Hardware Security Primitives for Vehicles

Carson Labrado, Himanshu Thapliyal

The current state of vehicular technology has the undesired side effect of introducing security risks. The push for autonomous vehicles for example will require vehicles at a minimum to be outfitted with additional sensors. Malicious actors could seize upon any one of these new features as a potential attack vector with devastating consequences. Embedded devices called Electronic Control Units (ECUs) are used to control the subsystems in vehicles. These ECUs are able to communicate with each other through a larger Controller Area Network (CAN) bus. The underlying design of the CAN bus, which is used to connect several critical systems, further complicates matters. The protocol used by the CAN bus does not include methods for message encryption or authentication. This is harmful when combined with the fact that messages within the network are broadcast to all nodes [1]. Researchers have shown that a vehicle can be physically compromised once an attacker gains access to the CAN bus such as through the on-board diagnostic (OBD-II) port that is mandatory on all vehicles. Headlines were made in 2015 when researchers were able to remotely compromise a Jeep Cherokee to the point that they could disable the brakes or kill the engine [2]. Introducing security into an inherently unsecure system is a challenge that will require the exploration of a wide range of possible solutions.

In this survey we will highlight hardware security primitives which are hardware devices that can serve as building blocks to create full-fledged security solutions. Other areas of security research such as guaranteeing security properties, modifying network protocols, and implementing cryptosystems will not be covered except for instances where they are directly applicable to hardware primitives.

I. Background

For the purposes of this survey, we will solely focus on the information as it relates to the hardware security primitives of vehicular security. Specifically, these primitives can be grouped into physically unclonable functions (PUFs) and security modules.

A. Physically Unclonable Functions

Physically Unclonable Functions (PUFs) can be thought of as a type of hash function in which a given input will result in a specific output. In PUFs, inputs are known as "challenges" and outputs are "responses". Collectively, a challenge and its associated response are known as a challenge response pair (CRP). A PUF is considered a strong PUF if it has a large number of CRPs. Ideally it would have a exponential number of CRPs so that a n-bit strong PUF would have 2^n CRPs. Conversely, a weak PUF has a very limited number of CRPs (typically just one) [3].

PUFs are designed to have a uniqueness property where each copy of the PUF will have a unique response for a given challenge. These inherent properties of PUFs make them attractive for use in vehicle security applications. An attacker would be forced to obtain the actual PUF itself if he or she wanted to use it in an attack as it would be impossible to create an exact copy of the desired PUF. Researchers have explored incorporating PUFs into such areas as key storage and generation, signature creation, and authenticity verification.

B. Security Modules

Trusted platform modules (TPM) are cryptographic co-processors designed to integrate security into larger computer systems. TPMs are physically wired into the system and are not removable. The specifications for a TPM were standardized by ISO/IEC 11889. The current standard requires AES-128 for symmetric cryptography, RSA and ECC algorithms for asymmetric cryptography, and the SHA-1 and SHA-256 hashing functions in addition to other hardware features such as a random number generator and secure non-volatile storage. Figure 1 contains a representation of the mandatory components that must be included in a given TPM 2.0 implementation.

The TPM specification was designed for a generic computing system. Directly integrating TPMs and vehicles is difficult due to the limited computing resources and real-time constraints that are uniquely present in vehicles. Instead, researchers have used the TPM specification as a blueprint of sorts to design vehicular hardware security modules (HSMs). These modules can be easily integrated into vehicles and tend to only include some of the cryptographic components that are required to be included in TPMs.

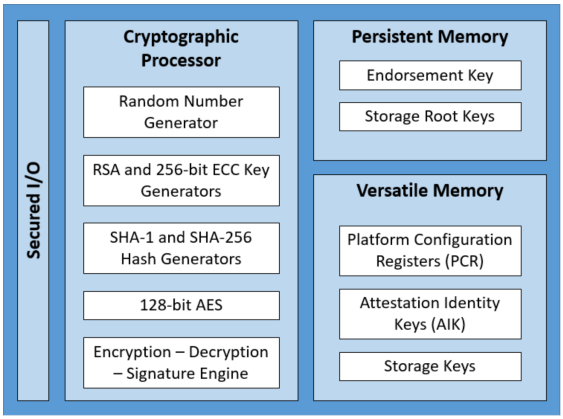

Fig. 1. TPM 2.0 Mandatory Components

II. Hardware Modules

One notable area of vehicular security research is the creation of security modules that can be introduced to the unsecured buses found in vehicles. These modules attempt to implement a measure of security without incurring significant costs. These modules aim to provide a secure hardware to facilitate secure communication between nodes in both internal and external vehicular communication networks. The unique security challenges facing vehicles prevent the direct adaptation of existing trusted platform modules (TPMs) from normal computers to vehicles. For this reason, hardware security modules (HSMs) have been proposed. HSMs are specially designed so that the modules may be effectively integrated into vehicular communication networks. However, this does not mean that vehicular TPMs will be completely unique from existing implementations. Both existing TPMs and their proposed vehicular implementations tend to share multiple hardware components such as storage mediums and crypto engines.

A. Hardware Security Modules

Researchers from the E-Safety Vehicle Intrusion Protected Applications (EVITA) project have previously proposed a vehicular hardware security module [4]. Their module is a general-purpose cryptographic co-processor designed to be connected to an in-vehicle communication network and comes in three variations: “full”, “medium”, and “light”. The “full” version has everything needed to secure vehicle-to-everything (V2X) communications. The “medium” variation is focused only on the security of communication within the vehicle. The “light” variation is designed to only secure interactions between ECUs and the sensors and actuators.

The module uses ECC-256 for asymmetric cryptography, AES-128 for symmetric encryption and decryption, and WHIRLPOOL as its hash function. The module also includes a pseudo-random number generator (PRNG), unsecured RAM, and nonvolatile memory (NVM). These features are kept behind a so-called cryptographic boundary that is separate from the actual application core that communicates with other ECUs attached to the in-vehicle communication system.

This design has since become an inspiration for more recent vehicular HSM implementations. For example, the HSM included in NXP Semiconductor’s MPC5748x family of automotive microcontrollers is a direct implementation of the EVITA spec [5]. Figure 2 contains a representation of this HSM.

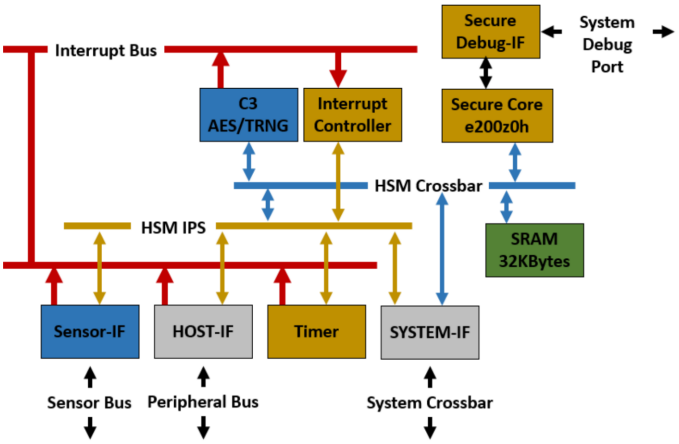

Fig. 2. MPC5748x HSM Architecture [5]

III. Vehicular Security Applications of PUFs

Physical Unclonable Functions (PUFs) have many potential uses in vehicular security and can be specially designed for use in such systems [6]. The unique attributes that are intrinsic to PUFs allow them to be used in ways that would be otherwise impossible to implement using classical technology. A few of the potential security applications of PUFs that will be highlighted in this section include key storage, pseudonym generation, and vehicle-to-vehicle communication.

A. PUF as a Secure Storage Method

Researchers have proposed using PUFs as a method for storing private keys for use in vehicular communication in [7]. The typical methods of securely storing keys involve the use of secure memory. Their method allows for secure key storage in completely unsecured memory. This is achieved by having the keys be derived from the responses of a PUF. The use of PUF ensures that an attacker would have to obtain the actual physical device in order to extract the key. The proposed methods involve using either a strong PUF or a weak PUF [7]. For a strong PUF, multiple challenges and other helper

data required to ultimately derive the key is stored in normal unsecured memory. Whenever a key is needed, the associated challenge is applied to the PUF and a key is derived from the PUF response. For a weak PUF, only a single challenge and associated helper data is stored in unsecured memory. The PUF response to that challenge is used to derive a "master seed". That master seed can then be used to further derive multiple keys. In both scenarios, the security of the keys does not come from using secured memory, but rather the fact that the keys can not be derived without access to the actual PUF.

An application of this method is shown in Figure 3 [8]. In the example, a vehicle needs to apply a software update from the manufacturer. The data is encrypted and sent from an OEM server to the vehicle. A security gateway on the vehicle is responsible for decrypting the information before allowing it to pass to the rest of the ECUs. In this way the vehicle's ECUs do not receive any external communications that do not originate from a trusted source because the gateway can block it. Traditionally secure nonvolatile memory NVM is used to store security keys. However, NVM is expensive and instead the researchers have used a PUF to help reduce the costs that would be incurred by adding a secure gateway. The response of the PUF is used as the security key. The gateway can then generate the key whenever needed instead of having to store the actual key itself.

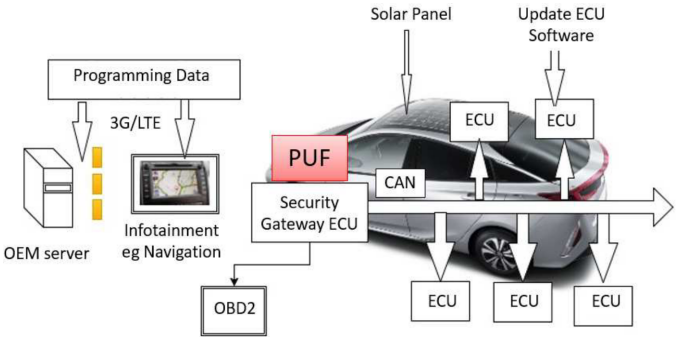

Fig. 3. PUF Integration from [8]

An alternative storage method has been proposed in [9]. In this method the PUF response itself is used to derive a pseudonym in the form of a public key and private key pair. An overall Certificate Authority (CA) issues a certificate as proof that it has verified the validity of the generated pseudonym. This certificate can then be included in communications with an external entity.

B. Use of PUFs in Communication

Approaches have been proposed in the literature for integrating PUFs with vehicular communication systems. One such approach uses an optical PUF as part of a non-forwardable authentication scheme for vehicle-to-vehicle communication [10]. Another approach is to incorporate a PUF into individual ECUs [11]. The use of PUF allows for the creation of an authentication method for connected ECUs. A built-in authentication method makes it much harder for an attacker to insert a malicious ECU into one of the communication networks on the vehicle. This would help to eliminate the threat of an attacker attaching a malicious ECU to the network that could then be used to send erroneous messages within the network.

1) PUF in Vehicle to Vehicle Communication: Researchers have proposed a way to use PUFs as part of a larger communication system that is capable of avoiding adversary coalition attacks [10]. In a coalition attack, two or more adversaries impersonate the sender and receiver by intercepting messages and then forwarding them to their intended recipients. In the presented scenario a given vehicle is communicating with other nearby vehicles. The recipients of these communications are authenticated using sensors to optically bind each communication with the correct vehicle. Visually binding to other vehicles allows the vehicle to know the relative location of each vehicle with which it is communicating. This knowledge can be leveraged to aid in the vehicle responding to warning messages from said vehicles. By forwarding messages, adversaries can trick targeted vehicles into visually binding to the adversary intercepting and forwarding messages instead of the actual intended recipient of the messages. Even if the adversaries in a coalition attack simply forward the intercepted messages without any sort of tampering, the attacked vehicle can still misidentify an adversary as the vehicle with which it is communicating. This could prove dangerous when attacked vehicles attempt to respond to warnings about emergency maneuvers such as emergency braking.

The researchers propose to create a message authentication method that is non-forwardable and therefore not susceptible to coalition attacks. Vehicles are assigned certificates from a trusted Certificate Authority (CA) which contain physical characteristic to aid in identifying the vehicle and challenge response pairs (CRPs). A vehicle wishing to establish communication sends its certificate to a given vehicle. The receiving vehicle extracts the information from the certificate and configures a laser to the challenge parameters contained in the certificate. The receiving vehicle uses its laser to stimulate the optical PUF on the sending vehicle and records the response. The receiving vehicle is able to authenticate the sending vehicle if the recorded response sufficiently matches the response contained in the certificate. The receiving vehicle then sends its certificate to the sending vehicle and the process is repeated with swapped roles to allow the sending vehicle to authenticate the receiving vehicle.

2) Integration of PUFs and ECUs: Researchers in [11] present a way of preventing Denial of Service (DoS) attacks within the CAN bus by implementing the ability to authenticate connected ECUs. The proposed method is based on assigning an ID to each ECU connected to the CAN bus. Each of the IDs are associated with specific challenge-response pairs (CRPs) of the PUFs that are incorporated into each ECU. A centralized reference monitor (RM) securely stores copies of the IDs and CRPs for each ECU within a trusted platform module (TPM). All of this data is determined and loaded when the vehicle itself is being built. Any ECU wishing to communicate along the CAN bus must first be authenticated by the RM. A similar method has been proposed [12] where the PUFs in individual ECUs are used to generate a private/public key pair for each ECU and a central server. The key pairs are used to first enroll each ECU's ID with the server. After enrollment, the server authenticates communication sessions between ECUs.

IV. Discussion and Concluding Remarks

Even the basic hardware security primitives of TPMs, HSMs, and PUFs have their own security vulnerabilities. Hardware modules can have weaknesses caused by design flaws. Recently, researchers revealed two attacks against TPMs that exploit previously unknown design flaws that require firmware updates to prevent [13]. Additionally, many common cryptographic algorithms used by TPMs and HSMs such as RSA are vulnerable to quantum attacks [14]. Post-quantum alternatives would need to be explored and included in existing security modules should quantum computers come to fruition. For example the TPM 2.0 specification allows for AES-256, but it would have to be updated to support post-quantum cryptosystems and larger hash functions such as SHA-512. Multiple PUFs have been shown to be vulnerable to machine learning based modeling attacks [15]. These vulnerabilities present additional design challenges, but not insurmountable hurdles. The primitives discussed in this work provide a solid foundation for developing vehicular security systems.

Acknowledgment

This research was partially supported by grant from National Science Foundation under Grant No:1738662. The authors would like to thank Dr. Robert Bridges.

About the Authors

Carson Labrado (carson.labrado@uky.edu) is currently pursuing the Ph.D. degree in electrical engineering at the University of Kentucky, Lexington, KY, USA.

Himanshu Thapliyal (hthapliyal@uky.edu) is currently an Assistant Professor and Endowed Robley D. Evans Faculty Fellow with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA.

References

[1] L. L. Bello *et al.*, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2018.

[2] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA 2015*, 2015.

[3] U. Rührmair and D. E. Holcomb, "Pufs at a glance," in *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014, p. 347.

[4] M. Wolf and T. Gendrullis, "Design, implementation, and evaluation of a vehicular hardware security module," in *International Conference on Information Security and Cryptology*. Springer, 2011, pp. 302–318.

[5] A. Robertson. (2014) MPC5748G gateway, HSM and secure OTA update. [Online]. Available: http://cache.freescale.com/files/training/doc/ftf/2014/FTF-AUT-F0347.pdf

[6] C. Labrado and H. Thapliyal, "Design of a piezoelectric based physically unclonable function for iot security," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[7] M. Feiri, J. Petit, and F. Kargl, "Efficient and secure storage of private keys for pseudonymous vehicular communication," in *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles*. ACM, 2013, pp. 9–18.

[8] H. Tomiyama, "Puf-based security enhancement for automotive software update," *International Forum on MPSoC for Software-defined Hardware*, 2015.

[9] J. Petit *et al.*, "On the potential of puf for pseudonym generation in vehicular networks," in *Vehicular Networking Conference (VNC), 2012 IEEE*. IEEE, 2012, pp. 94–100.

[10] S. Dolev *et al.*, "Optical puf for vehicles non-forwardable authentication," Technical Report 15-02, Department of Computer Science, Ben-Gurion University of the Negev, 2015., Tech. Rep., 2015.

[11] F. Syed *et al.*, "Authentication of electronic control unit using arbiter physical unclonable functions in modern automobiles," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, p. 112.

[12] A. S. Siddiqui *et al.*, "Secure intra-vehicular communication over canfd," in *2017 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2017, pp. 97–102.

[13] S. Han, W. Shin, J.-H. Park, and H. Kim, "A bad dream: subverting trusted platform module while you are sleeping," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1229–1246.

[14] L. Chen *et al.*, *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[15] J. Delvaux, "Machine-learning attacks on polypufs, ob-pufs, rpufs, lhs-pufs, and puf–fsms," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2019.