

TrustZone: Integrated Hardware and Software Security

Enabling Trusted Computing in Embedded Systems

Tiago Alves and Don Felton

July 2004

The rising interest in solutions for trusted computing is largely driven by the potentially severe economic consequences of failing to ensure security in embedded applications.

Ensuring security in both wired and mobile applications has become imperative. Making an embedded product safe from malicious attacks has consequences for hardware and software design, as well as the physical attributes of the design. It is now accepted that the best protected embedded systems must have security measures designed-in from the outset, starting with the specification for the processor or CPU core.

ARM is enabling system security by integrating protective measures into the heart of its cores and providing secure software to complement the efforts of semiconductor manufacturers, product OEMs and operating system partners. For OEM partners, the issue of platform integrity has become paramount. For network operators and content providers, concerns over digital rights management (DRM) and m-commerce are growing.

Through a combination of integrated hardware and software components, ARM's TrustZone™ technology provides the basis for a highly-protected system architecture, with minimal impact to the core power consumption, performance or area. TrustZone is a safe execution environment that enables semiconductor and OEM developers to incorporate their own application-specific security measures in tandem with their own hardware and software IP. TrustZone software components are a result of a successful collaboration with software security experts, Trusted Logic, and provide a secure execution environment and basic security services such as cryptography, safe storage and integrity checking to help ensure device and platform security. By enabling security at the device level, TrustZone provides a platform for addressing security issues at the application and user levels.

Why is Security So Important?

There are many examples of the very significant costs associated with the failure of embedded systems to resist malicious attacks. These span multiple applications and industry segments, and include both direct costs and lost revenue opportunities. The need to improve security has been particularly driven by the ever-increasing spread of wireless systems that encompass data services and payment applications.

The technical issues associated with the realization of data services over mobile devices provide both a revenue opportunity, but also a threat to security. A smartphone optimized primarily for data services requires that the terminal becomes an open platform for software applications. Whilst this is essential to deliver the full range of user applications and services, it also means that the mobile device becomes more vulnerable to attack.

There are several security scenarios that are causes for concern. The first is the potential to rapidly propagate viruses over a mobile network through a user's phone book, with the worst-case outcome being denial of the operator's service – essentially bringing down the wireless network. The second threat model involves the vulnerability of end-users' private data – for example, private keys for

enabling financial transactions or banking applications, email messages and remote access to corporate networks. The inability to safely hold this type of information on a mobile terminal may inhibit the growth of such services. Viruses also have the potential to disrupt operation of the phone itself – for example, blocking calls within the radio cell.

Within the mobile phone sector, security issues with handset identity codes cost the industry billions of dollars every year. The unique International Mobile Equipment Identity code (IMEI) is a 15-digit code used to identify an individual GSM mobile, while SIMLock should ensure that a handset can only be used with the subsidizing operator's SIM card. On many handsets, both of these codes can be broken with little effort. The result of this is an opportunity for fraud to be committed on such a scale that some statistics suggest it is driving 50% of street crime¹ through mobile phone thefts.

Protection of digital content through digital rights management is another important area where security is becoming a mandatory requirement for consumer protection, as well as the protection of commercially valuable content.

The significant growth in wireless connectivity is also elevating security to the top of the list of functional system requirements. The growth of wireless LAN is one aspect of this, but the opportunity for pervasive wireless connectivity, such as offered by Bluetooth™ and other similar standards, presents a potentially more widespread security challenge. With truly mobile computing, computers are no longer restricted to equipment that users or administrators manage themselves. Consequently, security must be considered in many devices as a fundamental implementation issue.

Economic Value in Security Issues

Practically every security issue can be related back to economic value, touching every point in the industry value chain. This includes content owners and providers who need to be able to protect and charge for their content and services, and be able to take advantage of new business models; Service providers who must protect their networks against malicious use and provide efficient channels to reach end users; and the end users themselves who want privacy, protection from street and cyber crime, but with convenience and the freedom to choose their source of service.

Fraud of any kind has an economic cost, often in lost revenues as a result of counterfeiting or abuse of digital media rights. For example, telecommunication frauds are estimated to cost the industry more than a billion dollars yearly.² One of the biggest contributors to that cost is the cloning of cellular handsets.

For end users, there may be loss of personal funds as a result of electronic theft. In the wider context, research has demonstrated that enabling easy and secure payment systems can boost consumer spending on credit by up to 20 percent.

Better security will enable new revenue streams and different business models for some industries. For example, the current use of credit cards for web-based transactions can be an expensive overhead when used for very small transactions, or 'micro-payments'. Better security measures will reduce the risk of using credit cards for micro-payments, thus reducing the transaction cost. The likely outcome is the generation of new revenue streams for industries such as online gaming, where end-users will be able to buy resources to enhance their game-playing experience.

For manufacturers, security will become an issue of competitive differentiation. Handset devices with inappropriate levels of security will be left on the shelves.

¹ <http://news.bbc.co.uk/1/hi/uk/3326171.stm>

² <http://www.secretservice.gov/Telecommunications>

At a corporate level, the adoption of mobile information appliances will be limited by the ability to demonstrate protection for company assets. The use of smartphones and wireless networks in the corporate environment brings a new range of vulnerabilities. Unsurprisingly, companies have demonstrated a willingness to pay for more robust security in mobile systems.

Open Industry Issues

There are a number of widely acknowledged but currently unresolved issues around security implementation within embedded applications.

Much of the effort towards implementing embedded security solutions to date has been focused on building security features into operating systems (OS). However, the fact that OS are by definition open, and extremely complex software systems, makes it difficult to provide robust security solutions based on the OS alone.

The lack of common security elements across different platforms is obstructing the development of integrated security solutions. With no standards in place, implementation of embedded security measures has been fragmented, costly and consequently adoption has been slow. Up to now, many OEMs have developed their own software modules based on the execution of a secure execution mode outside the CPU or OS. Inevitably this approach will be less safe than a solution that integrates hardware, OS and application measures.

The implementation of security measures requires the application of techniques that can inhibit the development and debug process. Currently, some manufacturers provide special pre-production debuggable handsets to developers to help accelerate the application development process. However, this can compromise security measures if these handsets become available within the public domain. What is really required is a solution that enables debug without compromising security.

The successful deployment of trusted computing within portable and wireless equipment depends on being able to address these key open issues.

The Options for Security

There are a number of possible approaches to building security measures into embedded systems.

One option is to add a hardware security module to the design. This approach suffers from all of the restrictions inherent in any pure hardware solution. Pure hardware solutions are inflexible; they cannot easily be adapted to cater for new security functions. Obviously if an error is discovered it cannot easily be fixed without a costly design re-spin. Additionally, adding hardware IP adds manufacturing cost to the design and can have an adverse affect on power consumption.

Off-chip hardware, such as co-processors and storage, offer another approach to embedded security, enabling the acceleration of demanding cryptography algorithms, for example. However, adding a second processor to the system adds to the cost, complexity and power budget. Additionally, this approach may not provide the fundamental level of security required in the CPU processing and operating systems. The nature of the physical implementation means that traffic may be exposed between the core processor and the off-chip device, and it may not be possible for the CPU device to ascertain the integrity of the off-chip device – it may be removed and interfered with. Performance may be an issue, as with any off-chip processing.

SIM cards have a role to play in securing wireless embedded systems. The strength of the traditional SIM card in enabling security within the handset is predominantly in guarding against physical attacks. There are two opposing trends in SIM card development. One trend is towards more functionally-capable SIM cards, or 'Super SIMs', containing larger memory and having more processing power.

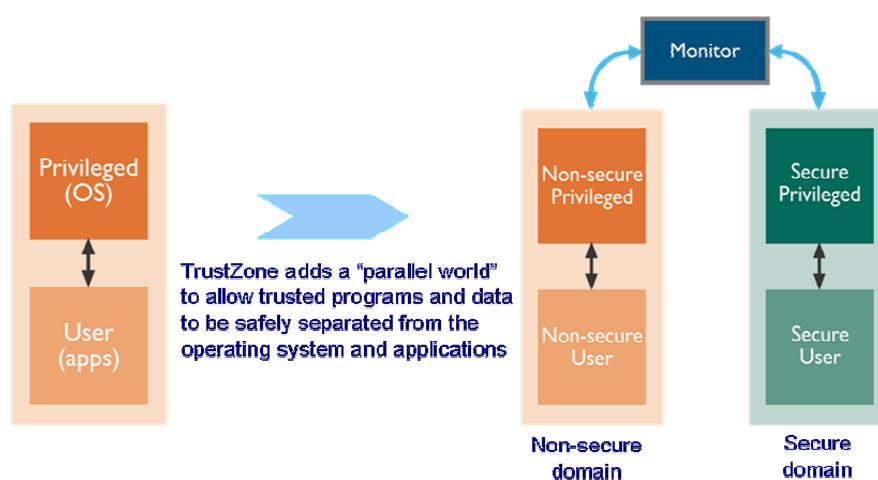
The other trend suggests a move towards smaller SIMs with more compact form factors, which consume less power and are cheaper to produce.

In its current form the processing that is possible on the SIM card will be restricted by the card's throughput and bandwidth. SIM cards cannot perform a security role at full processor speed. The primary role of the SIM card in contributing to system security is likely to remain the protection of more valuable security codes and keys.

ARM Approach – The TrustZone Solution

ARM's approach to enabling trusted computing within the embedded world is based on the concept of the Trusted Platform. TrustZone consists of a hardware-enforced security environment providing code isolation, together with secure software that provides both the fundamental security services and interfaces to other elements in the trusted chain, including smartcards, operating systems and general applications.

TrustZone separates two parallel execution worlds: the non-secure 'normal' execution environment, and a trusted, certifiable secure world.



TrustZone's Parallel Secure World

Key Benefits of TrustZone

TrustZone offers a number of key technical and commercial benefits to developers and end-users. These include:

- Primarily, TrustZone provides a safe environment for secure data on the chip. This enables a complete approach to security. For example, processing secure keys from a secure SIM card using the SoC CPU can only be performed safely if there is a safe area within the SoC. An unsecure OS is insufficient to enable this.
- Performance is an issue in some secure systems, especially in configurations where traffic must be encrypted between the core processor and an external store. With TrustZone, full bus-bandwidth access is provided to all storage areas to provide fast memory access speeds. In addition, safe local cache data is stored securely in decrypted form providing even faster access. The encrypted data can access the same FLASH memory as the non-secure world, ensuring cheap, large and flexible storage is utilized.

- Because the TrustZone solution consists of software and hardware elements, it provides flexibility to allow customization and upgrades to the secure system even after the SoC is finalised.
- TrustZone defines a secure world within the embedded system. This can include direct peripheral channels, the user interface, SIM and smart cards as well as audio output. For the non-secure world, TrustZone can enable security through integrity checking for all the features within a SoC device. For example, decoded DRM audio can be protected as it is passed to non-secure audio drivers by integrity checking the relevant part of the OS infrastructure.

TrustZone Reduces Development Risk and Cost

In developing TrustZone, ARM has recognised the importance of providing a safe solution that is capable of addressing many of the important industry security issues identified earlier. Primarily, this means protecting user and platform secrets. A solution that is capable of providing a basis for secure platform standardization, and at the same time enables differentiation, is most likely to achieve success.

The TrustZone solution has been architected to reduce implementation risk, time-to-market and development costs. By providing both the hardware and software in a pre-tested and integrated solution, the development team is free to concentrate on implementing the application-specific security functionality that is most relevant to their particular product.

The software elements of TrustZone have been co-developed with security expert partner company Trusted Logic S.A. This has resulted in code that is tightly-integrated with the TrustZone hardware architecture, yet founded on a well-established and mature code base. Another benefit of this approach is that the Security Module by Trusted Logic is available for all existing ARM CPUs, even the ARM7 and ARM9 family CPUs that do not have TrustZone built-in to their instruction sets. In these cores, the Security Module provides protection against software based attacks where its evolution, the TrustZone-optimized version, will provide greater levels of protection against software and hardware attacks when running in TrustZone CPUs.

Together, the integrated hardware-software TrustZone solution provides a number of key security features, including: platform identification and authentication, identity, key and certificate management, low-level cryptography, I/O access control, safe data storage, smart card access and code/integrity checking.

TrustZone as a Trusted Execution Environment

TrustZone provides a trusted execution environment for embedded applications. TrustZone becomes the foundation for other complementary security solutions, such as protected operating system features and hardware encryption engines. TrustZone is not intended as a substitute for features such as these, but it can enhance their security performance. TrustZone is not limited to a single security application such as cryptography services, but can be widely applied to many security requirements.

Security attacks are not limited to open systems where operating systems are in use. Within the automotive market, most of the embedded electronics is closed, or deeply embedded, and these kinds of systems can also benefit from improved security. For example, odometer fraud, where the mileage reading is rolled back before selling a vehicle, costs car consumers millions of dollars in inflated vehicle prices³. TrustZone can enhance the security of products across all market segments,

³ <http://www.nhtsa.dot.gov/cars/rules/regrev/evaluate/809441.html>

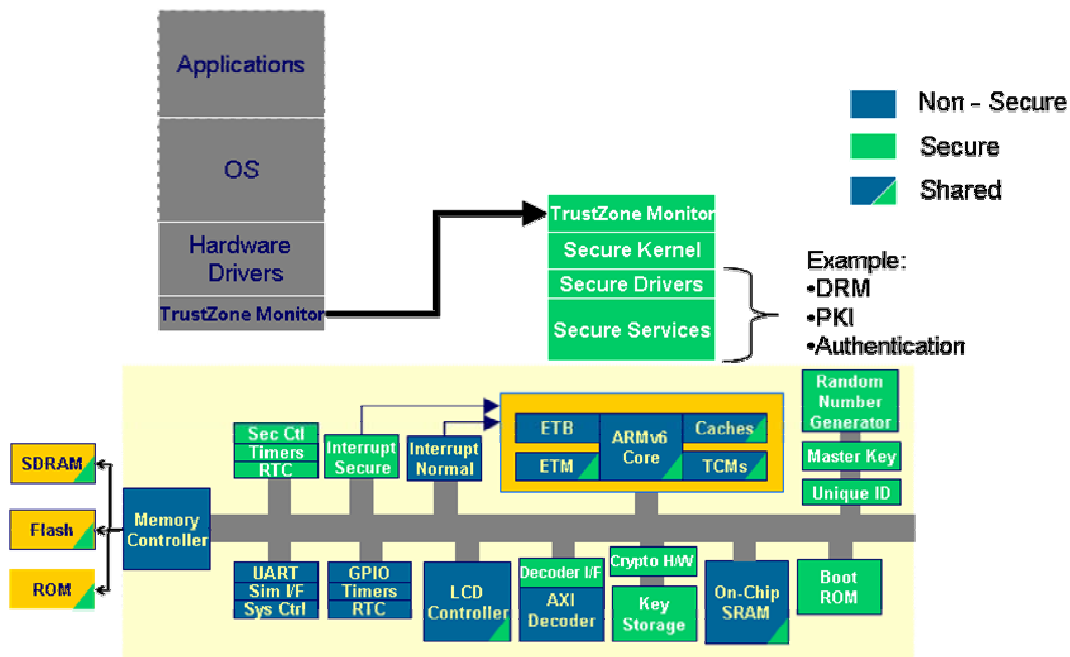
including automotive systems, consumer entertainment systems, hard disk drives, in fact any embedded application that can benefit from a 'trusted environment' capability. Requirements include enabling more protected processes for flash updates, enhancing debug, as well as running sensitive applications. It can enable all peripherals to easily achieve the standards for trusted devices that groups such as the Trusted Computing Group's (TCG) Peripherals Work Group⁴ are striving for. Similarly, although TrustZone has been designed to provide higher levels of security in complex open systems, simple systems with less rigorous security requirements can also benefit.

As well as providing full on-chip security for a SoC device, TrustZone can also be extended to enable security on systems that utilize off-chip memory. While this architecture is inherently less safe from physical attack than a system that uses on-chip memory – for example, it can be removed and interfered with; TrustZone can nevertheless enhance the overall security of such systems. Although the architectural aspects of TrustZone are implemented within the latest ARM11 CPUs, the Security Module brings the TrustZone framework to all the ARM CPUs through a set of common APIs.

The level of security provided by TrustZone is extremely high. However, unless specific manufacturing steps are taken to guard against physical attack, no secure system can be guaranteed to be unbreakable against very sophisticated and sustained attacks. ARM's goal is to raise security to the right level when considering future threats while not forgetting the economic and practical aspects of systems implementation.

TrustZone Operation

TrustZone operates by enforcing a level of trust at each stage of a transaction, including system boot. The trusted code will handle tasks such as the protected decryption of messages using the recipient's private key, and verification of the authenticity of the signature based on the sender's public key. TrustZone does this by executing secure commands within a parallel trusted execution environment.



System Example: Partitioning Secure and Non-Secure Worlds

⁴ <https://www.trustedcomputinggroup.org/about/>

TrustZone introduces a new secure state to the ARM architecture for both User and the existing Privileged modes. This determines whether the system is operating within the Secure or Non-Secure World. A new mode – Secure Monitor, controls switching between the Secure and Non-Secure World. The new instruction, SMI (Secure Monitor Interrupt) provides the main route to change Worlds.

A TrustZone-based SoC implementation will consist of both secure and non-secure elements. Key components include:

- a TrustZone CPU that is used to run trusted applications isolated from normal applications, and to access the memory space reserved for trusted applications,
- secure on-chip boot ROM to configure the system,
- on-chip non-volatile or one-time programmable memory for storing device or master keys,
- secure on-chip RAM used to store and run trusted code such as DRM engines and payment agents, or to store sensitive data such as encryption keys,
- other resources, such as peripherals, that can be configured to allow access by trusted applications only.

Industry Standards Collaboration

ARM is working with a number of standards bodies to ensure wide compatibility with the TrustZone infrastructure. The Trusted Computing Group is one example, working towards security standards in PCs, cellular handsets and peripherals.

The Trusted Platform Module, or TPM, provides the basis for propagating security throughout the system. The TPM enables authentication of the platform's secure state to third-party applications via a 'tree of trust'. This concept relies on having isolated code, outside of the standard operating system, that can be assigned a guaranteed level of security. This is the basis for TrustZone's operation, which can be leveraged to implement software instantiations of TPMs.

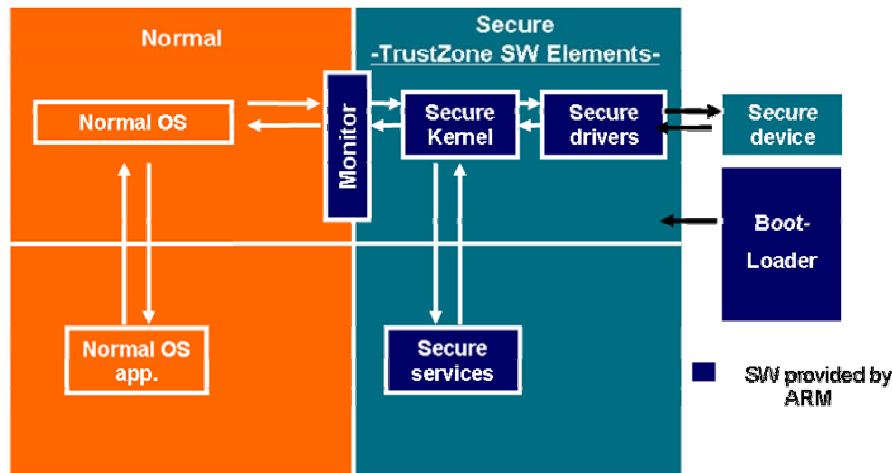
The Trusted Software Stack, or TSS, is normally resident in the non-secure world. This is a standard API for talking to and using the facilities routed in a TPM. Other APIs may be required to provide interfaces to specific cryptography functions or security measures provided within the host operating system. Both the TPM and TSS concepts are formally defined aspects of the TCG standards.

TrustZone Software Elements

Software for a TrustZone-enabled device consists of both non-secure elements, such as the normal OS and applications, and the protected software components. The TrustZone-optimized secure software components include the Monitor software, which enables the interface between the Secure and Non-Secure worlds, the Secure Kernel, Secure Drivers and Boot Loader, and basic secure software services that will be provided by ARM as part of the software solution.

The TrustZone-optimized software is an evolution of the Security Module developed by Trusted Logic, which operates as a secure kernel. This can be ported to any ARM CPU, and provides security roadmap compatibility for future TrustZone devices. The Security Module by Trusted Logic and the TrustZone-optimized software enjoy the same security protocols, which means that secure applications that are developed for the Security Module will be compatible with TrustZone-enabled devices.

The Security Module and the TrustZone-optimized software feature an independent and certifiable secure framework. It has exclusive access to dedicated protected memory, dedicated persistent storage, SIM card, crypto-accelerators and a possible trusted user Interface. By way of security services, it provides integrity checking (SIMLock, IMEI protection, secure boot), access control, secure storage and cryptography services. Future services may include frameworks for DRM, digital signature and e-banking.



Elements of the TrustZone Software

TrustZone can have multiple layers of API depending upon the target application requirements. A number of key APIs will be made publicly available to assist in the proliferation and standardization of the TrustZone solution. These include:

- **TrustZone Generic API** provides a simple message-passing interface. It is designed to enable low-level communication across the security boundary.
- **TrustZone Security Channel API** is a more tightly-defined interface API designed to allow access to commonly-available security functionality that resides behind the TrustZone security barrier. This API can be expanded with proprietary extensions as appropriate for tasks in the Security Module.

For developers working within the Security Module, two further APIs will be available:

- **The Security Module Internal API**
- **The Security Module HAL API**

These provide access to the internal workings of the Security Module to allow development or porting of function-specific drivers and task modules. This may include real-time DRM codecs, proprietary encryption protocols and proprietary secure communication protocols.

Enabling Secure-Aware Applications

The TrustZone environment enables security measures to be applied at many levels within a complex embedded system.

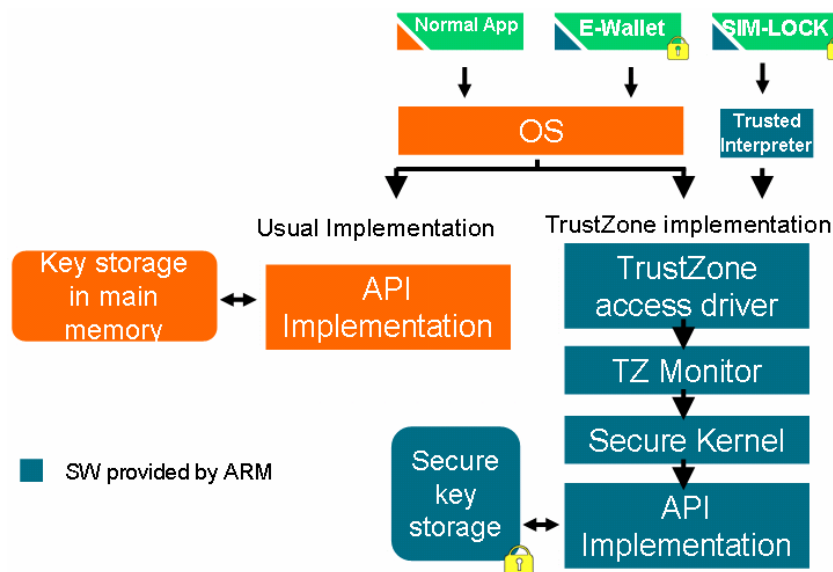
Non-secure operations will be run completely within the OS with no help from TrustZone. Although the OS may have its own security measures, securing a complete OS to the standards of the security certification authorities can be impractical. To enable security within the OS, TrustZone can provide integrity checks against attacks in three ways. First, TrustZone can verify that the OS is unaltered before booting it. During run time, TrustZone verifies that critical paths remain unaltered. Finally, a restricted set of approved functionality can be executed safely within TrustZone – a private space remote from the main OS.

Different operations have different requirements for security services, placing varying demands on the CPU and TrustZone execution environment. While data in a protected format (e.g. encrypted) can be passed around by an OS with little risk, issues occur when something has to be done with that data – i.e. it is used. Any time that secure data is actually transformed there is a danger of it being intercepted.

For data such as a secure key, it would be unacceptable to allow non-secure exposure for even one CPU cycle, as that would risk the key being captured through the use of a logic analyzer. Processing such data should be kept entirely within the TrustZone environment.

For something like the PCM audio output from a DRM codec, it might be acceptable for the output to be exposed for short period before the integrity checker blocks it. For this type of secure-aware application, TrustZone supports integrity checking to provide confidence that the output chain has not been tampered with. Integrity checking and most of the security services provided by the TrustZone-optimized software can be implemented in one of two ways. A simple TrustZone access driver will provide a communication channel for the integrity checking to be performed with complete independence from the OS. Alternatively, an in-built OS security feature, such as a cryptography API, can interface directly with TrustZone to facilitate integrity checking or the usage of any other service.

For service operators wishing to perform protected transactions such as over-the-air upgrades, it can be desirable to have a security facility that is completely independent of the handset and the OS. This could be enabled by using a Trusted Interpreter (based on the Small Terminal Interoperability Platform specifications - STIP byte-codes), similar to the current mode of operation for the SIMcard.



Application Deployment Scenarios

The Application Deployment Scenarios diagram illustrates three alternative security deployment scenarios. The normal application is shown running directly on the OS in the non-secure world. The e-wallet application is secure, and goes through the OS, which calls the access driver which switches to the secure world. When the kernel receives the request, the API manages the secure key storage. The SIMLock scenario demonstrates how the secure operation can be enabled directly through the Trusted Interpreter, bypassing the OS completely.

Designing with TrustZone Technology

The design of protected systems must be approached in such a way that security issues are considered from the outset, including the implications for the control of protected code during the development process.

Key questions must be addressed before undertaking the design, in order to specify the elements of the design chain, the components to enable the entire solution, and the potential architecture decisions and trade-offs.

- o What level of security is required?
 - Fully on-chip SoC
 - On-chip SoC but signed code from off-chip SoC
 - Software-only protection so can run fully off-chip SoC
- o How do you control the development of protected code?
 - Who holds the on-SoC Master Key?
 - Who authors the on-chip SoC boot code?
 - What other key management is required for trusted developers working behind the TrustZone security barrier?

Other industry intellectual property, or proprietary components, may be required to fulfill specific implementations. This may include DRM IP, on-chip ROM and other off-chip security resources such as cryptography accelerators.

As with any complex SoC design, there are architectural parameters and hardware-software trade-offs to be made. These are determined by the security requirements, for example:

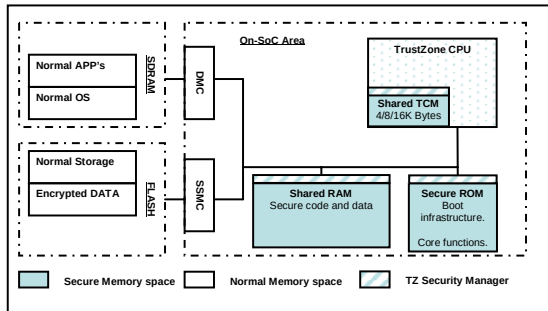
- o On-chip RAM is expensive
 - If the main concern is software attacks then off-chip execution is acceptable given suitable memory partitioning.
- o On-chip ROM is inflexible.
 - So the ability to load code into protected RAM needs to be considered
 - Such code must be authorized and signed/checked in some manner.
- o Secure Peripherals mean extra code for drivers in the protected space.

Generally extra code is to be avoided wherever possible, but when it must be added, there are three possible options:

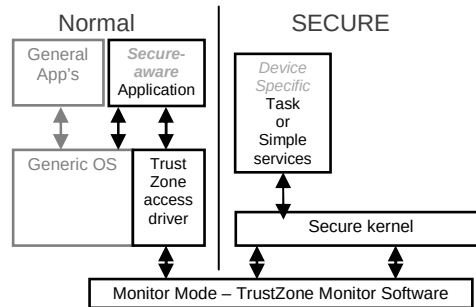
- o The peripheral driver code can be transferred from the non-secure OS to the secure space, and a simple interface driver placed in the non-secure world that communicates with the secure driver.
- o The code can be duplicated between two worlds and a handshake system arranged for control of the resource.
- o For interrupt-generating resources, for example keyboards, the interrupt can be re-directed to the secure world. This causes the non-secure driver not to be called and so the handover is transparent.

The following two examples demonstrate two design extremes; a small SoC system with a simple software architecture and a large SoC system design with a complex software architecture. For the best levels of protection, all code would be held in on-chip ROM or RAM, and all data within on-chip

RAM. However, this may be unacceptable in terms of design flexibility and cost. Code space may be limited and flaws in ROM code cannot be corrected after manufacture.

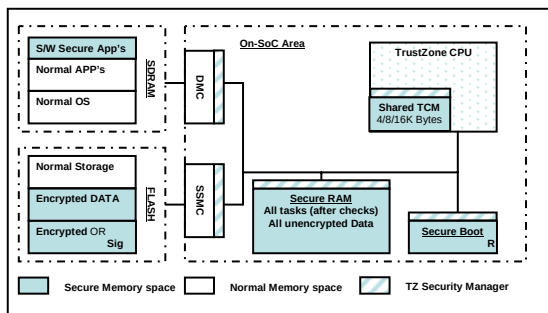


Small SoC System Example

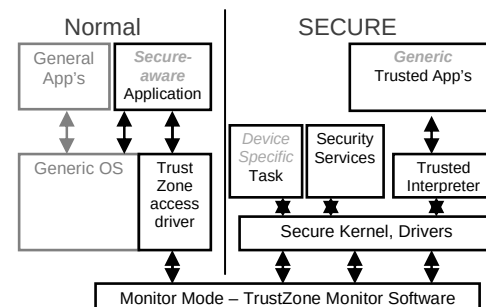


Simple Software Architecture

In the large SoC design, full off-chip execution is only suitable for environments where run time hardware attack is of limited concern. It should be acceptable to load signed code blocks from FLASH to on-SoC RAM for execution.



Large SoC System Example



Complex Software Architecture

Secure Debug

Debug is clearly a challenge with all secure systems. An essential part of the development process, it nevertheless has the potential to become a backdoor to security breaches once the device is in production. Debug may be restricted to user mode debug access for authorized applications only, or full system debug access, or alternatively no debug access may be provided at all. TrustZone technology enables full system debug in development, but provides the facility to completely disable debug of the secure domain once the device is shipped.

TrustZone Product Configuration

The TrustZone licensing model provides flexible access to the key components of the secure technology. TrustZone-enabled CPUs, such as the ARM1176JZ-S, provide high levels of security. In combination with TrustZone Software, rapid deployment of complete security applications is possible. Because TrustZone is an open architecture, independent developers are able to develop their own security software to run within the TrustZone execution environment. With this in mind, the TrustZone

optimized software is licensed as an optional element designed to bring concrete design benefits to customers.

For ARM7 and ARM9 family CPUs, the Security Module by Trusted Logic is available to provide a software-only security facility, incorporating the TrustZone abstraction layer and APIs enabling a smooth roadmap evolution. When optimized for the TrustZone-enabled CPUs, the TrustZone Software provides an even more secure execution environment. A Trusted Interpreter is available as a separate software component. Both the standard Security Module and the TrustZone Software are available with and without the Trusted Interpreter.

The product roadmap for TrustZone focuses on providing commercial availability of the Security Module for all ARM CPUs before the end of 2004, with full TrustZone Software available during the first half of 2005.

TrustZone: Enabling Platform Integrity and Application Security

TrustZone is a secure execution environment that enables semiconductor and OEM developers to incorporate their own application-specific security measures in tandem with their own hardware and software IP. TrustZone provides the building blocks for a complete secure solution, helping partners to differentiate their products through unique security implementations.

Through a combination of integrated hardware and software components, ARM's TrustZone™ technology provides the basis for a highly secure system architecture, with minimal impact to the CPU power consumption, performance or area.

TrustZone facilitates integrity checking and other security services to help ensure device and platform security. By enabling security at the device level, TrustZone provides a platform for addressing security issues at the application and user levels.

By taking advantage of its market-leading position and working with key industry partners, ARM will facilitate the growth of a viable community of developers around the common TrustZone framework. A common framework for embedded security will help to avoid industry fragmentation, and enable partners to benefit from a strong knowledge and code base. The widespread adoption of the TrustZone execution environment will bring economies of scale to the industry, and ensure that time-to-market benefits and cost efficiencies are ultimately available to semiconductor, OEM, OS and operator companies alike.

In summary, TrustZone offers a more secure solution from a trusted environment that provides a safe initialization to the secure world, with benefits that include:

- Easier to certify software applications.
- Implementation of flexible system-wide security, without constraints.
- Basis for consistent OS support – a step towards CPU security standardization and all the economies of scales that bring to the industry.
- Software compatibility between different TrustZone-enabled SoCs.
- Lower cost in terms of added hardware and software.
- Minimum impact on system performance.

TrustZone technology enables a flexible and modular approach to security – designers and manufacturers can implement the security measures they need in the parts of the system where it is required, knowing that the security foundation is in the core of the system.