

Increasing System Security by Using the DS5250 as a Secure Coprocessor

This application note describes how to use the Dallas Semiconductor DS5250 High-Speed Secure Microprocessor as a secure coprocessor to protect passwords, PINs, encryption keys and other critical data. The DS5250 employs triple-DES bus encryption, elaborate tamper sensors, and battery-backed SRAM to form a cryptographic boundary that shields valuable intellectual property. A user-configurable external tamper sensor allows the system architect to extend the DS5250's security shield to include the application's enclosure, protecting all of the system components if desired.

Introduction

Many embedded systems are constructed using general-purpose computers. The computers are based on standardized architectures with proven designs and code legacy. Although these features may be attractive to the designer, such systems are not constructed with data security in mind. This disqualifies them from a significant portion of the embedded systems market, namely applications that process sensitive or confidential information. Embedded systems such as financial terminals, identity authorization, and any system that stores passwords and encryption keys require strong measures to safeguard their data. In addition, physically exposed applications such as automatic teller machines (ATMs) are at risk. It is obvious that a need exists to combine data security along with the convenience of general-purpose computers.

The most common attempt to add security to general-purpose computers is to contain the system within expensive physical enclosures. Constructing impenetrable physical barriers is impractical for most applications, however, and it must be assumed that given time any level of physical security can be defeated. Experts in the field of data security assert that the only true protection for private data is to surround it with a cryptographic barrier. In addition, the system must be able to detect both physical and cryptographic tampering and quickly erase its memory contents as a tamper response. A tamper reactive cryptographic boundary makes a safe cocoon for secret keys, program and data.

This application note describes how to protect general-purpose architectures by adding a secure coprocessor such as the Dallas Semiconductor DS5250 High-Speed Secure Microprocessor. The use of a secure coprocessor divides computing functions within the system between secure and non-secure functions. Secure functions are those that involve the use and protection of encryption keys, passwords, Personal Identification Numbers (PINs), and other intellectual

property. System functions that do not require security are delegated to the non-secure processor. The addition of a secure coprocessor to a non-secure system increases the level of system security without forcing an architectural redesign.

What is a Secure Coprocessor?

Simply put, a secure coprocessor increases the security of a system containing other major components. The secure coprocessor assumes responsibility for security-related tasks in a system, allowing the non-secure processor to perform the primary system functions. This separation of functions simplifies the design process and increases system performance. The application can be roughly divided into secure and non-secure functions as shown below.

Non-secure functions	Secure functions
Operating System	Secure communications with network
Displays	Monitor enclosure for tampering
Keyboard scanning	Storage of passwords, encryption keys
General purpose I/O	General purpose data encryption
Magnetic Stripe reader	Public Key authorization
Printer	Secure program/data upload/download

The non-secure processor can be an ARM, SHARC, Power PC, etc., and can use nonvolatile memory such as flash or EPROM. There will typically be a dedicated serial communications channel between the non-secure and secure processors to transmit status information and data. The system must be carefully designed so that the secure processor cannot be tricked into revealing protected information by a compromised non-secure processor.

The primary function of the secure coprocessor is to provide a place to store passwords, PINs, and private keys. Such a device must be specialized, incorporating strong cryptography functions, yet be flexible enough to support a variety of system-level security functions. The most important secure coprocessor features are summarized below:

- Provide a secure vault for confidential data including passwords, PINs, and encryption keys. Memory holding this data and external memory buses must be encrypted using strong cryptography, such as DES or similar algorithms. This memory protection allows the secure coprocessor, and the application as a whole, to function as a trusted witness in secure transactions.
- Must use active tamper detection to detect intrusions both physical (probing) and environmental (voltage or temperature modifications). It should also be able to detect exterior level tamper attempts, such as breaking into the case.
- Must use active tamper reaction once an intrusion is detected. Once a tamper event is detected, the secure coprocessor must be able to zeroize / erase critical data very quickly to prevent any chance of recovery by an adversary. The only memory technology that can be erased in this amount of time is SRAM.

- Must be able to detect code substitution and prevent many side-channel attacks. Once detected, the device must execute a tamper response.
- Security features, including tamper detection and memory zeroization, must occur even if primary system power is removed.
- Should have encryption hardware available to the application software to support system level encryption needs.

Now that the requirements of the secure and non-secure processors have been described, it is possible to visualize an embedded system. The block diagram of such a system follows in Figure 1. It shows the general purpose CPU connected to the system peripherals. The general-purpose CPU has its own dedicated program and data memory, which does not contain confidential information. The secure coprocessor interfaces to the secure communications channel and also performs cryptographic operations in support of the general-purpose processor. The cryptographic protection of the secure coprocessor forms a cryptographic boundary to protect all the critical data against theft. In addition, any system-level cryptographic computations performed for the general-purpose CPU can be securely performed inside the cryptographic boundary. This includes modulo-exponentiation operations needed for RSA calculations. In a properly designed system, secret or protected information is never available outside the cryptographic boundary as plaintext.

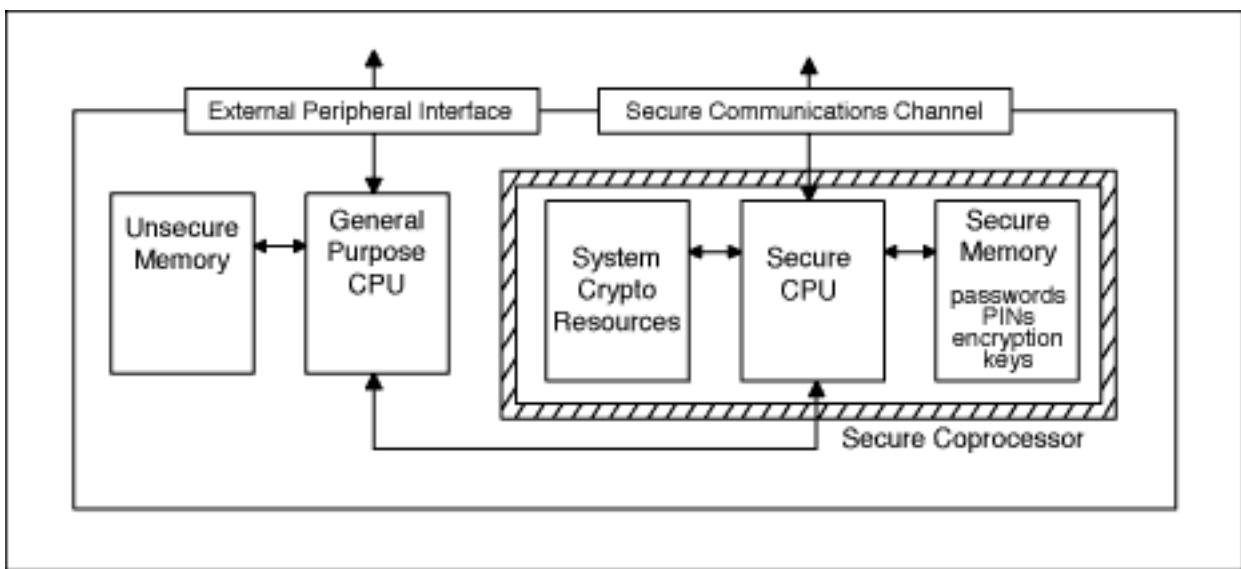


Figure 1. Block Diagram of Secure Embedded System

Security Features of the DS5250

One secure coprocessor that meets these requirements is the DS5250 High-Speed Secure Microcontroller from Dallas Semiconductor. A member of the Secure Microcontroller family, it is a highly secure, 4 clocks-per-machine cycle, 8051-instruction-set-compatible microprocessor. It was designed to be the cryptographic engine of PIN pads, financial terminals, and any other application where data security is paramount. A key feature of the device is that it encrypts its program memory with a hardware-based single or triple (3) DES (data encryption standard) algorithm, making it nearly impossible for an attacker to extract information or secret keys.

Optional byte-wise encoding is available for data memory. This makes the device ideal for storage and transmission of passwords, personal identification numbers, encryption keys, and other highly confidential information.

A truly robust system should go even further than passive encryption, and must be smart enough to detect a tamper attack in progress and react appropriately to defend the crucial data inside. The DS5250 incorporates an arsenal of sensors to detect attempts to physically compromise the die, as well as high/low manipulation of environmental conditions. The program memory can be optionally protected by a block checksum to detect attempts to force random instructions. When any of these attacks are detected, the device executes a destructive memory reset, which causes it to erase its internal encryption key, all its internal MOVX SRAM, and in some circumstances even the external encrypted SRAM. Once the internal encryption key is erased, the external encrypted memory is unintelligible.

In addition to providing strong cryptographic protection of its own memory, the DS5250 has resources available to the application software that can be used to increase system-level security. A modulo arithmetic accelerator supports modulo-exponentiation operations, an integral part of RSA (public/private key) calculations. A dedicated user-accessible DES engine allows the application software to perform customized single or triple DES operations. As a result the application software is flexible enough to interface to a variety of secure networks.

Expanding the Fence

In addition to its direct cryptographic defenses, the DS5250 can extend its security cloak to the system. The DS5250 has a self-destruct input pin that can be tied to user-defined external tamper sensors. When activated, this pin will cause the erasure of the internal encryption keys and memory of the microcontroller. In this way adding a secure coprocessor extends the tamper detection capability of the DS5250 to include the general-purpose processor as well. Now, attempts to gain access to the general-purpose processor will render the system inoperative. The simple design of the self-destruct input makes it easy to add tamper sensitivity to the enclosure of an application. Some of the sensor types and their uses are listed below. It is possible that multiple sensors may be used for added protection.

Contact Sensor

Contact sensors can range from simple limit switches to complex designs that break traces of conductive ink when the enclosure is opened.

Light Sensor

Simple optical sensors inside the case can detect light when the case is opened, triggering a tamper response.

Pressure Sensor

Breaking the seal of a pressurized enclosure can be the stimulus that activates a pressure sensor.

Motion/Tilt Sensor

Depending on the application, the motion of the enclosure may signify a tamper attempt and can activate a motion sensor.

Building a Secure Embedded System

Below is a sample application utilizing a DS5250 Secure High-Speed Microcontroller as a secure coprocessor for a financial terminal. The application drives a number of external peripherals under the control of a central operating system. The application must communicate confidential information over an encrypted communications channel using public key infrastructure (Pki). The application is required to protect the private key of the terminal, the public key of other devices on the network, as well as PINs and passwords.

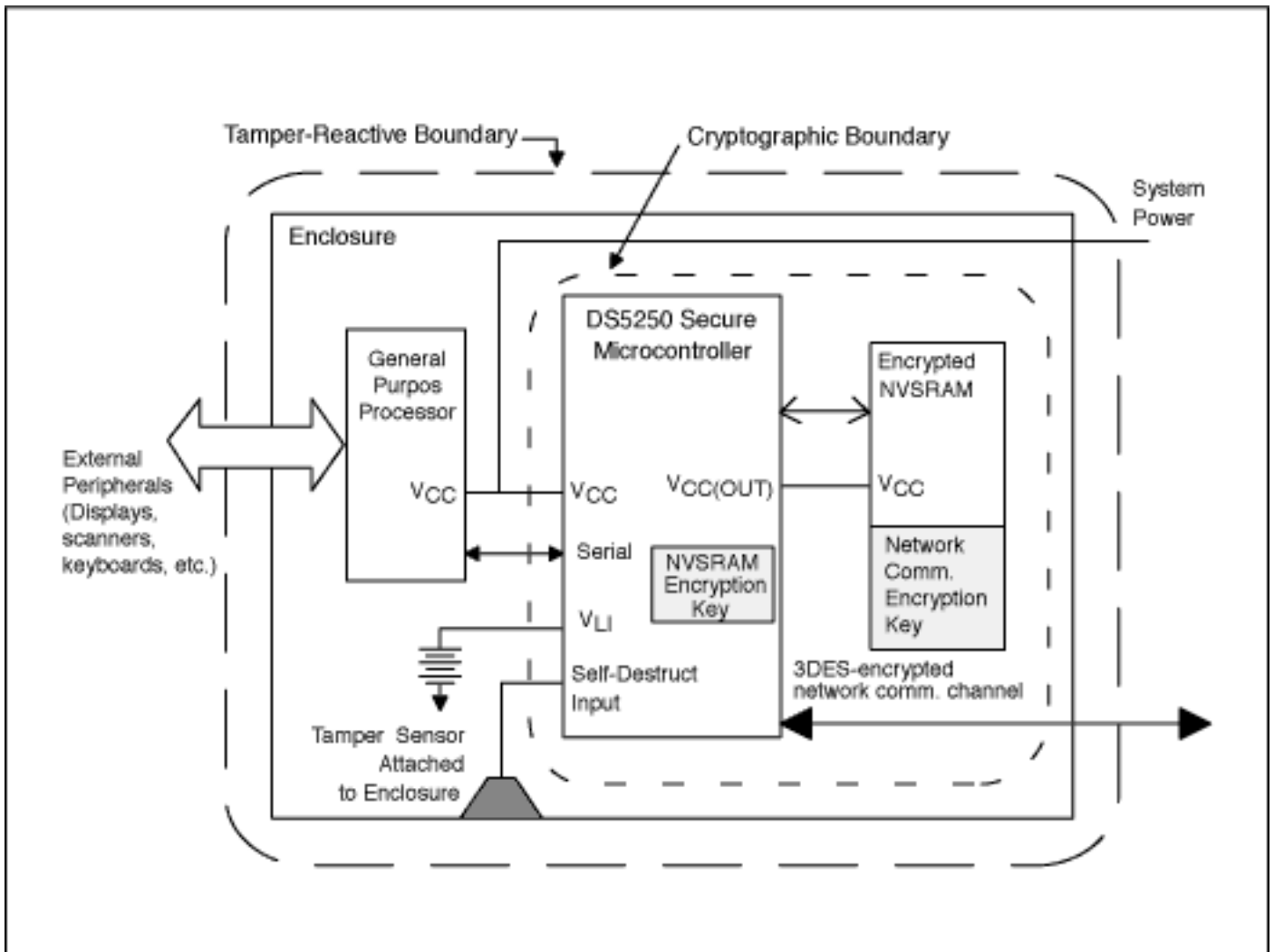


Figure 2. Secure Embedded System Using The DS5250

The general-purpose microcontroller's primary function is to execute a high-level operating system. This processor drives system functions such as sophisticated graphics displays, and communicates with outside peripherals such as scanners, printers, and keyboards. The processor communicates with the secure microcontroller using a generic serial protocol such as simple (RX/TX) asynchronous UART, SPI, or SCI, minimizing any compatibility problems

between microcontroller architectures.

In this application the DS5250 is the secure coprocessor. The secure coprocessor is responsible for performing secure communications, as well as protecting critical information such as encryption keys, PINs and passwords against theft. This is accomplished by storing the essential information in the 3DES-encrypted, battery-backed SRAM. The internal SRAM affords the strongest data security, as it is the most strongly protected. The encryption and other security features form a cryptographic boundary around the microcontroller and its memory that cannot be breached without instantaneously destroying the data.

The encryption key for the DS5250 memory is contained within the DS5250 itself, which is protected against probing and other physical attacks by a sophisticated array of voltage/probe/thermal sensors. Because the external program and data memory is encrypted, it and the internal SRAM of the DS5250 are contained within a cryptographic boundary.

The application interfaces to the network via a 3DES encrypted communication, enabled by the user-accessible DES engine of the DS5250. This provides a secure communications channel for financial transaction information. The application software determines the communication format, so it can be flexible enough to interface with almost any network.

Summary

A need exists to be able to add security to designs based on non-secure architectures. Physical security is a sub-optimal solution, due to its expense and low level of effectiveness. By segmenting the embedded system into non-secure and secure functions it becomes apparent that the addition of a secure coprocessor to existing designs can increase the security of a system without requiring a redesign of the basic architecture. The secure coprocessor acts as an nearly impenetrable vault for critical data such as encryption keys, PINs and passwords. The DS5250 Secure High-Speed Microcontroller from Dallas Semiconductor is one such secure coprocessor. By encrypting its external program memory with the DES algorithm, the operations and sensitive data of the processor are unintelligible to eavesdroppers. A variety of tamper sensors can detect intrusion attempts and wipe the internal memory and encryption keys of the microcontroller, rendering all of the data useless. The DS5250 supports a variety of external tamper sensors, which can detect attempts to penetrate the enclosure and access information. In this way the tamper resistance of the secure coprocessor can be extended to the entire application, and protect the general-purpose processor as well as the secure coprocessor.

More Information

DS5002FP: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

DS5240: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DS5250: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)