Follow the money…
A path to monetizing IoT services with secure controllers and processors

Donnie Garcia
Solutions Architect for Secure Transactions
NXP Semiconductor, Austin, Texas USA

Money, the conceptual contract of entitlement to wealth used to pay debts was an important milestone for modern day civilization. Today, as intelligence is integrated into the things we interact with, there is a future where payment transactions will occur at our appliance, vehicle and home. Managing account data protection is the Payment Card Industry Security Standards Council®. The guidelines provided by this council allow developers to understand and implement highly secure devices that monetize the services which they provide. Essential for meeting the PCI standards for accepting payment cards and performing payment transactions are processors using ARM® CPUs with integrated security features that enhance trust, cryptography and tamper response, all of which are needed to implement the state of the art security for embedded systems.

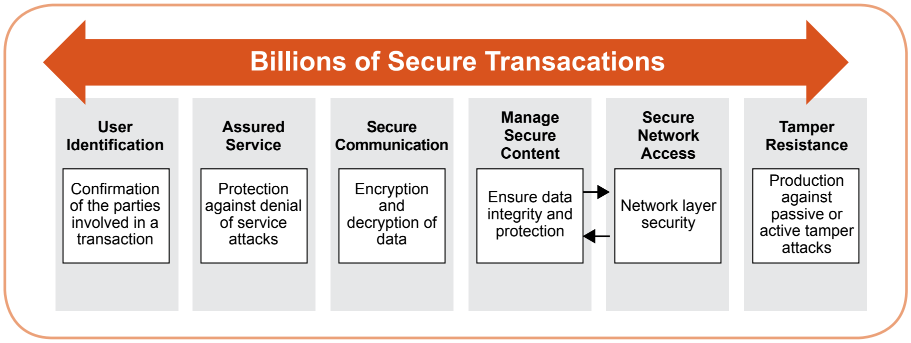
Payment applications are diverse. They can be simple pin entry devices up to the latest multimedia smart terminals you would see at a high-end restaurant table. These smart POS devices can provide a platform for advertisement in addition to completing payment transactions. For simple pin pads and slave terminals that connect to smart phones or tablets, the ARM Cortex-M® processor performance power and cost are utilized. For more advanced systems leveraging Android operating system and rich displays are processors leveraging the ARM Cortex-A® CPUs. For both cases, there are essential capabilities around anti tamper, cryptography and trust that must be implemented. This paper will overview key processor features, the software components and methods used to ensure compliance to Payment Card Industry Standards.

Keywords—Security, microprocessor, payment, secure transactions, standards, attack scenarios

I. SECURE TRANSACTIONS

So many of us use our payment cards multiple times a day. In a taxi cab, at a restaurant or at our favorite lunch time food truck. We are all participants in a secure transaction between a card holder, the merchant acquirer and the bank. In the United States alone, the number of card based payment transactions exceeded 100 billion in 2015 [1]. These secure transactions have a direct relation to value, with an average of 55USD per transaction in the United States of America [1]. As the embedded world progresses to an era where everything is smart and connected, there is a great deal to learn from this mature application space that faces a constant and growing security threat.

As represented by figure 1, a payment card transaction embodies the aspects of a secure embedded application. At one edge of the transaction you have the payment card. This node ties a unique identification to the user. For the entities involved, there is an expectation and support for assured service. Communication is secured with cryptography and physical protection schemes for communications interfaces. The sensitive data that is generated to complete the payment is managed securely and only secure network interfaces can be used. At both the payment card and the terminal, there is integrated tamper resistance to enhance the security of the application. All if these aspects of security must be present to support the secure transaction.



Billions of Secure Transacations
User Identification
Confirmation of the parties involved in a transaction
Assured Service
Protection against denial of service attacks
Secure Communication
Encryption and decryption of data
Manage Secure Content
Ensure data integrity and protection
Secure Network Access
Network layer security
Tamper Resistance
Production against passive or active tamper attacks

Fig. 1. Functional security of a payment card transaction

As with other embedded application markets, the functionality for payment terminals can vary based on several factors. The richness of human interaction supported, the payment card types accepted, and if the system is a table top device or portable. At the low end, card readers are attached to mobile devices, accepting only a single type of payment. More standard and the most common type of payment terminal is a hand held, accepting pins and working on countertops. High end systems are integrating complete cash registers along with rich displays. The system requirements for each of these devices leads to diversity in the embedded processors which are supporting them. For all form factors, industry standards require certification for the security of the system.

This landscape of embedded diversity is not specific just to the payment terminal. Looking across the Internet of Things, we see a similar variation in applications such as smart watches, thermostats, home monitoring systems and others. For embedded designers, if you are building a payment terminal or a smart watch, it is essential to understand the methods and guidelines which can be applied to protect your systems and deter costly attacks.

II. PAYMENT TERMINAL ARCHITECTURES

Payment terminal architectures can be categorized into four types. To understand the challenges of security it is important to start with an overview of the underlying functionality provided by these designs. The following sections will detail the main architectures found in this applications space.

A. Microcontroller with RTOS

The simplest payment devices supporting functions as a secure pin pad, an mPOS attaching to a mobile device or a portable point of sale devices. These are commonly built from a microcontroller architecture. For example, an ARM Cortex-M® class device running at greater than 100MHz. The embedded memory for such devices will vary broadly based on the end device functions. The general size will range from 256KB of embedded flash to 1MB of embedded flash and up to 256KB of SRAM. The following figure 2. details the most complete representation of this architecture with all embedded functionality supported.

The functions provided are human machine interface provided through a pin pad, display, status LEDs and buzzer connectivity via wired or wireless interfaces, system clocks and power and finally card reader interfaces. The card reader interfaces can include magnetic stripe, contact cards and contactless cards using near field communications.

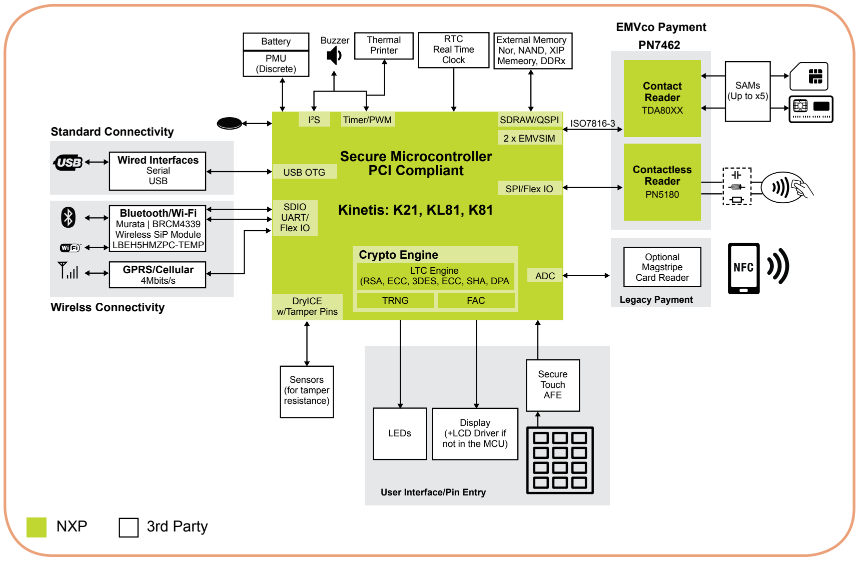


Fig. 2. Microcontroller with RTOS block diagram

B. Applications processor Running Linux

The most common form factors of table top and portable point of sale devices are built from applications processors running the Linux operating system, The functions provided by this architecture can range just as the microcontroller based systems. These can be devices that attach to the mobile device (mPOS), up through devices that work with their own modems to complete the payment transactions. ARM Cortex-A® devices are needed to support Linux. Offering more expandability to the end users This type of device is needed to allow merchant customizations and to work with payment network infrastructures. Leveraging the Linux OS, the applications processors provide a path to richer displays and more performance.

C. Applications processor running Android

A newer trend are payment terminals built from the Android operating system. These devices allow end users the same experience as they are familiar with on their smart phones with a rich 3D user interface. Leveraging Android Applications developers, this trend is bringing more functionality to the payment terminal with applications for inventory management and resource planning to supplement the payment accepting functions of the device.

D. Split Architecture:Applications Processor with Secure MCU

In some instances, the processing needs are extremely high. For example, the need to playback high definition video and support the presentation of advertisements. In this case a split architecture can be deployed. This is essentially taking the MCU system architecture and adding a high-end processor that would be typically be found in a tablet or a smartphone. The applications processor will typically be running Android. The Secure MCU in this architecture has the task of communicating with the contact and contactless readers as well as performing the bulk of the functionality to support the security of the end device. The high end applications processor provides the multimedia functionality.

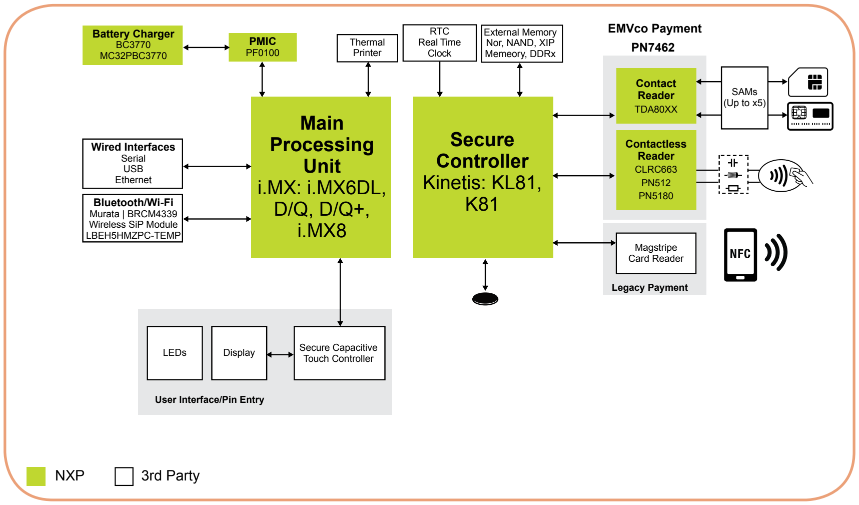


Fig. 3. Split architecture block diagram

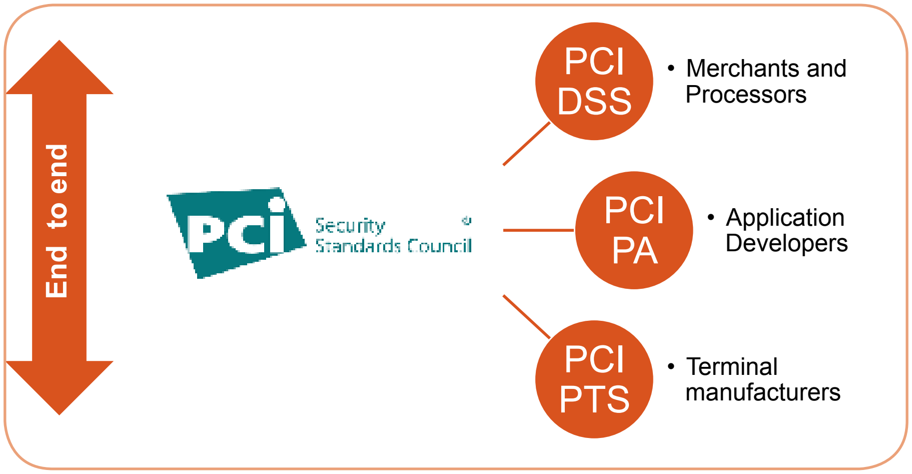


Fig. 4. Overview of PCI Security standards Council

III. SECURITY STANDARDS FOR PAYMENT

In 2006, the major card brands, Visa, MasterCard, Discover, American Express and JCB partnered to form the Payment Card Industry Security Standards Council. Driven by the need to raise consumer confidence in their products, they created the council to provide the guidelines needed for developers to understand and implement highly secure devices [2]. This standard is the predominant standard applied to payment terminals, and has been adopted worldwide. As shown in figure 4, the PCI SSC partitions their standards into three categories, guidelines for terminal manufacturers, payment application developers and merchants and processors.

Most relevant for embedded processors are the guidelines put in place for terminal manufacturers. These are the PCI Pin Transaction Standards. The PCI PTS standards are updated every three years or whenever a significant threat is present. The important changes over time indicate the growing threats to securing any type of embedded application. These include the need for third party evaluation, tamper response, requirements for secure communications, end to end cryptographic methods and the need for documenting for users how to maintain the security of the devices once they are deployed. Within the PCI PTS standards there are documents that detail the security requirements as well as guidelines for 3rd party labs to test end devices.

The strength of the standard for payment terminals makes attacks at the terminal a high effort for the attackers. The standard provides two important documents. There is a document for detailing the Security Requirements or SRs, and another document providing in depth details about the testing which the payment card accepting the device will be put through [3]. These Derived Test Requirements are applied by approved assessor labs based on the functionalities of the end device undergoing evaluation. Together the PCI PTS documentation provide both guidelines and implementation advice, raising the level of security for all payment devices

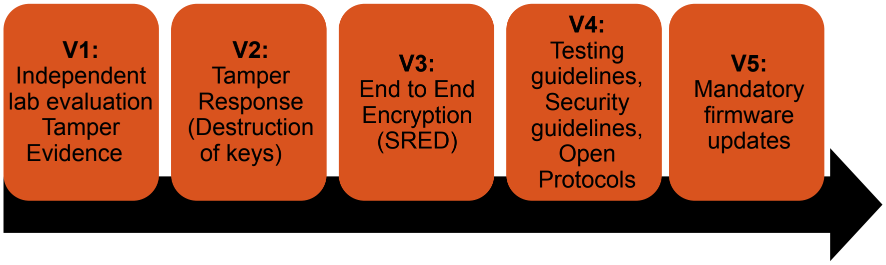

V1:
Independent lab evaluation Tamper Evidence
V2:
Tamper Response (Destruction of keys)
V3:
End to End Encryption (SRED)
V4:
Testing guidelines, Security guidelines, Open Protocols
V5:
Mandatory firmware updates

Fig. 5. PCI PTS changes over time

A. PCI PTS Evaluation

To achieve compliance to the PCI PTS standard, a manufacturer must work with an approved PCI PTS evaluation laboratory. This third-party involvement is a way to ensure that the security integration meets the requirements of the standard. When a PCI lab evaluates a payment device, they will be performing a system level evaluation that involves analyzing all aspects of the design and manufacturing of the target of evaluation. The lab will be equipped with insider knowledge that includes access to design files and firmware. The manufacturer will deliver the payment device with housing in its final form factor. The evaluator will architect attacks against the device as per the guidelines of the standard.

The PCI PTS evaluation lab will challenge two aspects of the security of the device, the physical security and logical security. Physical security are protections in place for PCB traces, resistance to overlay attacks on the keypad and even emissions or leakage from sensitive operations. Logical security protections include the firmware protections in place such as, application partitioning and how firmware updates are done.

As the evaluation lab attacks the device, they will look at attack costing, this is the measures in place to see how well the implemented security features are performing. Attack costing is based on a point system that is detailed in the PCI PTS standard in the derived testing documentation. It breaks up the attack into two phases, first the identification, then the exploitation. The attack costing considers the following aspects of the attack: Time, Expertise, Equipment, Knowledge of the device, Access to the device, and Specific parts or tools needed for the attack. There is a point system in place which identifies a score for the attack. There are requirements associated with total score and score for the exploitation phase of the attack [3]. This methodology emphasizes an important aspect of security, once an attack has been identified for a system, subsequent attacks become easier to do as the attack plan is available.

Analyzing the attack costing requirements in this standard we can see the hierarchy for the most sensitive data that must be protected in the application. The most stringent security testing will be done on the security processor and gaining access to firmware or secret keys. The keypad where the user can input their secret PIN must be strongly protected. Even the information that is presented displays must pass the PCI PTS attack costing criteria set in the standard.

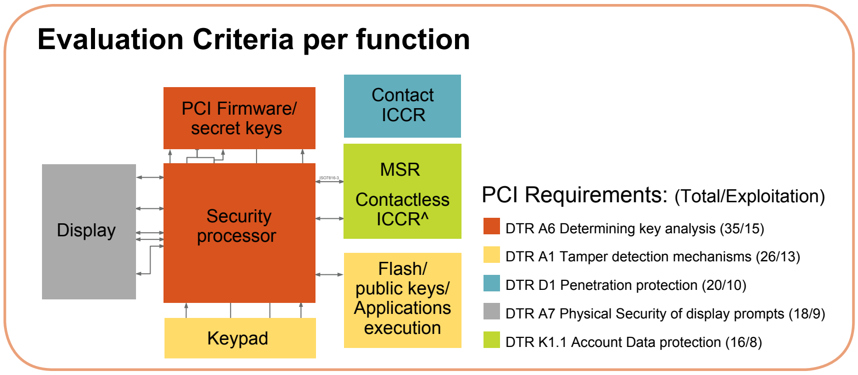
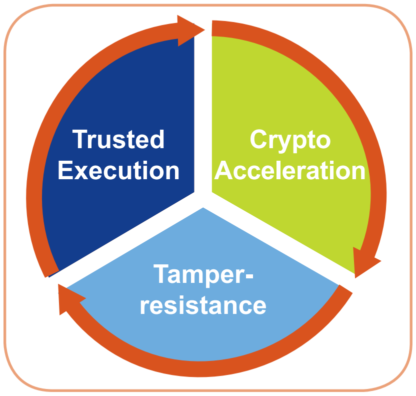

Evaluation Criteria per function
PCI Firmware/ secret keys
Contact ICCR
MSR
Contactless ICCR^
Display
Security processor
Flash/ public keys/ Applications execution
Keypad
PCI Requirements: (Total/Exploitation)
🟥 DTR A6 Determining key analysis (35/15)
🟨 DTR A1 Tamper detection mechanisms (26/13)
🟦 DTR D1 Penetration protection (20/10)
⬜ DTR A7 Physical Security of display prompts (18/9)
🟩 DTR K1.1 Account Data protection (16/8)

Fig. 6. PCI Attack costing requirements versus function

IV. SECURE PROCESSOR TECHNOLOGY

To meet the PCI PTS requirements, the secure payment accepting device must be architected with the right components. Without the security technology integration into the components, achieving the certification is not feasible. Semiconductor manufacturers like NXP Semiconductor provide secure processors with hundreds of features related to the security of the end design. These can be categorized into the following aspects:



Cryptographic Acceleration: The science of encoding and decoding data.

Trusted Execution: Allowing only what you expect from the application

Tamper-Resistance: Monitoring and protecting the system from attackers and responding if needed

These aspects of security technology work in harmony to secure the embedded system. The tamper resistance is applied to the sensitive keys and data which are processed by the cryptographic accelerators. In turn, the cryptographic accelerators are applied to ensure trusted execution by testing firmware. Finally, any issue with the execution of firmware can initiate a tamper response in the system.

To support the variation in payment terminal form factors, how these mechanisms are integrated into the secure processor varies. For the minimal functionality provided by pin pads, the processors driven by ARM Cortex-M CPUs running RTOS, will have a different level of functionality versus what is needed for the advanced architectures running Linux or Android. Understanding how these processors address the PCI PTS requirements provides the insight needed to secure all types of embedded designs.

A. Physical security implementation for microcontrollers

To pass evaluation, the PCI PTS design must be strongly resistant to a PIN disclosing bug. Because the electronics involved have digital interfaces which pass sensitive data, many designs implement a secure enclosure to protect from probing of sensitive signals. These enclosures restrict access to the ICs that are managing the secure data. These secure enclosures must follow PCI PTS guidelines for their design to protect vias in the PCB, set trace width and spacing and only have unrelated tamper signals be adjacent to one another. An example PCB design is shown in figure 7.

This type of physical barrier does not work alone. There are multiple zones of physical protections built into the designs, including sensors for detecting if the device housing has been removed. The mechanical and electrical design work together to have the housing close switches that connect protective mesh signals together.

Managing this physical protection are anti-tamper peripherals on the MCU. For NXP Kinetis MCUs this peripheral is named DryICE. The DryICE generates a random output signal

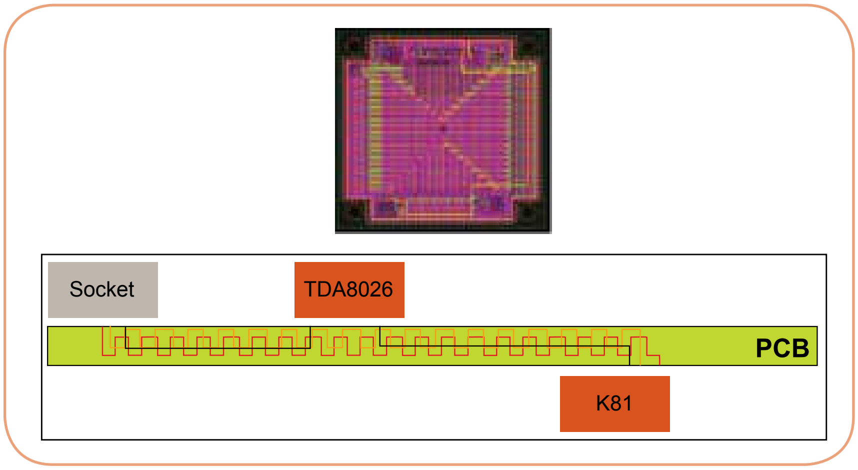


Fig. 7. PCB design and protecting the contact card interface

that is routed across the protective area. This same signal must be read on an input pin, or a tampered state is generated. These protective traces are distributed throughout the design to provide the protection for the sensitive functions such as the payment card interfaces, the display and keypad. The representation in figure 7 shows how the contact card interface is protected.

The DryICE also provides sensors for monitoring clocking, voltage, temperature and internal state signals of the secure processor. For both the analog sensors or the digital inputs into the DryICE, the response to a violation is that a secure memory space is zeroized. This operation destroys access to sensitive data by destroying the keys associated with the that data. This "Tampered" state of the payment terminal makes it unusable for completing that payment transaction.

These tamper functions are powered by an independent and secure battery embedded into the device. This is generally a coin cell battery inside the housing of the payment device. An important aspect of the tamper functionality for payment applications is the current consumption with these functions enabled. Having the tamper sensors work with very low leakage allows 3, 5 or even 7-year deployment times. For this reason, the tamper functionality implemented on the Kinetis MCU is optimized to reduce leakage and can support this essential function with only 2uA current.

B. Logical security implementation for microcontrollers

To support the PCI PTS logical security evaluations, the manufacturer must identify sensitive information and how this information is protected. This summary table which is generated as part of the certification process assist both the evaluator and the manufacturer for ensuring a thorough security implementation.

TABLE I. IDENTIFICATION OF SENSITIVE INFORMATION

Protection of sensitive information		
Sensitive information	Storage area	Method of protection
Plaintext PINs	Internal K81 RAM	Plaintext PIN is immediately erased after encryption; CPU is protected by mesh and environmental tamper detection.
Passwords	Internal K81 flash	CPU is protected by mesh and environmental tamper detection.
POI (Point of Interaction) Firmware	Internal K81 flash	Firmware is validated using a hash, and CPU is protected by mesh and environmental tamper detection.

There is explicit control of where data gets placed and most secure data is stored encrypted using the keys that are protected by strong physical security provided by the DryICE. Essential microcontroller features assisting with the logical security are Memory Protection Units, hardware firewalls and flash protection mechanisms. These capabilities work together to ensure that the identified sensitive information is protected during runtime.

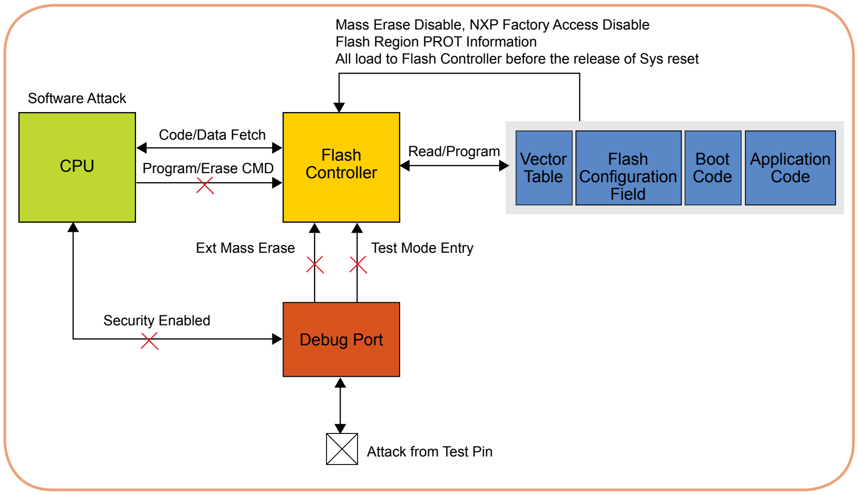


Fig. 8. System level protections for MCU

In addition, at every boot a self-test function in the main firmware runs that does a SHA-256 hash of the firmware and checks for tampers as an authenticity check. If the device is powered for more than 24 hours a device reset is forced so that the authenticity is checked again. If checks fail, then a tamper condition is forced. This design ensures that only authorized application code can be executed. The sensitive firmware is not re-programmable so the application cannot be modified.

There are also guidelines in the PCI PTS standard to detail key management. For example, bulk data encryption cryptographic keys must be separated from the financial information keys. Checks in firmware must be in place so that the key management guidelines are maintained. Part of the PCI PTS certification process will include an analysis of the key management in the firmware of the device. The evaluator will check that the key management documented by the terminal manufacturer is realized in the secure firmware.

C. Security Architecture for Application Processor

With the proliferation of mobile phones and tablets, technology adoption has pushed its way into the point of sale devices. The type of secure processor driving more capable payment devices are pushing the limits of performance and multimedia integration. These application processors by design are built to be open systems. To maintain the strong security needed for the payment device, there are many more functions done at the hardware level by highly capable subsystems. The figure 9 represents the NXP i.MX6 and i.MX7 security architecture.

The ARM Trust zone architecture is utilized to add the layer of protection for sensitive software functions. To supplement the ARM Trust Zone, there are peripherals like the CSU (Central Security Unit) that controls access from other bus masters to peripherals to provide hardware firewalls. There is monitoring by the SNVS subsystem. This is performing the similar function

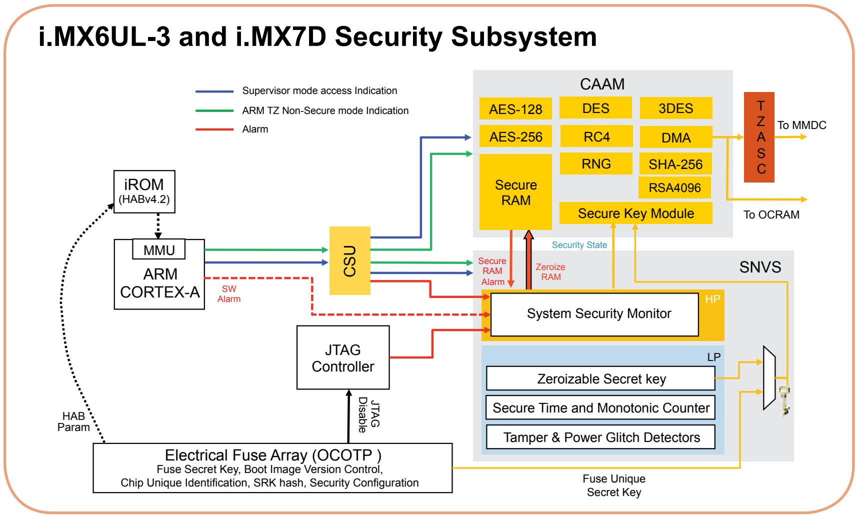

i.MX6UL-3 and i.MX7D Security Subsystem
Supervisor mode access Indication
ARM TZ Non-Secure mode Indication
Alarm
CAAM
AES-128
DES
3DES
AES-256
RC4
DMA
RNG
SHA-256
RSA4096
Secure RAM
Secure Key Module
T Z A S C
To MMDC
To OCRAM
iROM (HABv4.2)
MMU
ARM CORTEX-A
SW Alarm
CSU
Security State
Secure RAM Alarm
Zeroize RAM
SNVS
HP
System Security Monitor
LP
Zeroizable Secret key
Secure Time and Monotonic Counter
Tamper & Power Glitch Detectors
JTAG Controller
JTAG Disable
HAB Param
Electrical Fuse Array (OCOTP)
Fuse Secret Key, Boot Image Version Control,
Chip Unique Identification, SRK hash, Security Configuration
Fuse Unique Secret Key

Fig. 9. i.MX6 and i.MX7 Security Architecture

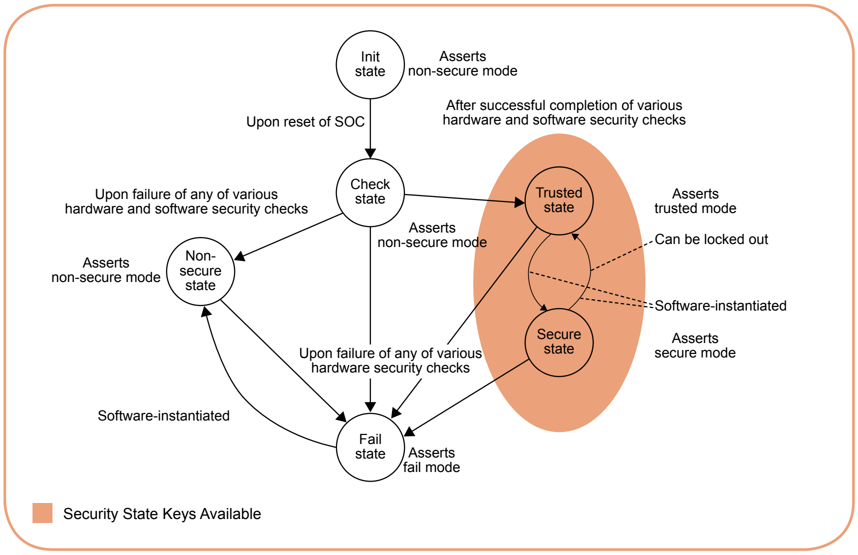


Fig. 10. Key management with chip security state

to the DryICE peripheral in the MCU. In the case of the i.MX application processor, there are alarms coming from the chip such as the JTAG controller and system watchdog.

To reduce the amount of software access to the cryptographic keys in the system there is a strong interaction between the secure key storage, the cryptography hardware and the boot up of the device. The most trusted security keys in the system are not available unless the chip boots and runs through a firmware authentication using the ROM of the device.

NXPs application processors use a block called the CAAM (Cryptographic Assurance and Acceleration Module). The CAAM by itself integrates features related to the security pillars of Trust, Cryptographic Acceleration and Anti-Tamper all within this single subsystem. With regards to cryptography, the CAAM differentiates itself versus microcontroller implementations with the algorithm support and throughput. This block has its own DMA to perform multiple jobs. The CAAM also integrates a Random Number Generator along with the logic to perform the whitening needed to meet the PCI PTS standards associated with proving the strength of the random numbers used by the system. There is a direct interface for the OTP master Key and the ZMK (zeroiazable master key) to the CAAM to perform the cryptographic operations without software passing of the keys. Again, these keys are not usable unless the system enters the secured state by authentication of the firmware.

Tightly coupled with the CAAM is up to 32KB of secure volatile memory. This memory is linked to the tamper protection features of the chip. It provides the space needed for the CAAM to temporarily hold plaintext data before it encrypts it and passes it to off chip memory. The CAAM has built in functions to create large external encrypted files based on a key that is encrypted and stored securely with the file. The only way to gain access to the data is if the security state of the chip matches the state when the file was originally written.

The CAAM also integrates a critical trust feature called Run Time Integrity Checking. This capability monitors the execution of firmware by performing a hash of the firmware as it is executed. If the calculated hash does not match the expected result, a security violation is raised. This is a way of ensuring that the application firmware has not been modified by an attack.

V. A SOLUTION OF SECURITY

To enable payment terminal manufacturers, NXP has launched the SLN-POS-RDR. This secure card reader solution based on an MCU brings together the relevant NXP technology into a package that includes collateral related to PCI PTS and EMVCo certifications. The hardware is based on a modular development platform, the Tower System [4]. These boards plug into one another to bring human machine interface with TFT color display, payment card reader support for both contact and NFC payment cards and the secure processor card based on the Kinetis K81 MCU.

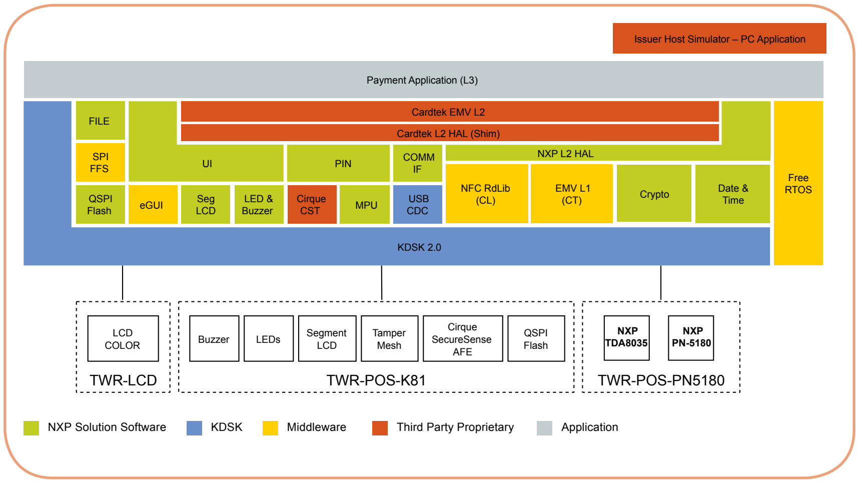


Fig. 11. MCU based solution product hardware and software

The software utilized in the system is comprised of on the shelf NXP components, like the Microcontroller Software Development Kit (KSDK) that provides the peripheral drivers. In addition, there are specific software modules built to perform the payment application and address the needs for security provided by NXP. This includes support for tamper and the memory protection unit. There are also 3rd party components with open licensing terms and components that are propriety to partners Cirque and Cardtek. Cirque provides a secure touch controller which is used to support a capacitive touch interface that can pass the PCI PTS certification. Cardtek provides the EMV L2 software that can handle the interface to the payment card brands.

The full solution package also includes all the collateral that was needed to achieve PCI PTS certification as a Pin Entry Device (PED). This includes board design files, evaluation questionnaires and evaluation reports associated with the design. Equipped with these components, the end designer can save time and money when creating their end devices in the point of sale applications space.

VI. CONCLUSION

The design and maintenance of a secure device is complex. Leveraging industry standards like those built to secure payment accepting devices from the PCI SSC provides valuable guidance. From the PCI PTS standard, we see that security must be maintained over time and that there is value in seeking third party analysis. We also learn that once a device has been compromised the barrier for future attacks is lowered, so a diligent response plan is essential for deployed devices.

The PCI PTS security requirements documents provide an indication on how to manage cryptographic keys, which cryptographic algorithms are robust enough for protecting data, and the guidance for securing communication channels. The most sensitive data in the payment application is protected by most robust requirements. In addition to the requirements are the PCI PTS derived test requirements (DTRs). Referring to the test requirements documents allow a deep understanding on both physical and passive attacks. Leveraging PCI standards developers will have insight and guidelines on how to configure their secure processors and create devices that are prepared to withstand security threats. Solution packages as the NXP SLN-POS-RDR not only include the hardware and software components, but provide the collateral for those seeking the proofs behind the implemented security.

REFERENCES

[1] A Federal Reserve System publication "The Federal Reserve Payments Study 2016", December 2016, p5, p12 (https://www.federalreserve.gov/newsevents/press/other/2016-payments-study-20161222.pdf)

[2] PCI Security Standards overview webpage , December 2016, (https://www.pcisecuritystandards.org/pci_security/)

[3] PCI Security Standards document library webpage, December 2016, (https://www.pcisecuritystandards.org/document_library)

[4] NXP SLN-POS-RDR Summary webpage, December 2016, (www.nxp.com/sln-pos-rdr)

[5] NXP Tower System Overview webpage, December 2016 (www.nxp.com/tower

How to Reach Us:

Home Page: www.nxp.com
Web Support: www.nxp.com/support

USA/Europe or Locations Not Listed:
NXP Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.nxp.com/support

Europe, Middle East, and Africa:
NXPHalbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.nxp.com/support

Japan:
NXP Semiconductor
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@nxp.com

Asia/Pacific:
NXP Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@nxp.com

Document Number: IOTSECURITYWP REV 0