failures. But also, it acts as an encryption mechanism to protect the communications. Although IDA increases the overall size of the data being transmitted and stored, it reduces the length of each packet sent to different nodes. A smaller packet size in wireless communication means higher success transmission rate, which eventually increases the possibility of data recovery.

B. Active Status Polling

It is possible that the sensor node in the compromised container is totally isolated by the attacker, since the wireless signals can be easily interfered in open air. In CC-WSN, the nodes not only send out alert messages when intrusions are detected, but also actively query/update each other on their status. If a node is not responsive to another node's polling message, it is highly likely that the polled node has been compromised. The polling node will raise a warning event associated with the polled node, and then report the event. By doing so, potential intrusions will be quickly identified, even if the attacker is able to block the communications of victim node.

If one were to represent messages issued when an intrusion has taken place with M_B, the frequency of occurrence of M_B is much smaller than that of the rest of messages. Therefore, for the sake of simplicity, the response from the polled node should not be delivered using the IDA method mentioned previously since the response is meant for the inquiring node and peer-to-peer approach is the most direct and simplistic approach to accomplish the status check. However, for security reasons, not only secrecy of message contents but also authenticity of messages must be ensured. While public key crypto-systems are commonly believed to be inefficient to use on resource constraint devices, Watro et. al. have developed a lightweight security system for wireless sensor networks named TinyPK [20] which could be utilized in the CC-WSN depending on the sensitivity of cargo. The assumption here is that the more sensitive a shipment (e.g. a vessel carrying military hardware), the stronger encryption will be needed and hence once can afford to place a larger power supplies into those containers. In CC-WSN, a pair of public/private keys is generated for each sensor. The private key is known to that specific sensor and the investigator, while the public key is public to all other nodes. All cargo containers need to be pre-screened at the port of origin, after which an encryption key for each container can be deployed.

A digital signature is computed using the same private key to ensure the integrity and authenticity of the data. For example, let us assume that node C_i is going to send data $m^{i \rightarrow j}$ to node $C_{j \neq i}$, and the private key of C_i is x_i, while the public key is g^{x_i}. Also, the signing function is denoted as $sign()$, the encryption function is $enc()$, and $h()$ is a cryptographic hash function. Upon communication, C_i uses $h()$ to compute the hash value of $m_{(i \rightarrow .)}$, and signs the message by encrypting the hash value with its private key x_i using the signing function $s()$:

$$S_{m_{i \rightarrow .}} = sign(h(m_{i \rightarrow .}))$$

then, C_i sends $C_{j \neq i}$ the whole packet $P_{(i \rightarrow j)}$ that contains the message $m_{(i \rightarrow j)}$ as well as its signature $S_{(i \rightarrow j)}$, as:

$$P_{(i \rightarrow j)} = [m_{(i \rightarrow j)}, S_{(i \rightarrow j)}]$$

When C_j receives $P_{(i \rightarrow j)}$, C_j can check the integrity and authenticity of the message by verifying the attached signature $S_{(i \rightarrow j)}$, since C_j knows C_i's public key g^{x_i}.

Therefore, an attacker will not be able to generate a valid response since the private key is unknown to him/her. Moreover, the signatures are much shorter and thus save time since hashing is generally much faster than signing in practice. By doing so, the integrity of the messages is guaranteed by the hash function while the authenticity is ensured by signing the hash value.

IV. Conclusion

In this paper, we proposed to deploy wireless sensors in cargo containers to ensure the integrity of the containers having pre-screening process taken place at the port of origin. When an intrusion is detected, an event is raised. The warning message, issued as a result of an intrusion is split into n pieces using an IDA and each message slice is delivered to a nearby node. Therefore, even if a node is compromised and the intruder removes the computing element from the container, the intrusion event can still be reported by its neighbors that are alive. After the ship arrives at its destination port, the log messages can be reconstructed using the IDA decoder and then analyzed. To prevent blocking of wireless signals by the attacker, the nodes actively query nearby neighbors for their status. An intrusion alert will be raised if the queried node is failed to respond or responded with an invalid message. The messages for the active status polling are encrypted using symmetric encryption with a pre-distributed key only known to that specific node and the investigator.

V. Future Work

Investigations into development of a secure communication protocol for CC-WSN, simulations regarding signal reception versus placement of containers, and potential use of JigDFS [21] in CC-WSN to store the table of containers with their status are the next phases of our investigation. JigDFS is a distributed file system which mainly aims to protect user privacy by providing plausible deniability. However, the security properties of JigDFS will not only protect the communication between nodes, but also secure the contents of the intrusion logs.

Acknowledgment

This work is based, in part, upon research supported by the National Science Foundation (under Grant Nos. CNS-0619069, EPS-0701890 and OISE 0729792). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

The authors would like to thank Dr. Mhamed Itmi, Dr. Sean Geoghegan, Dr. Edi Tudoreanu, Grady McCorkle, and Greg Fundyler for fruitful discussions.

References

[1] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[3] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, 2002.

[4] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 129–143.

[5] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228–258, 2005.

[6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2002, pp. 41–47.

[7] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "Spins: security protocols for sensor networks," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2001, pp. 189–199.

[8] R. D. Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2003, pp. 62–71.

[9] T. Park and K. G. Shin, "Lisp: A lightweight security protocol for wireless sensor networks," *Trans. on Embedded Computing Sys.*, vol. 3, no. 3, pp. 634–660, 2004.

[10] J. Deng, R. Han, and S. Mishra, "Insens: Intrusion-tolerant routing for wireless sensor networks," *Computer Communications*, vol. 29, no. 2, pp. 216–230, January 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2005.05.018

[11] ——, "A performance evaluation of intrusion-tolerant routing in wireless sensor networks," in *In the 23rd IEEE International Conference on Distributed Computing Systems (IPSN 2003)*, 2003, pp. 349–364.

[12] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *J. ACM*, vol. 36, no. 2, pp. 335–348, 1989.

[13] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[14] R. G. Gallager, "Low density parity-check codes," Ph.D. dissertation, M.I.T., Cambridge, MA, USA, 1963.

[15] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, pp. 1064–1070 vol.2, May 1993.

[16] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 569–584, Feb 2001.

[17] M. Luby, "Lt codes," in *FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2002, p. 271.

[18] A. Shokrollahi, "Raptor codes," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2551–2567, 2006.

[19] C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.

[20] R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus, "Tinypk: securing sensor networks with public key technology," in *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2004, pp. 59–64.

[21] J. Bian and R. Seker, "Jigdfs: A secure distributed file system," in *Proceedings of 2009 IEEE Symposium on Computational Intelligence in Cyber Security*. IEEE, 2009.