Electronic Design

JIM YASTIC, Senior Technical Marketing Manager,
Macronix America (www.macronix.com)

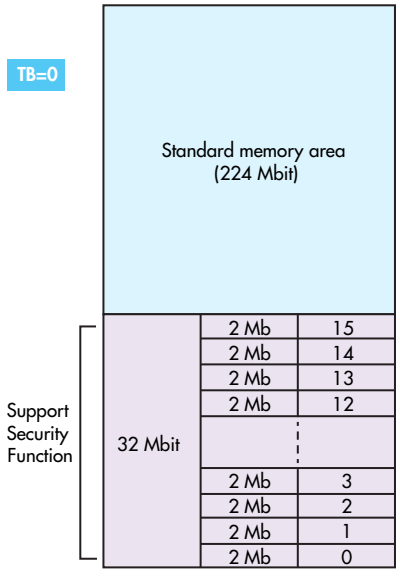
Integrating Secure Non-Volatile Memory in Internet-of-Vehicles Systems

Automotive electronics, particularly in autonomous vehicles, is skyrocketing, and advanced secure memory storage in non-volatile memory will be crucial to next-gen computing architectures for these applications.

Modern society is currently experiencing exponential growth of connected devices on public Internet Protocol (IP) networks. In 2018, the overall number of connected devices worldwide was 17 billion, approximately half of which were Internet of Things (IoT) devices, according to market research firm IoT Analytics. These numbers don't even include smartphones, tablets, laptops, or fixed line phones.[1] Targeted IoT devices range from atmospheric sensors, remote payment systems, IP cameras, smart lighting systems, and home routers to connected vehicles, which is the main focus of this article.

Connected vehicles are, in essence, connected IoT devices or edge computing platforms. In vehicles with autonomous driving-assistance system (ADAS) functionality, they could be also be considered connected artificial-intelligence edge devices. Internet of Vehicles, or IoV, is the term that describes this classification. Vehicles that fall within this definition cover a wide range of types—from ground to air, from consumer to commercial.

Non-volatile memory and security features commonly offered in these devices will be the focus here, because they're commonly targeted in attacks, either directly or indirectly. It could be by installing malware by modifying software stored in memory, or installing a sniffer device that modifies software or data in transit between memory and a host. The attack could also be the theft of sensitive data ranging from personal financial data to information that could be used to bring down the defenses of a vehicle causing unauthorized use.



Secure flash memory can be configured for two main sectors—standard memory and a secure region that's partitioned into 16 independent units for data storage.

Infusion of Features

The amount of advanced electronics in autos is growing exponentially. The main reasons for these increases are the growing inclusion of driver-assistance features that include adaptive cruise control, cameras, object identification, and notification/crash avoidance. The emission and powertrain control systems are more complex to manage hybrid propulsion. Clusters are being converted from mechanical to electronic displays along with the addition of head-up displays (HUDs). Infotainment options and connections to onboard LCD monitors are growing, too. Lighting and other environmental controls are increasing and becoming more automated. Take, for example, a rainfall-detection system that automatically activates the car's windshield wipers when it senses precipitation on the windshield.

And then there are requirements for connectivity to secure servers to download mapping and positioning information as well as entertainment content (video and audio), and to manage software updates for onboard computer systems. Moreover, security

systems to authenticate authorized drivers are getting more complex, while recording systems are being added to capture the last seconds before and after an accident to capture driving patterns, aiding law enforcement and insurance companies.

A phenomenal growth in data has accompanied these new features, and with it a need to store them. The data comes in multiple forms: at rest, in use, and in motion. Most data is in use or in transit, and a small percentage that's stored should be done so securely. However, different levels of security are required. Logging or calibration data is typically the least secure. The most secure is in the form of personally identifying information (PII), which are credentials and keys that provide access to services or levels of capability.

Opening the Door to Threats

Many potential threats to the electronics are lurking with the emergence and adoption of what's called vehicle-to-everything, or V2X, technologies:

V2I: vehicle-to-infrastructure
V2N: vehicle-to-network
V2V: vehicle-to-vehicle
V2P: vehicle-to-pedestrian
V2D: vehicle-to-device
V2G: vehicle-to-grid

As highlighted in *Table 1*, from the ETSI TR-102-638 Intelligent Transport Systems technical report[2], the connectivity options and opportunities for hacking are great.

The hacking and controlling of a Jeep by Dr. Charlie Miller and Chris Valasek, through a flaw in the Uconnect system, is a commonly cited example on the dangers of insecure vehicles. This sort of security breach should get the respect it deserves.

However, there are many practical reasons for security in thwarting more likely forms of attack. Take, for example, ensuring that odometers can't be rolled back to change the monetary value of a vehicle (fraud) or hacking into a car's electronics to take control of it (auto theft). Another not-so-obvious example is keeping a competitor from reverse-engineering and copying the design of an electronic control unit with the intent of using it in their vehicle design (product cloning).

In addition, an auto must protect against the vehicle owner or someone else accessing information she/he isn't authorized to access, such as services related to the vehicle experience like mapping, levels of autonomous service, audio, or copyrighted video content. Then there's the emergence of electronic commerce, a good example of which is the adoption of integrated toll modules to replace RFID stickers on windshields. Personal information is

TABLE 1: BASIC SET OF APPLICATIONS DEFINITION

Applications Class	Application	Use case
Active road safety	Driving assistance - Co-operative awareness	Emergency vehicle warning
		Slow vehicle indication
		Intersection collision warning
		Motorcycle approaching indication
	Driving assistance - Road Hazard Warning	Emergency electronic brake lights
		Wrong way driving warning
		Stationary vehicle - accident
		Stationary vehicle - vehicle problem
		Traffic condition warning
		Signal violation warning
		Roadwork warning
		Collision risk warning
		Decentralized floating car data - Hazardous location
		Decentralized floating car data - Precipitations
		Decentralized floating car data - Road adhesion
		Decentralized floating car data - Visibility
		Decentralized floating car data - Wind
Cooperative traffic efficiency	Speed management	Regulatory / contextual speed limits notification
		Traffic light optimal speed advisory
	Co-operative navigation	Traffic information and recommended itinerary
		Enhanced route guidance and navigation
		Limited access warning and detour notification
		In-vehicle signage
Co-operative local services	Location based services	Point of Interest notification
		Automatic access control and parking management
		ITS local electronic commerce
		Media downloading
Global internet services	Communities services	Insurance and financial services
		Fleet management
		Loading zone management
	ITS station life cycle management	Vehicle software / data provisioning and update
		Vehicle and RSU data calibration.

[TABLE 1] Connectivity options—and opportunities for hacking—are numerous.

TABLE 2: ATTACK SURFACES AND PROTECTION APPROACHES		
Attack surface	**Vulnerabilites**	**Protection approaches**
Communication channels	• Man in the middle • Snooping, injection • Weak entropy sources • Clear text	• Use of sequenced frames • Strong random-number generators • Use of encryption and authentication
Physical	• Side-channel analysis • Timing, power, EM emissions, acoustics • Fault injection (power and clock glitches) • Data remanence in memory • Brute force: probing, de-capsulation, reverse engineering, uncovering special debug modes	• Fault injection—redundancy and fault-tolerant computing • Hiding and scrambling the bus • Data encryption and de-encryption in a trusted zone • Sensor mesh in top metal layer • Fault detection – memory erase • Other anti-tampering designs
Software	• Fault injection • Version rollback • Interrupts • Buffer overflows • Malware injection • Code in clear text form over non-encrypted channels	• Fault injection • Execution redundancy • Checksums on data transfers • Randomized execution • Encryption/decryption • Creation of trusted execution zones • Designs based on compartmentalization and isolation, and access control • Root of Trust and Chain of Trust implementations
Lifecycle	• Unsecure memory provisioning environments • Code and data programmed and stored in clear text • Provisioning and re-provisioning in the field • Root-key discovery and exploitation	• Secure provisioning environments or services • Root of Trust and Chain of Trust implementations • Encryption and authentication • PKI – public/private keys, certificates, authentication • In the field secure secret key sharing

[TABLE 2] Various attack surfaces can be met with proven protection approaches.

increasingly being stored in onboard systems, such as contacts and passwords. Like a computer at home or your smartphone, this data must be kept secure and privacy must be maintained.

Finally, an important and intertwining link exists between safety and security. Applying the proper levels of security, in of itself, provides the determinism needed for safe operation. This is often overlooked when thinking about security alone. A robust security framework protects against unauthorized actions taken by individuals, while improving safety by incorporating additional controls in the system design.

General Attack Types and Protection Approaches

Three major objectives need to be met to achieve security of information in connected devices:

• *Integrity:* Ensuring that information is authentic and hasn't been compromised. In other words, providing a security feature called non-repudiation that provides proof of authenticity and origin.

• *Confidentiality:* Securing information so that it's private and can't be accessed by or made available to unauthorized users.

• *Availability:* Making sure that information is accessible when it's needed.

The best place to start in understanding device security would be to take a high-level view of it. There are general categories of attacks that are useful in understanding the threat landscape, as well as common approaches to protecting against these threats.

When designing a component for a vehicle, the potential attacker's motivations must be considered. Is it stealing intellectual property, a denial-of-service (DOS) attack, financial theft, or some other type of data pilfering? Next, the would-be attackers need to be understood. Are they government-funded, competitors or just weekend hackers? After analyzing the attacks that seem likely, are they invasive, semi-invasive, or non-invasive? Lastly, what kind of attack methods might be used and how can a secure design thwart the attack or make it too expensive to hack. In short, a thorough security analysis needs to be conducted up front.

Table 2 summarizes some of the major attack surfaces, their vulnerabilities, and protection approaches against likely attacks.

Security Features in Non-Volatile Memory

It's always assumed that the details of the cryptographic algorithms are well-known by an attacker, so it's only the secrecy of the key that ultimately provides security. Trying to keep keys secret is one of the toughest challenges in deploying security. Beside the keys, other user-sensitive or private data are usually stored at the same safety level as cryptographic keys.

In some secure execution environments, a

	TABLE 3: NON-VOLATILE MEMORY SECURITY FEATURES				
	Security features	**NOR flash**	**NAND flash**	**e.MMIC**	**Advanced flash**
Hardware	BGA package: The ball grid array under the chip protects against probing.	X	X	x	x
	Protection pin: The block-protect operation using the protection pin can protect the whole chip or selected blocks from erasing or programming.		X		
	Hardware write-protect pin: There are two versions, depends on the flash type: NOR: It protects the register settings that configure the program/erase protection of blocks and sectors. NAND: The memory will not accept the program/erase operation. It is recommended to keep "write-protect" active during power on/off sequence.	X	X		X
Software protection	Temporary block protection: This avoids accidental program/erase to specific blocks.	X	x	x	x
	Solid protection: This permanent block forbids malicious modification to block-protection configurations	X	x	x	x
	Unique ID: This is a value that is unique to each non-volatile memory device	X	x	x	x
	Password-protect block locking: This feature uses an advanced method to protect block-locking configuration from modification	X		x	
	Read protection: Protecting against data corruption	X	X		
	Sanitize: All data is physically erased to prevent data reuse.	X	x	x	x

[TABLE 3] Standard security features common in non-volatile memory are effective in protecting against threats.

dedicated secure storage is required not only for secret key information, but also for application-specific data. Furthermore, for modern operating-system architectures that support virtualization or multi-tenancy, it's a requirement to support and store the security credentials of multiple users for multiple applications. For complex systems, this secret information will be stored in external secure flash memories due to the storage size requirements.

Table 3 shows a wide range of standard security features that are common in today's non-volatile memory offerings. They're effective in protecting various threats to its contents.

Why would a product need to implement memory with advanced security features? To start with, an overabundance of existing designs is based on older microprocessors. Thus, if there's a requirement to make the product more secure, it's expensive to do major redesigns of existing products, particularly with complex applications in a hard, real-time environment that have been field-proven. If a requirement has come from an organization's marketing department that it must now be connected to an open network, where will the required keys and credentials be securely stored to support secure access?

Three major advanced security memory products are available today: RPMC, Authentication Flash, and a more full-featured secure flash. RPMC is a memory device whose sole purpose is to provide non-volatile monotonic counters for the host to support sequenced frames in secure communication protocols like TLS/SSL to thwart man-in-the-middle attacks. Several memory providers currently offer this type of flash as a dual-purpose memory. This means there is a region in flash memory reserved for RPMC functionality, with the rest reserved for normal NOR Flash. Authentication Flash devices only perform authentication with the host before a secure operation. A good example of this would be the RPMB feature in eMMC5.1. Full-featured secure flash devices perform authentication and encryption along with a full range of additional security features.

A good example of a secure flash with advanced security features is the Macronix ArmorFlash. It's a NOR flash device and incorporates all of the standard security features listed in *Table 3*, as well as advanced security features listed in *Table 4*. It has a standard SPI interface and leverages the SPI command infrastructure. ArmorFlash introduces special packet operations that provide secure read and write operations to its secure memory region for data storage.

The memory layout shown in the *figure* is one example of how it could be configured. It has a secure memory region that's further partitioned into 16 independent 2-Mbit units

for data storage. Another standard memory area of 224 Mbits for code storage is accessed directly with regular SPI flash commands after an initial gated authenticated exchange.

Advanced security flash devices have the following security elements:

Non-volatile monotonic counter

The use of non-volatile monotonic counters is the best weapon against replay attacks, even if power outages or glitches occur. This is because monotonic counter values are stored in non-volatile memory. ArmorFlash, for example, has four independent 32-bit monotonic counters, which only respond to authenticated operations. Counter configurations can be programmed to change counter behavior depending on different application scenarios.

TRNG

High-entropy random-number generators (RNGs) are fundamental to almost all secure systems because cryptographic systems depend on secret data that is only known to authorized users. There are two main uses for RNGs. The first use is in producing a random value called a nonce, used in calculating a unique MAC challenge-response authentication value. In this case, the client initially receives a random challenge, usually a message embedded with a nonce value from the host. The device uses this value as one of the inputs into a MAC calculation. The nonce value could also be provided by the memory device.

The second use is in generating encryption keys. In some implementations, a dynamic/session symmetric key is generated from random number. The true RNG (TRNG) in ArmorFlash has spatial and temporal dependence that produces a TRNG with high levels of entropy.

Key storage and management

The storage and management of keys are crucial to authentication and encryption. For example, a specific memory region in ArmorFlash is capable of storing four 256-bit keys. Each key is set up through an independent and lockable process. Several key-management commands are available to set up keys according to their intended use cases. The configuration allows for separate keys for authentication and encryption operations.

Authentication

For its part, ArmorFlash introduces a packet read/write command for various security operations. The packet command is a proprietary link-layer packet protocol that's similar to the RPMB protocol. It has its own sub-command operation codes and data structures. The design for the layer above the link layer involves a shared symmetric key used for the authentication process and a Cipher Block Chaining Message Authentication Code (CBC-MAC), which is used to produce the required MAC values. CBC-MAC is implemented as part of a block cipher algorithm called Counter with Cipher Block Chaining-Message Authentication Code, or CCM. It's a mode of operation of a block cipher algorithm that integrates both authentication of the host and device, and encryption of the payload. It's a NIST standard (NIST Publication 800-38C).

Data encryption

As part of the Advanced Encryption Standard (AES)-CCM algorithm, the data field in the packet contains the encrypted payload based on a symmetric key block cipher algorithm with a minimum block size of 128 bits. In the case of ArmorFlash, AES-256 is used, making for a 256-bit block size.

Unique ID and extra serial number

An additional 8-kbit secure one-time program (OTP) area is added for storing unique identifiers and static data according to system applications' demand. The 8-kbit secured OTP area further decomposes into two rows of 4-kbit configurations. The flash device also could incorporate an extra 8-byte, factory-coded serial number, used in cryptographic calculations to uniquely bond the host and secure flash.

Non-volatile PUF

Although programmed unique ID or serial numbers are okay for identification, a physically unclonable function (PUF) code can be used that's truly unique to each memory device. PUF is becoming common as a unique identifier or digital fingerprint

TABLE 4: NON-VOLATILE MEMORY ADVANCED SECURITY FEATURES

Security features	NOR flash	NAND flash	e.MMIC	Advanced flash
One-time programming: Protect configuration setting from others	X	X	X	X
OTP space: A space for OTP data	X	X		X
RPMB: Authenticated access	X		X	
Replay protected monotonic counter (RPMC): Monotonic counter support				X
Secure region: Authenticated access				X
Independent areas for encryption/ decryption: Support for multiple users				X
True random-number generator (TRNG)				X
Key generation				X
PUF code: Hardware feature that provides a unique sequence of values based on intrinsic process variations during the manufacturing				X

[TABLE 4] Advanced security features are found to varying degrees across non-volatile memory types.

for semiconductor devices. Devices that implement PUF take advantage of random and physical variations of a flash-memory array to produce a non-volatile PUF code. Users can leverage this code both for identification purposes and as an input for key generation.

1. Conclusion

The growth of IoV and connected transportation continues to explode, as does the associated data. Computing architectures in this space are moving to decentralized models to adapt, bringing about the growth of powerful, mobile edge computing platforms that include support for applications such as personal automobiles, autonomous vehicles, mobility-as-a-service (MaaS) systems, and other new forms of transportation like robo-taxis (ground/air).

This all has to be done safely and securely in an untrusted environment. Therefore, non-volatile memory requires a range of security mechanisms and policies to ensure identity, confidentiality, integrity, authenticity, and availability. Advanced secure memory-storage features found in non-volatile memory are crucial to achieving these objectives.

Jim Yastic is senior technical marketing manager at Macronix America, responsible for regional ecosystem, market, and business development and product definition. With 30 years of experience in the high-technology sector, his previous roles include engineering, product marketing management, and business development for organizations focused on semiconductor, communications, and embedded-software markets. Jim holds a BSEET and computer science degree from Chapman University in Orange, Calif., and an MBA degree from St. Edwards University in Austin, Texas.

References:

1. IoT Analytics, State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating, August 18, 2018, https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

2. ETSI TR-102-638 Intelligent Transport Systems technical report, Retrieved on 5/7/2019 from: https://www.etsi.org/deliver/etsi_tr/102600_102699/102638/01.01.01_60/tr_102638v010101p.pdf