# Security in resource-constrained computer systems

Mahnaz Malekzadeh Amirhossein Fouladi
*Email: {mahma796,amifo436}@student.liu.se*
Supervisor: Christian Vestlund, {Christian.vestlund@liu.se}
Project Report for Information Security Course
*Linköpings universitetet, Sweden*

## Abstract

*The resource constraint issues in embedded system networks affects the security mechanisms which are proposed for such networks in comparison with traditional networks with no resource consideration.*

*In this paper we make a survey on different security concepts in resource-constrained networks. Furthermore we exemplify the survey by presenting several security mechanisms which are affected by the specific resource constraint in three different types of networks.*

*We conclude that the limitations in power, bandwidth, memory, time and computational units in such networks highly restrict the security mechanisms.*

## Introduction

Security mechanisms are needed in Embedded System networks like as in conventional networks. However a main difference is appeared in terms of resource constraint issues in embedded system networks. This noticeably limits the choice of security mechanism in such networks. For instance in a Wireless Sensor Network we have several distributed nodes within the network. Each node is supposed to do a specific task which is sensing the environment by its integrated resources such as processing unit, memory space and power supply. Obviously no other external supply is connected to the sensor node. So in such a resource-constrained network we have to be very careful about resource consumption by different strategic mechanisms that should be applied to that network. One of these mechanisms is security. Security mechanisms usually add overhead to the packets which are traversing between nodes in the network. This overhead undoubtedly needs more resources and the designer has to consider this issue and propose a suitable mechanism accordingly.

In this paper first we study the general security issues and challenges in embedded networks. Then we specifically focus on networks such as wireless sensor networks, mobile Ad Hoc networks and real time embedded networks. In each sample network we study the resource constraints and their effects. Furthermore we go through the proposed security solution by the authors to overcome these constraints.

At the end we compare the security mechanisms in both resource-constrained embedded network and a traditional wired network like LANs. We specifically base this comparison on the issue of resource availability and how this it affects the choice of security mechanism in resource-constrained embedded networks.

## Design challenges

There are various issues that arise when designing security mechanism for embedded systems and especially in a network environment. These issues include different resource constraints which exist in such environments such as limited amount of available energy to feed to the system, low processing capabilities, limited amount of storage space and low bandwidth for network communication. There are also a set of principles and practices that should be considered when designing security mechanisms for resource-constrained systems. For example one should always keep design simple and avoid features unless absolutely necessary. Interactions between different parts of a system usually causes many unknown security problems, so keeping explicit boundaries between the parts is a must. Since we are dealing with constraints, methods for optimizing the system seems very appealing but one should always bear in mind that not every optimization method is safe to apply. For example if the optimization involves removing or tweaking some parts of an algorithm or changing the core functional parts, the implementation should be thoroughly tested to ensure that security level provided by the algorithm is not degraded in anyway.

In [7] it has been suggested that security should be considered through out the whole process of software development from requirement analysis to coding and testing and not just as a set of additional features that should be attached to the system to make it secure. A set of best security practices has been proposed to be applied in each stage of software development. For example security requirements can be captured along side the functional requirements of the software during requirement analysis phase.

This holistic approach towards security has led designers to incorporate security mechanism in software and hardware architecture of embedded systems. For example in [9] J.Zambreno et al. propose a security architecture which involves having a FPGA-based security component for the purpose of lifting the security load from the main processor and isolating security operations from other functions of the system. The compiler for the system is capable of producing executable which are partly encrypted so the developer can fine-tune the performance and security of the software according to the needs of the system. The FPGA component is then responsible for fast decryption of encrypted parts and providing tamper-resistance for unencrypted parts of the program using a distance function.

Another security challenge for today's embedded systems is the issue of reconfigurablity. Manufacturers of embedded and mobile systems want to be able to reconfigure the devices in order to provide more versatile and improved functionality but that just poses more security risks to the system by increasing complexity of the software and hardware [8].

In [8] a security architecture for reconfigurable networked embedded systems is proposed. There are three security components in this architecture which provide required security services. A secure communication component which provides secure communication services for remote reconfiguration. This functionality is mostly provided through cryptography and secure communication protocols. The second component is a rekeying component which is responsible for key distribution and revocation. The third component is called ALoader and it is the main component that allows reconfigurability for the device through loading of new configuration from network. In order for the ALoader to do its job securely, it checks that the new configuration is coming from a trusted source and the integrity of the it is not compromised along the way.

As embedded systems are getting more pervasive in mission-critical environments such as nuclear power-plants and vehicles' braking systems where failure is no option, the issue of availability of these systems are getting more important. Failure of these devices even for a very short period of time can cause irreversible damages such as death [5], [11].

Unfortunately traditional cryptographic mechanisms which provide confidentiality, integrity and authentication services are not enough to deal with the issue of availability and further mechanisms should be employed to protect embedded systems from denial of service attacks. Real-time embedded systems are even more vulnerable to this kind of attacks since attacks that cause resource exhaustion can easily lead to missing of deadlines.

In [10] a method for detecting these kind of attacks on network embedded systems is proposed. Since embedded systems are less complicated that conventional enterprise network systems, it is possible to use behavioral analysis of the system to detect abnormal activities that might be a sign for potential attacks. In this way, one can avoid using packet probability and signature based method that might need additional resources which is not affordable for embedded systems with limited resources. The proposed method is called Multi-stage packet filtering and the aim of it is to do packet filtering as early as possible when a packet comes into the device. For example the first stage for incoming packet detection would be the I/O interrupt produced by the hardware. This approach can be followed up to the TCP/IP stack, dropping or accepting packets in each stage. In order for attack detection, state machines are constructed to identify the expected behavior of the system. Then frequent deviation from this behavior can show possible attacks on the system

In the following, cryptography as a major security mechanism to provide confidentiality and data integrity is discussed in the context of networked embedded systems. After that, three examples of resource-constrained embedded networks has been considered with respect to challenges in cryptography as well as other security issues.

### Cryptography

Since cryptography is one of the fundamentals of network security it also plays an important role in security of resource-constrained systems. The problem is that encryption/decryption is a resource intensive task that many embedded systems with small storage and limited processing capabilities can not cope with[1]. In [6] a comparison of various cryptographic algorithms has been made in terms of memory usage, speed and energy consumption. Algorithms that has been compared are DES, 3DES, AES and RC4. Among them RC4 tend to be faster, more energy efficient and use less storage space for encryption/decryption operations and therefore more suitable in resource-constrained environments.

## Wireless Sensor Networks

Wireless sensor networks are a type of ad hoc networks with severe constraints in resources such as power, computational unit, memory and bandwidth. In some applications the transmitted data between nodes are highly sensitive and need to be protected against security threats. On the other hand applying security is challenging in wireless sensor networks due to the resource constraints rather than a conventional network. For instance in a traditional network, security mechanisms normally adds 16-32 bytes overhead to the packet being transmitted over the network while this could not be handled in a wireless sensor network due to the resource constraints. So a data link-layer security mechanism which is called cipher stealing method is proposed in [1] to address the message expansion problem in wireless sensor networks.

The main reason to choose a data link-layer mechanism is that in wireless sensor networks communication is one to many. It means that all nodes send the sensing data to a particular central base station while in a traditional network an end-to-end communication is available between two nodes. So in order to avoid data duplication and aggregation in a wireless sensor network each packet's content has to be accessed by nodes. This leads to the selection of data link-layer security architecture.

A wireless sensor network would encounter problem with message expansion within the applied security mechanism since it is limited by bandwidth and computational resource.

Cipher text stealing method solves this problem with changing the last two blocks of the plain text, reordering the transmission of the last two blocks of cipher text while avoiding message expansion.

Cipher text stealing encryption uses a standard cipher block chaining. It pads the last plain text block with 0, then the padded plain text is encrypted by the standard CBC mode. It swaps the last two cipher text block and truncates the cipher text to the size of the initial plain text. For decryption, the second to the last ciphered blocks is decrypted. The cipher text is padded to the nearest multiple of the block size. Then the last two ciphered blocks are swapped. The cipher text is decrypted using CBC mode again except the current last block. The generated plain text is truncated to the length of the original cipher text.

## Mobile ad hoc Networks (MANET)

Mobile ad hoc networks are consisted of mobile nodes communicating by a wireless channel. Security in mobile ad hoc networks is analyzed based on bandwidth and energy constraints. Since there is no central authority in such a wireless channel every node could act as a router no matter if it is a trusted user or a malicious one. So energy saving may become difficult if a node is not able to forward the packets correctly. Nodes within mobile ad hoc networks have limitation in terms of power supply and physical protection. So security mechanisms which are defined for traditional network would not be effective in such networks. For instance if each node wants to encrypt packets before transmission, this would drain unnecessarily its energy supply. Broadcasting in such networks for packet transmission causes many nodes which are not the target nodes receive the packet and waste their energy in this way.

To address security concern is MANETs, distributed trust based security architecture is proposed in [2]. First a clustering mechanism is suggested and a node which is called guard node would be responsible for local monitoring to force security.

The trust model is based on an algorithm that has several trust parameters such as number of forwarded, misrouted, dropped, falsely injected packets, etc. If a node A wants to calculate the direct trust value of node B, it monitors B's behavior and collects local data of B. Node A then asks the common neighbors for their recommendations about node B. After getting all trust information A stores all data in the direct trust table.

In the proposed approach Clusterheads (CH) have to monitor the incoming and outgoing traffic of neighboring nodes to identify and block the malicious node. In this proposal a node called guard node overhear the traffic of one-hop neighbors in specific intervals. While the Clustehrheads and nodes are deployed, every node wants to join the network by sending the public key. When this message reaches to the Clusterhead, the Clusterhead generate a unique IP address and hash it with a random number to node's id. The node's public key is used to encrypt node id. IP address and random number are encrypted with node id. This is broadcast to the network and the intended node would receive it. The message is first decrypted by public key to achieve the random number. Then it decrypts node id and IP address by symmetric key cryptography.

All nodes in the network get an initial trust value that allows them to communicate within each cluster. They could not exchange critical data until they get a Trust Certificate. Guard nodes calculate the direct trust and send it to the Clusterhead based on trust model described above. Clusterhead generate the global trust and calculates trust certificate based on received information.

By this method routing table overflow and resource consumption attacks are prevented since one of the parameters in local monitoring is the number of packets which are falsely injected.

## Network Real Time Applications

One of the key constraint on real-time embedded systems are timing constrains. Consider the case of a car braking system the uses embedded system. If the deadline between pushing the brake pedal to actually enforcing the brake is not met, it's not hard to image how catastrophic the result would be. In some real time applications, security concerns are as important as timing constraints. In [3] a security overhead model is proposed to measure security overheads in security-critical applications. Furthermore a security-aware scheduling strategy or SAREC is suggested which is based on security overhead model. Then a security-aware real time scheduling algorithm (SAREC EDF) is utilized to evaluate SAREC.

Clusters are recently used in real time applications where meeting deadlines are as necessary as computation results. Due to the sensitive data and information in some of these applications, security becomes a main concern besides timing issues. SAREC is proposed to combine security requirements with real time applications running on clusters.

In this article clusters are considered as nodes that are connected through a high speed interconnection network. Tasks are queued to be scheduled based on earliest deadline first (EFD) schedule policy. If they are accepted to be schedule, they would be treated while satisfying two conditions: 1) security level requirements will not cause a task misses its deadline 2) security level will not lead any subsequent accepted task misses its deadline as well. Obviously in such a network timing is considered as the main resource constraint that affect the level of security could be applied.

A security-aware scheduling tries to maximize the security level of accepted tasks. In a resource-constrained embedded network security overhead causes performance degradation. The proposed security model is capable of measuring the security overhead experienced by schedulable tasks. This leads to construct a working security-aware scheduling while enforcing the required security level. In this way security-aware scheduler knows the security overhead and consider it while scheduling the tasks.

The security overhead model consists of three security overhead as follows: 1) encryption overhead 2) authentication overhead 3) integrity overhead. The sum of these overheads are calculated as the total overhead which is used by SAREC strategy.

The SAREC-EDF algorithm incorporates EDF scheduling in to the SAREC strategy. The task could be scheduled if it could be completed by its deadline. For a cluster a task is schedulable if a schedule exists for it on at least one node. SAREC-EDF tries to meet the real-time requirements of each task. This can be achieved by calculating the earliest start time and minimum security overhead. Beside that it is check if the task can be completed before its deadline. For every node if the deadline could not be met the security level is set to zero indicating that the task could not be scheduled on that node. Consequently if there exists no node for scheduling task, it would be rejected which means it is not schedulable with particular timing constraint as well as security level requirement.

Simulations are performed to compare the proposed SAREC-EDF with three basic scheduling algorithms: 1) SHMIN that chooses the lowest security level for arriving task 2) SHMAX that picks the highest security level for each task 3) SHRND that selects a random security level. SAREC-EDF which is capable of calculating the security level required by a task results in an improved performance with more qualified security compared to the three mentioned above algorithm.

## Conclusions

In this survey, first we try to give a clear and summarized picture of different security mechanisms that could be applied in a network. The ability and level of security mechanism in each network varies based on the security which is required for protecting the information being transmitted over the network.

As the main concern in this survey is security in embedded systems with different resource constraints we continue with exemplifying some popular networks. Obviously there is a big difference in such networks compared to a conventional one. In an ordinary network such a LAN there is no need to be concerned about resources while deciding over security mechanisms. In other words these sort of networks are not highly affected by limitations in resources like power, computational unit, bandwidth and time. Unlike that, in a resource-constrained embedded network any desired level of security could not be applied due to limitations in resources.

In our sample networks we try to show that the proposed security level in each security-critical application is significantly affected by resource constraints.

For instance in the selected security in wireless sensor network bandwidth and computational units are the main constraints that highly affected the design of security mechanism.

In the second scenario for mobile ad hoc networks bandwidth and energy resource are the main concerns while choosing a suitable security mechanism.

Finally in the last real time application it is critical that tasks meet their deadline while they are provided with adequate level of security. So the proposed scheduling policy is designed in a way that both above are satisfied in the end.

So the proposed security mechanism in such networks may have some weaknesses compared to a conventional network. But efforts are made to construct mechanisms which fulfill the desired security levels.

## References

1. Ramnath Venugopalan, Prasanth Ganesan, Pushkin Peddabachagari, Alexander Dean, Frank Mueller, Mihail Sichitiu “Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis", CASES '03 Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems. 2003.
2. Md. A. Rahman, M. K. Debnath, “An Energy-Efficient Data Security System for Wireless Sensor Network”, 11th International Conference on Computer and Information Technology, Bangladesh., 2008.
3. P. Chatterjee, I. Sengupta, S.K.Ghosh, “A Distributed Trust Model for Securing Mobile Ad Hoc Networks”, IEEE/IFIP International Conference on Embedded and

Ubiquitous Computing, 2010.

4. T. XIE, X. Qin, A. Sung, “SAREC: A Security-aware Scheduling Strategy for Real-Time Applications on Clusters”, 34th International Conference on Parallel Processing , Norway, 2005
5. Timothy Stapko, “Practical Embedded Security: Building Secure Resource-Constrained Systems”, Newnes, 2007
6. Prof. Dr.-Ing. Gunar Schorcht, Miller Alexander: “Embedded Systems Security: Performance Investigation of Various Cryptographic Techniques in Embedded Systems”, IT Security for the Next Generation - European Cup, 2011.
7. Paul Kocher, Ruby Lee, Gary McGraw, Anand Raghunathan and Srivaths Ravi “Security as a New Dimension in Embedded System Design”, Design Automation Conference, 2004. Proceedings. 41st
8. Gianluca Dini, Ida Maria Savino, “A Security Architecture for Reconfigurable Networked Embedded Systems”, International Journal of Wireless Information Networks, 2010
9. J. Zambreno,A. Choudhary, R. Simha , B. Narahari, N. Memon, “SAFE-OPS: An Approach to Embedded Software Security”, ACM Transactions on Embedded Computing Systems, Volume 4 Issue 1, February 2005.
10. H. Karen Lu, “Attack Detection for Resource-Constrained Network Devices”, ICONS '08 Proceedings of the Third International Conference on Systems, 2008
11. Philip Koopman, Carnegie Mellon University, “Embedded System Security”, *IEEE Computer*, July 2004.