Towards Secure Time-Triggered Systems

Florian Skopik[1], Albert Treytl[2], Arjan Geven[3], Bernd Hirschler[2], Thomas Bleier[1], Andreas Eckel[3], Christian El-Salloum[4], and Armin Wasicek[4]

[1] AIT Austrian Institute of Technology, Safety and Security Department
firstname.lastname@ait.ac.at
[2] Austrian Academy of Sciences, Institute for Integrated Sensor Systems
firstname.lastname@oeaw.ac.at
[3] TTTech Computertechnik AG
firstname.lastname@tttech.com
[4] Vienna University of Technology, Institute of Computer Engineering
firstname.lastname@tuwien.ac.at

Abstract. This paper presents the development of a novel joint safety and security architecture for dependable embedded time-triggered systems. While fault-tolerance properties of time-triggered protocols have been very well studied, research on security aspects for time-triggered systems have hardly been covered. Therefore, we explore system design principles which efficiently realize security mechanisms for time-triggered architectures. A particular focus is on synergistic effects of security and safety-related functions, thereby supporting the roll-out of safety-critical embedded systems even in 'untrusted' environments. As a main contribution, we present the Secure COmmunication in Time-Triggered sYstems (SCOTTY) approach to build secure time-triggered systems.

Keywords: time-triggered systems, security challenges, safety-criticality

1 Introduction

It is widely acknowledged that security is gaining significant importance in the area of embedded systems and in particular in safety-critical systems. An important aspect of these emerging security requirements is that traditional embedded systems were operated in physically secured environments like within a nuclear power plant. The trend towards ubiquitous and pervasive computing creates open environments that do not offer this physical security anymore. In some environments even the owners of a system can be potential attackers. Successful attacks could lead to catastrophic events like mechanical damage on the equipment, financial loss, or - in the worst case - the loss of human lives. It is of utmost importance that **safety** *and* **security** is seen in an integral way [10], because an attacker could target the whole sphere of control of the embedded system that also encompasses its physical environment.

The predefined time-triggered schedule can be used as a basis to introduce a synergetic security concept for time-triggered communication protocols. The bus guardian supervising the correct execution of the schedule can serve as the

core for a firewall component protecting the network infrastructure and the application traffic using the network. However, this direct use of the bus guardian requires assumptions hardly to be met in practical implementations, especially, if distributed systems are considered: (i) First core assumption is that an attacker has no physical access to the switches and the connected devices (e.g., the system has to be located in a locked room). An attacker having physical access to the system could simply change the wiring on the switch, or physically bypass the bus guardian. (ii) Secure (initial) authentication of devices during startup is required to prevent the insertion of malicious devices. An attacker can - intentionally or unintentionally - insert malicious devices, e.g., as replacement of defect devices. (iii) Third core assumption is that no man-in-the-middle attack [1] is possible. If an attacker can modify the content of messages during transmission, confidentiality and integrity of the message content is endangered.

Thus, the natively provided properties, implemented in a time-triggered protocol are not sufficient. The major objective of our approach is to extend the features of time-triggered protocols to a full security architecture that can deal with threat models, including 'physical access' of attackers. The isolation in the time domain is the basis for a novel security architecture that in a synergetic way extends safety functionality by security functionality to counteract malicious attacks. Additionally, the following challenges need to be addressed: (i) *Scalability from small-scale closed networks to large-scale integrated networks.* At the point, when a system becomes larger (i.e. system of systems, large-scale time-triggered networks), the possibility and the interest in unauthorized access to such systems also increases. (ii) *Protection from intended and unintended modifications.* In addition to increased scale there are also possibilities that are brought up by the mere availability of a technology, where the owner of a system can achieve access and accidentally or intentionally modify parts of the system behavior, thereby altering important dependability properties.

The contributions in this paper are: (i) *Security Challenges and Design Aspects.* We discuss the fundamental requirements for hardening time-triggered systems in order to reach an appropriate level of security. (ii) *Security Architecture for Time-triggered Communication.* We highlight promising approaches and propose basic architectural models to meet the aforementioned requirements.

The remainder of the paper is organized as follows. Section 2 discusses background and related work, Sect. 3 introduces the SCOTTY design principles and basic models, and Sect. 4 sums up our work and concludes the paper.

2 Basic Concepts and Related Work

Dependable embedded computer systems [2] are a well-accepted solution for many applications in the fields of transportation, automation, and medicine [6]. The dependable-computing community has made a tremendous progress in the past decades building ultra-dependable systems out of less reliable components. Further, a multitude of practical techniques with respect to fault masking, error detection, fault diagnosis, and recovery has evolved to improve the reliability of

a safety-critical system. Yet, almost all of these approaches put a focus on the safety aspect and assumes that the system is not under active malicious attack. The SCOTTY approach has the aim to introduce a security architecture allowing operation of these systems in non-isolated environments and withstanding malicious attacks [1] - a trend that needs to be addressed [15].

The communication system which interconnects single components is crucial for the safety *and* security properties of the final system [16]. When designing real-time systems, the time-triggered communication paradigm [7] proves to be particularly promising, because of its determinism and predictability facilitating validation and verification efforts for accordingly built systems [9]. A multitude of time-triggered protocols (e.g., FlexRay, TTP, TTEthernet) [4] has already been successfully deployed in many application domains. However, none is known dealing with security services. Within SCOTTY, the TTEthernet system is used as a representative of time-triggered protocols, mainly because of its broad industry acceptance, e.g., by the NASA [5]. Highly dependable time-triggered communication protocols like TTEthernet, FlexRay or TTP/C provide timeslots statically assigned to unique nodes according to a pre-defined time-triggered schedule. Each node is allowed to transmit data on the physical communication medium exclusively during its assigned time slots. This policy is usually enforced by special encapsulation mechanisms, called bus guardians, which prevent any misbehaving node from disrupting the communication among other nodes by transmitting outside of its allocated time slots thereby preventing message collisions. Whereas there exist generic solutions for dependable embedded computing [3] and security for event-triggered protocols has been very well researched and applied (e.g., in wireless sensor networks [8]), research on security in the time-triggered domain is still in its infancy [13]. The core of time-triggered protocols is a common knowledge and usage of time itself. Security research needs to focus on providing a common knowledge of time to implement security features.

Security for clock synchronization is essential, since the domain of clock synchronization for industrial communication systems and sensor networks is becoming a vital aspect for system operation. The synchronized clocks are building the foundation for many critical application domains. A variety of services is enabled by synchronized, distributed clocks, ranging from the application layer (timestamping of measurement data) down to the network layers where clock synchronization is used to schedule media arbitration (e.g., TTEthernet, Flexray) or location determination of sensor nodes. Due to the increasing interconnection of networks, security is of growing interest for industrial networks. Field level devices are connected to external networks and assumptions based on restricted physical access do not apply any more. Clock synchronization and security have to be carefully thought about when they are used together [11, 12]. Two considerations have to be taken into account. Firstly, security mechanisms often make use of distributed synchronized time bases. Secondly, clock synchronization in general and timestamp information in particular, which is exchanged over the network to achieve synchronization, are assets that need to be protected by appropriate security mechanisms.

3 The SCOTTY Approach

The SCOTTY security architecture is realized on top of an existing time-triggered real-time communication system. It provides a highly flexible, adaptable, and applicable security layer, which closely integrates with the safety functionality of the system and facilitates existing fault tolerance mechanisms.

3.1 A Security Architecture for Time-triggered Communication

The fault-tolerance mechanisms and dependability properties of time-triggered communication protocols already provide mechanisms to separate traffic and manage access control. The security part of a time-triggered system can benefit from its fault tolerance mechanism, because both require similar properties [6, 13]. Yet, whenever an attacker is able to physically modify parts of the system, like replacing components with malicious components or changing the wiring, the safety properties of today's system cannot be guaranteed anymore.

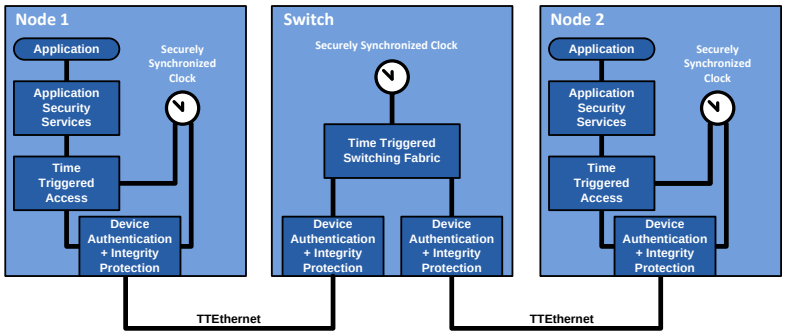

Fig. 1. Functional overview of a secured SCOTTY time-triggered system.

Figure 1 shows a schematic overview of the system. Core element is the time-triggered switch and its predefined timeslot scheduling module (Time Triggered Access), which facilitates the access control and traffic separation in the safety domain of the time triggered protocol. To be able to gain synergies for the security system subsystems for (i) device authentication, (ii) secure clock synchronization and (iii) application level security have to be added.

Device authentication subsystem. One of the main prerequisites for security is that devices properly mutually authenticate themselves to prevent insertion of malicious devices or messages in case of a man-in-the middle attack. Physical device security is not in the primary scope of our work. Nevertheless, the design of the security protocol foresees these important requirements by allowing the integration in security modules such as smart cards or trusted platform modules. The main challenge in the design and implementation of device authentication mechanisms is to retain the temporal properties of a real-time system, i.e., the designer has to take care that introducing an authentication scheme in

the real-time communication does not spoil the original real-time properties of the time-triggered system [14]. Any additional and unpredictable delay in the communication path is critical for the communication and consequently for the access control and traffic separation based on the time-triggered protocol.

Secure clock synchronization. A major objective of the security architecture is to protect the temporal properties of the system in the presence of malicious attacks. In a time-triggered system, the accurate temporal coordination of distributed activities is controlled by a consistent global time base. TTEthernet already provides fault-tolerant clock synchronization mechanisms that harden the global time against accidental faults like single-event upsets (SEUs). In order to harden the global time against malicious attacks within SCOTTY a secure clock synchronization protocol is developed on top of the existing clock synchronization protocol. The targeted security goal is to maintain the integrity of the global time base, even under attack, which is a service for communication and application layer security. The main goal is that any node should either be in trusted synchrony with the global time base or reliably detect that it has lost synchronization. Malicious message delay needs to be detected and compensated even considering an attacker model where every message can be potentially created, forged, replicated, deleted, delayed or accelerated by the attacker. This scenario is totally different from the fault hypotheses of classical clock synchronization algorithms targeted on accidental faults (e.g., as used in FlexRay or TTP), which are always based on an upper bound of faulty nodes or messages. The trusted global time base is realized based on mutual checking of signed clock synchronization messages. Additionally, measures need to counteract the modification of messages. These measures are in particular important during the start-up phase. In this respect, authentication and message integrity are the two major issues. Although security systems exists to protect these properties the challenges in SCOTTY lie in the fact that on the one hand high real time requirements need to be met and that on the other hand in opposite to many other real-time security protocols time cannot be used as a base. In particular execution time and jitter need to be considered. Possible solutions are based on a very close integration of security in the time slot scheme to avoid jitter and dedicated algorithms to detect delay.

Application level security. Applications on the one hand demand that the communication infrastructure is secure, yet on the other hand they also need additional application specific security service to protect data on an end-to-end basis. The envisioned application level security provides the following secure communication services on top of TTEthernet in order to guarantee real end-to-end security: (i) Authenticated unicast and broadcast; (ii) Application authentication; (iii) Confidential message delivery and integrity protection. The security layer establishes authenticity, integrity and confidentiality even if the attacker has physical access to the system. Since the secure clock-synchronization developed in SCOTTY already protects the global time, time itself can be used to efficiently establish other security properties with low computational and communication overhead. Examples for security-related benefits of a trusted global

time are replay prevention of messages containing application data, or broadcast authentication via efficient symmetric protocols like TESLA [5, 14].

3.2 Scalability and Legacy Support of the Security Architecture

The security components described before protect the communication infrastructure of and the application data exchanged via a single time-triggered system. Coverage of applications in a cross-domain approach demands the scalability and the support to integrate security-unaware devices. The SCOTTY concept foresees native support for these features.

Scalable extension of secure time-triggered networks. Figure 2, shows a simple scenario, where all physical elements that are associated with a link (i.e., the devices, switches and the cabling) are attached to a single switch. This example corresponds to the scenario with a physically secured wiring within a single cabinet. In TTEthernet multiple security-unaware virtual communication channels can coexist on a single shared physical communication infrastructure. The encapsulation mechanisms of TTEthernet ensure that faults cannot cause any influence between virtual communication channels. With the SCOTTY security architecture these channels are also protected against malicious attacks. Hence, the different virtual communication channels (orange and green channels in Figure 2) are also security domains protected against each other. The security services add the following additional properties to normal TTEthernet channels (i) Malicious or bogus devices cannot connect or send messages; (ii) Man in the middle modification of messages is not possible. If application layer confidentiality is used also eavesdropping can be prevented; (iii) Clock synchronization is protected.

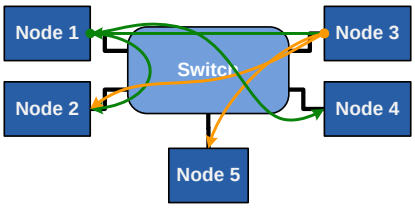

Fig. 2. Single communication cell with secured virtual communication channels.

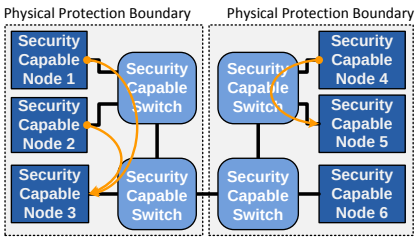

Fig. 3. Cascaded communication cells with secured virtual communication channels.

The security domains can be extended by cascaded switches where the interconnection between the cascaded switches occurs transparently (Figure 3). Each link is separately authenticated, i.e. to connect additional switches only the link between the switches has to be added to the security system. The concept of time-triggered scheduling already supports such cascaded switches and therefore can be used as it is. For the SCOTTY security architecture it is irrelevant whether these connections are within a single switch (e.g. a cabinet) between

switches within a single physical protection boundary, or even between different physical protection boundaries (e.g. multiple buildings).

Inclusion of security-unaware devices. To support existing security-unaware devices the SCOTTY security architecture also foresees the bridging of two network segments via a secure tunnel established by security gateways. In this scenario, the security layer only has to be installed in both security gateways. The other devices do not have to be modified and are relieved from computational intensive calculation of cryptographic operations. As shown in Figure 4, such a tunnel allows to interconnect two existing networks that do not support security. On the left side (physical protection boundary A) an implementation with a firewall-like security gateway is depicted. This security gateway has two ports and includes all security functionality. This concept is preferred from a security and validation point of view, since there is a single component maintaining the security. On the right side (physical protection boundary B) a security-capable switch and a detached security gateway managing the tunnel(s) is used. Given the traffic separation properties of the security-capable switch no security breach occurs. This solution has advantages in the safety concept since multiple parallel tunnels can be used to connect two boundaries.

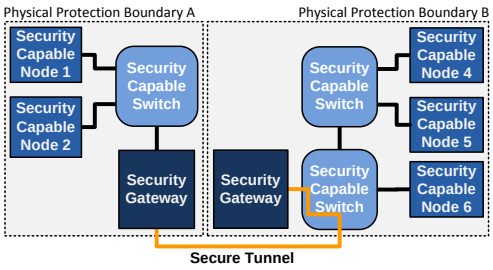

Fig. 4. Secure tunnels between security unaware networks.

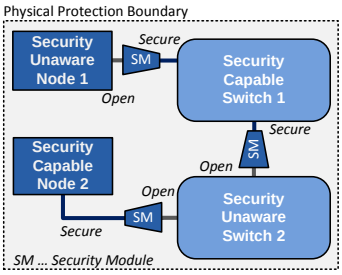

Fig. 5. Interconnection of security-unaware devices..

The concept of the security gateway is also scalable in a way that it can be used to interconnect single security-unaware devices. Figure 5 shows different examples how end-devices and switches can be connected. In this case not the complete aspect of the tunnel functionality is used but the gateway rather serves as a translator. The SCOTTY approach pays special attention to this aspect to develop versatile and re-usable security components that are in particular needed to offer a migration path for existing installations.

4 Discussion and Conclusion

In this paper, we introduced the SCOTTY approach, enabling secure communication in time-triggered systems. The key assumption is to relax the requirement of 'no physical access' of former safety-critical systems. To sum up, the advantages compared to existing solutions are:

- Full protection inside the time-triggered system that allows secure communications beyond physical protection boundaries.
- Synergetic use of safety and security components by reuse of functional modules and integration in existing safety-related design concepts and tool approaches.
- Protection of temporal properties of a real-time system in addition to the standard security attributes, authenticity, integrity and confidentiality.

Future work includes the implementation and evaluation of the introduced concepts under realistic conditions. Special focus will be set on the real-time behavior, attack analysis and versatility of the developed concepts and components to cope with existing and emerging threats and to allow use in numerous application areas.

References

1. Anderson, R.J.: Security engineering - a guide to building dependable distributed systems (2. ed.). Wiley (2008)
2. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.E.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Sec. Comput. 1(1), 11–33 (2004)
3. Bar-El, H.: Intra-vehicle information security framework. Tech. rep., Discretix Technologies Ltd. (September 2009)
4. Berwanger, J., Ebner, C., Schedl, A., Belschner, R., Fluhrer, S., et al.: FlexRay The Communication System for Advanced Automotive Control Systems (2001)
5. Cooney, M.: Nasa takes ethernet deeper into space (2009), http://www.networkworld.com/community/node/40899
6. Kopetz, H.: Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, Norwell, MA, USA, 1st edn. (1997)
7. Kopetz, H., Bauer, G.: The time-triggered architecture. Proceedings of the IEEE 91(1), 112–126 (2003)
8. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: Spins: Security protocols for sensor networks. Wireless Networks 8(5), 521–534 (2002)
9. Rushby, J.: A comparison of bus architectures for safety-critical embedded systems. Research Report NASA/CR-2003-212161 pp. 112–126 (2003)
10. Schoitsch, E.: Design for safety and security of complex embedded systems: a unified approach. In: Proceedings of the NATO Advanced Research Workshop on Cyberspace Security and Defense: Research Issues. pp. 161–174. Springer (2004)
11. Treytl, A., Gaderer, G., Hirschler, B., Cohen, R.: Traps and pitfalls in secure clock synchronization. In: ISPCS. pp. 18–24 (2007)
12. Treytl, A., Hirschler, B.: Securing IEEE 1588 by ipsec tunnels - an analysis. In: ISPCS. pp. 83–90 (2010)
13. Wasicek, A.: Security in Time-Triggered Systems. Ph.D. thesis, Vienna University of Technology (2012)
14. Wasicek, A., Salloum, C.E., Kopetz, H.: Authentication in time-triggered systems using time-delayed release of keys. In: ISORC. pp. 31–39 (2011)
15. Wolf, M., Weimerskirch, A., Paar, C.: Embedded security in cars: Securing current and future automotive it applications (2006)
16. Wood, A.D., Stankovic, J.A.: Denial of service in sensor networks. IEEE Computer 35(10), 54–62 (Oct 2002)