

Cyber security for embedded communications

Author
Lars Wolleschensky

Title
CTO

Organization
ESCRYPT

**2017 NDIA Ground Vehicle Systems Engineering &
Technology Symposium (GVSETS)
AUGUST 8-10, 2017 - NOVI, MICHIGAN
AUTONOMOUS GROUND SYSTEMS (AGS) / Cyber security**

Introduction

The purpose of this paper is to discuss embedded security, specifically for automotive sector. Most modern cars will have indirect Internet connections to safety-critical control systems. The proposals for using wireless roadside transmitters to send real-time speed limit changes to engine control computers, the use of Vehicle-to-Vehicle communications, are some obvious examples. Internet connections expose applications to intrusions and malicious attacks; security techniques developed for enterprise and desktop computing might not work for embedded application requirements. In this paper we discuss about existing security protocols used in modern vehicles. The detail discussion section of this paper provides information about modern automotive security protocols. The background section of this paper discusses about existing research work in embedded security area. The paper concludes with future research directions.

Detail Discussion

As the U.S. Army adopts intelligent vehicle mobility solutions, for instance in the Ground Vehicle Robotics' Autonomous Ground Resupply (GVR-AGS) project, the need to secure embedded communications channels in a resilient manner is paramount. Understanding the attack modes of potential cyber attackers against and the basic the prevalent automotive communications technologies will enable the U.S. Army to better refine suitable approaches.

Nearly every electronically run device, from cars to cell phones, video equipment to MP3 players, and dishwashers to home thermostats—embedded computers increasingly basically run our lives. Unfortunately, security for these systems is a serious problem and may be a bigger issue than security for desktop and enterprise computing.

Embedded systems are often highly cost sensitive, so most CPUs manufactured worldwide use 4-bit and 8-bit processors, which have limited room for security overhead. Many 8-bit microcontrollers cannot store a large cryptographic key, so uses in the enterprise world could be too expensive to be practical in embedded applications.

Another key issue: many embedded systems are interactive. A security breach can result in physical side effects, including property damage, personal injury, and even death. Software damages are harsh but can be fixed; many types of physical damage are irreversible.

Another issue is that embedded systems often have energy constraints, and most are battery powered. Some embedded systems can get a fresh battery charge daily, but others must last months or years on a single battery. By seeking to drain the battery, an attacker can cause system failure even when breaking into the system is too difficult. This vulnerability is critical in battery-powered devices that are power-hungry when it comes to wireless communication.

One more constraint is that embedded systems are created by small development teams and sometimes lone engineers. Organizations that write only a few kilobytes of code per year usually cannot afford a security specialist and often do not realize they need one. There is no standard development practice that includes rigorous security analysis, and until there is, developers may overlook even the solutions already available.

Centralized Control can be dangerous if the system permits transition only between a pair of “comfort” and “saver” set points in a home thermostat system, where an attacker could send false “I am coming home” messages to change set points and waste energy. If arbitrarily changing set point are allowed, the attacker could subject the house to extremes of heat and cold or even turn off the system, causing physical damage like bursting pipes in the winter. A properly designed system with safety interlocks and a well-administered password policy could prevent this from happening, but natural the potential for it to occur makes it a threat that must be countered before these matters are resolved.

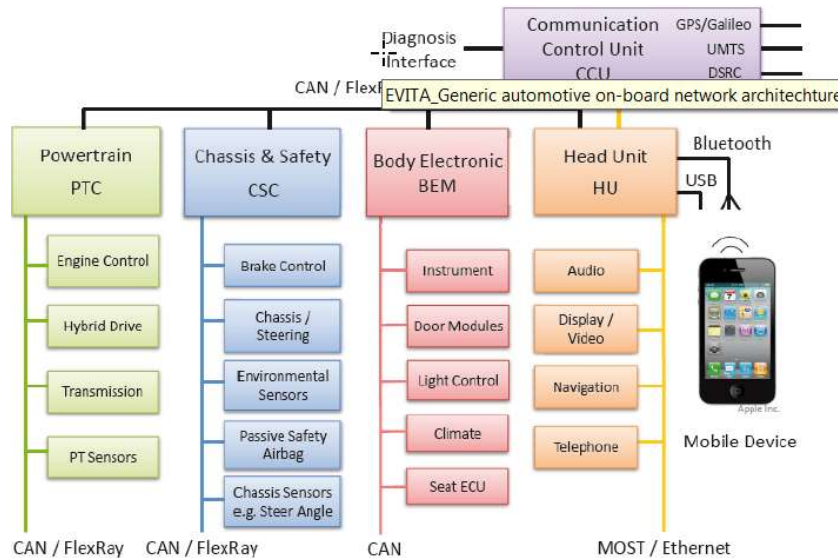
This same example can be used for another type of attack. Many thermostats, including at least one Internet brand, are battery powered. This is partly because line voltage is not available and partly because safely converting line voltage to a thermostat’s needs takes a large converter that costs money, requires extra wiring, and complicates electrical safety. Some thermostats use wireless networking to avoid wiring costs, but too many networking conversations can run the battery down quickly. If the thermostat is connected to the Internet, an attacker could run the battery down simply by repeatedly querying the thermostat’s status. A low-voltage detection circuit could disable the wireless connection before the battery died, but again, the author suggests that the developer needs to design this capability into the system.

The last issue is related to privacy. A person who can monitor your thermostat setting could also determine whether a person is likely to be asleep at home or out of the house. Even if an attacker cannot query the thermostat directly, simply monitoring traffic for inbound packets talking to the thermostat can indicate whether the house is vacant—and a potential burglary target.

It is concluded that, in many ways, we are not ready to deal with the security challenges we are sure to face at this point in time. Some involve simply ensuring that design teams acquire the right skills as they start making products that are exposed to security risks, but others involve significant research before we can hope to address them. Only when all possibilities and bases are covered can we rest a little easier in regards to embedded security.

This paper and presentation is based on secure onboard communication that covers the concepts of automotive bus system, threats on automobiles, examples of attacks on automotive bus system, secure onboard communication, AUTOSAR, and V2X. The paper starts with basic security goals and how they are achieved. A table is presented with different automotive bus systems (LIN, CAN, FlexRay, MOST, Bluetooth) and their difference with respect to their

architecture, access control, transfer mode, data rate, and few others. Automotive attackers have 3 categories: 1. Owner 2. Mechanic 3. Third party. The automotive bus system can have attacks related to injecting unauthorized frames, manipulating frames, and misuse of diagnostics interface. Automotive bus attacks are easy because of bus characteristics, and gateway characteristics.



The next part of the presentation provides detail about secure onboard communication. It starts with an architecture of unsecured CAN network, then partially protected and fully protected CAN network. Secure onboard communication is meant for securing sensor and inter-ECU communication. To protect symmetric keys HSM (Hardware security module) is used. Authentication of message is achieved through anti-replay counter and MAC. Message authentication and freshness are verified using AUTOSAR 4.2.1. Network messages can be protected with the HSM and SHE+ interface. Well tested algorithm like AES is used for encryption; some of the modes used for AES are AES-CMAC and AES-CBC. Some of the advantages of using SHE+ include 10 extra keys and verification flag. The presentation provides detail architecture and stack of AUTOSAR 4.2.1.

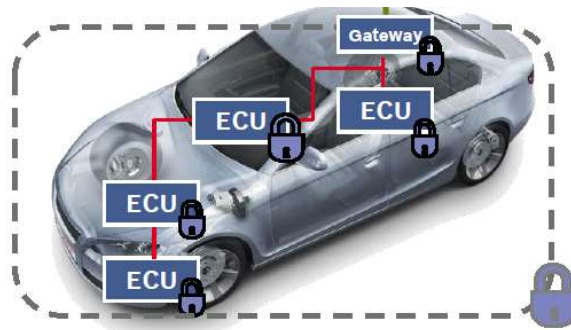
External communication via CCUs

→ Exchange of service data

- Vehicle service data
- Increasing driving efficiency

→ Flashing of Software

→ Internet Connectivity



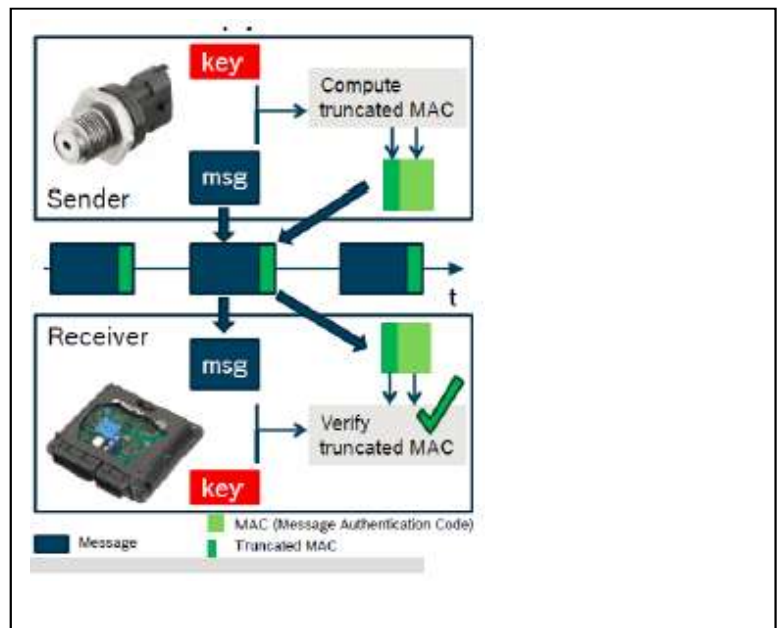
Requirements & Security Goals

- **Protect** critical functionality, vehicle safety and stakeholder assets
- **Authenticity & Integrity** protection of incoming messages
- **Confidentiality** to protect data
- **Vehicle Gateway:** to ensure E2E security

Solution

- **Vehicle Firewall** to inspect communication
- **Secure communication channels**
- **Security Hardware** to achieve secure vehicle identities
- **Mechanisms to isolate vehicles**

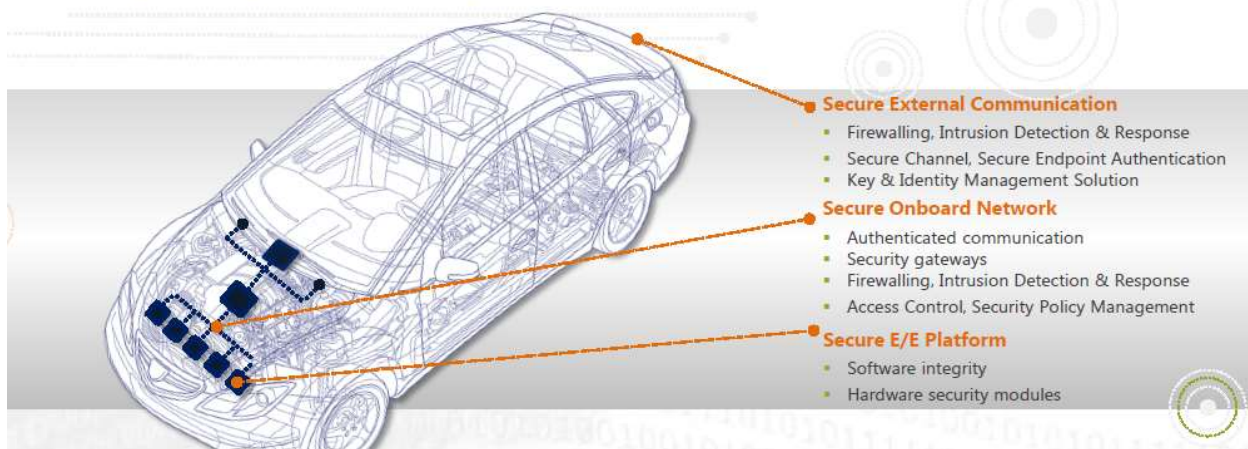
This paper and presentation also covers future automotive security challenges which are based on Vehicle-to-Vehicle(V2V) and Vehicle-to-Infrastructure(V2I) network. It discusses the underlying idea of V2V and need for security. Some of the recent development of V2V is discussed along with the DOT model deployment strategy.



While the underlying need to protect and secure the vehicle and its related infrastructure, is common between the general automotive industry and the Army's fleet of vehicle platforms, the repercussions of an intrusion or hack on an U.S. Army platform is significantly, even exponentially, greater in a number of areas. Understanding the current areas typically attacked in an automotive application, as well as possible avenues that could be exploited in the future, will assist the U.S. Army in applying the very best methods, tools, assets and processes to ensure the required level of operational security is retained as the move from analog/mechanical based systems to embedded mechatronic systems is realized during the modernization efforts and new developments of the mobile assets of the U.S. Army.

Holistic Security Solution

- Defense-in-depth approach
- Security building blocks on each layer



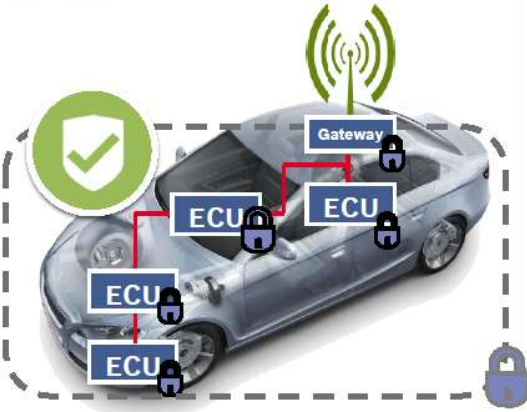
The digital revolution in vehicles has come a long way from using only a few microcontrollers to a smart phone on the wheel. There are many known embedded security problems in vehicles and software. For example, security issues with Sony's software, remote unlocking of BMW vehicles. Security experts like Miller and Valasek demonstrated how easy it is to control infotainment system and engine of vehicles remotely. There are other IT threats to vehicles; which include threats related to wireless communication, central gateway, engine, odometer, ECU, diagnostics interfaces and car2X.

Security in vehicles should be designed and implemented in layers, for instance, in software layer secure boot and secure communication is used, in hardware layer microprocessor and secure architecture is used. It is recommended to follow strict design processes for embedded systems, the process includes 1. Requirement engineering, 2. System model design and architectural design 3. Specification of hardware and software 4. Design and implementation of hardware and software 5. Unit tests of hardware and software 6. Integration test of hardware and software 7. System test 8. Acceptance test. Hardware Security Module (HSM) is widely used in embedded security. General architecture of HSM and its safeguard for keys and crypto-processing is discussed. Some of the topics discussed in details include secure boot, secure in vehicle communication, secure infotainment ECU, and secure over the air update.

There are many future automotive security challenges which are based on Vehicle-to-Vehicle(V2V) and Vehicle-to-Infrastructure(V2I) network. In the presentation we discuss the underlying idea of V2V and need for security. Some of the recent development of V2V is discussed along with the DOT model deployment strategy.

Secure cars require a Multi-Layered Security Concept:

- **Secure in-vehicle communication**
 - **Secure external communication**
 - **Secure platforms (HW and SW)**

 - **Security mechanisms on all layers complement each other to provide a holistic security concept for enabling sustainable secure E/E architectures**
- 
- **Security mechanisms and techniques will start being employed in next generation cars going in production 2020+**

Background work

Xie et al.[1] propose that the 64-kb $\text{Cu}_x\text{Si}_y\text{O}$ -based RRAM chip for embedded key storage provides smaller area, lower write voltage and power, faster read speed, and more security features than the Antifuse counterpart. This makes it more suitable for embedded key storage. The design of the RRAM structure TaN is in M1 on the backend. It contains one extra mask to form the RRAM storage layer. The LRS is tens of kilohms, while the HRS is tens of megaohms. The max ratio of HRS/LRS at 125°C is 50x. This allows for stable operation. When first built the RRAM cell is in HRS state, however, it is then switched to a bipolar state with a voltage of 2V applied from BL to SL to switch it LRS with a reset voltage of 1V. The 1T1R cell is smaller with a lower write voltage than the Antifuse. Its size is 30 F^2 compared to 75^2 . Also, with set and reset voltage less than 3V no high voltage is required. The authors' go on to explain the fully invasive attack-resistant features of the RRAM chip. As the On-/Off-states depend on the formation of conducting filaments composed of small copper vacancies along the storage layer several nanometer in thickness it is difficult to differentiate between "0" and "1." In addition, due to random locating of the switching region it is nearly impossible to reprocess the RRAM device without damaging the storage structure. Effectively allowing the device to self destruct. Lastly, because no charge is stored in the chip, charge detection type attacks are rendered useless. Next, the authors explain the side-channel attack resistance based on the bit structure and read scheme. Authors show that the Monte Carlo simulation shows that for the 2T2R average power readings for 0 and 1 randomly at HRS/LHS of 200, there is no obvious power difference as compared to the 1T1R. In addition it is smaller at 60 F^2 and timing for reading 0 and 1 is the same eliminating time attacks. The authors' chip has nonreversible state changes back from HH to HL and LH preventing malicious writes. Low operation voltage of the RRAM eliminates charge pump and decreases write power. Removal of reset circuits provide write protection and the small size provides more chip area efficiency. The author shows that power analysis attacks are rendered useless as there was a correlation coefficient of 0.025 over the 200 measure bytes where each one stored randomly 0 or 1. Thus giving the

RRAM more security advantages and a lower integration cost than the Antifuse. RRAM has more area and power efficiency making it more suitable for embedded key storage; in addition, faster read times allow for adequate key storage.

Choo et al. [2] propose that although deeply-embedded systems are continuing to grow in popularity they still face issues pertaining to confidentiality, integrity, availability, and privacy. The authors then summarize four articles out of thirty-five submissions pertaining to possible techniques for embedded systems and cyber-physical systems that can help solve these issues. The first article, “On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age,” discusses how a group of researchers created a family of elliptic curves for resource-constrained devices and a design for scalable, regular, highly-optimized ECC library. The second article, “Towards a Reliable Detection of Covert Timing Channels over Real-Time Network Traffic,” researchers from the University of Nebraska Lincoln showed a way of detecting covert timing channels over live network traffic. The authors used three unique statistical tests to generate separate scores that would differentiate between overt and covert traffic inter-packet delays. The third article, “SMA: A System-Level Mutual Authentication for Protecting Electronic Hardware and Firmware,” researchers analyzed supply chain vulnerabilities in electronic systems. The researchers presented a system-level mutual authentication approach that allowed the hardware to detect and authorize the firmware and vice versa using the secure protocols TIDP and TIDS. The fourth article, “Don’t fool me!: Detection, Characterisation, and Diagnosis of Spoofed and Masked Events in Wireless Sensor Networks,” researchers developed a way to detect malicious interference resulting from faulty behaviors using a wavelet transform-based approach that detected malicious data input in wireless sensor networks. Lastly, the authors reaffirm that issues in embedded and cyber physical system security are still prominent. They continue to list these as examples of remaining issues:

- Advances in Health-care IT and cyber-physical medical systems security and privacy
- Green cryptography for deeply-embedded data security
- Smart building security and spatial/temporal privacy preservation
- Privacy in cyber physical systems
- Secure and trustable cyber-physical systems
- Emerging cryptographic computing schemes for embedded security
- Novel anonymous sensitive data handling and restricted computing methods in cyber physical systems
- Novel deeply-embedded computing reliability methods

Dattathreya et al. [3] propose that the current Risk Management Framework (RMF) used by the DOD for certification and accreditation of target systems does not effectively or efficiently assess automotive Embedded systems. When assessing automotive embedded systems, the main four elements that must be considered include: the threat set, the mitigation set, the cost of implementing and supporting mitigation, and the quantifiable residual risk (after implementation of chosen mitigation(s)). An important step to the controls tailoring approach used to assess automotive embedded systems is the creation of a representative threat/risk model. This allows engineering teams to optimize security controls and reduce unnecessary controls and focus on effective measures. The major threats to automotive embedded systems include: Over the air attacks, In-vehicle data bus attacks, reconfiguration attacks and control override (unauthorized software modifications), vehicle degradation attack, data stealing attack, data falsification attack,

external sensor attack, and supply chain. The overall objective is to derive and tailor the control set to mitigate the threats. Only four of the eighteen control families in the RMF deal with technical security controls: access control, audit and accountability, identification and authentication, and system and communication protection (with system and communication the most important for automotive embedded systems). The authors propose a six step technique for the technical requirements behind the automotive embedded systems:

1. Assign Important Index (It) assigned per threat (higher = more important)
2. Assign Probability Index (Pt) per threat scenario (higher = more priority)
3. Assign Implementation factor (IFt) based on standards available, threat knowledge, and applicable solutions or products availability (Lower factor are riskier)
4. Develop and rank resolution cost factor (RCt) for each threat using an algorithm comprised of It, Pt, and IFt as variables. Determine a cutoff based on funding and scope available (Lower costs are ideal)
5. For each short listed threat and scenario use the cutoff ranking to assign authentication, authorization, accounting, non-repudiation, and intrusion detection
6. Using the applicable technical standards and the tailored security controls, derive technical requirements for the shortlisted threats and scenarios

The authors remind that the factors (It, Pt, IFt, and RCt) are subjective and have values that often have bias or expert values. The factor It, Pt, and IFt are used as inputs in mathematical models to derive combinations of RCt values. The different combinations produce different RCt values that should be ranked from min. to max. among all scenarios for all threats. (i.e. higher It and Pt values with a lower IFt value produce lower RCt values.

Gu et al. [4] begin by introducing the growing problem in the Automotive industry of insecurity in embedded network connected systems in vehicles. Specifically in message authentication to prevent malicious hackers. The issue is one that lends itself to the safety of drivers as by hacking into one ECU within the vehicle can allow malicious attackers to gain control over the whole network including: braking, steering, and the vehicle engine. Because the automotive industry is mass producing, cost effective solutions are desirable. As a result, the need for an HW (hardware co-processor) for each ECU is costly and inefficient. The authors propose a solution for this using Mixed Integer Linear Programming (MILP) to optimize the process. The authors discuss two techniques for message authentication: symmetric encryption (sending and receiving of one secret key) and public key encryption (requires both a secret and public key). Both systems rely on the use of Message Authentication Codes (MAC) to verify incoming messages. This paper focuses primarily on time delayed release of keys using symmetric key encryption used in the Timed Efficient Stream Loss-tolerant Authentication (TESLA protocol). In this system, key generation, MAC signature, and MAC verification are the three security measures. Where key generation allows for each time slot to have its own unique key, MAC signature allows the first time interval to verify the entire “key-chain” using the MAC signature operation, and MAC verification refers to the time delayed verification process where each key must be verified before a within a given authentication delay. SHA-1 is used as the key generation function in TESLA.

Conclusion

In order for automotive technology to be secure, new protocols should be put in place. Security policies and procedures need to be transparent so that industry wide standards can be developed. This paper has looked at the existing security protocols for automotive embedded systems. As the connected and autonomous vehicle is becoming reality it is of high importance for organizations to address security vulnerabilities and improve security standards of vehicles. For future work, it is important the embedded devices in vehicles should have the feature of self-diagnosis, self-detection and self-warning.

References

- [1] Xie, Y., Xue, X., Yang, J., Lin, Y., Zou, Q., Huang, R., & Wu, J. (2016). A logic resistive memory chip for embedded key storage with physical security. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63(4), 336-340.
- [2] Choo, K. K. R., Kermani, M. M., Azarderakhsh, R., & Govindarasu, M. (2017). Emerging Embedded and Cyber Physical System Security Challenges and Innovations. *IEEE Transactions on Dependable and Secure Computing*, 14(3), 235-236.
- [3] Dattathreya, M. S., Bechtel, J. E., & Mikulski, D. (2016, December). On Synthesising Technical Cybersecurity Requirements for Automotive Embedded Systems. In *Computational Science and Computational Intelligence (CSCI), 2016 International Conference on* (pp. 1074-1076). IEEE.
- [4] Gu, Z., Han, G., Zeng, H., & Zhao, Q. (2016). Security-aware mapping and scheduling with hardware co-processors for flexray-based distributed embedded systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(10), 3044-3057.