Keywords: secure, POS, tamper, encryption, 3DES, tamper reaction, security

APPLICATION NOTE 3824

Security in Embedded Systems

May 26, 2006

Abstract: Encryption alone is not enough to protect sensitive information. This application note describes why the key should be protected. The DS5250 is the best choice for this true tamper detection, providing the highest level of security for your embedded system.

Security in embedded systems is usually an afterthought. Engineers design products to get to market quickly, saving a security upgrade for a future revision. This is not illogical behavior, because products with a higher level of security can be more expensive and later to market.

Many systems, however, need a high level of security at the outset. Sometimes the requirement for security derives from the government or a trade organization. The PCI requirements drawn up by credit card companies Visa and MasterCard, for example, provide a detailed description of the security required in a point-of-sale terminal or a PIN pad. In other cases a design incorporates security to protect revenue flow. A secure application can impede reverse engineering, prevent a device from being copied, or provide true tamper detection.

But what is it that a secure microcontroller really does, and why is a secure microcontroller so crucial to sensitive applications?

A System Is Only as Secure as Its Key

Security is not accomplished by encryption alone. While the choice of encryption algorithms and key management routines are critical, they are usually not the weak link in a secure application. Imagine that Alice and Bob each have a secure phone that can only communicate with one another. The encryption implemented on the phone is practically unbreakable, and would take a century to break with all the computational power in the world. What is the weak link? The phones. If an attacker gains control of one of the phones, he or she can pose as Alice or Bob and immediately gain access to their secret information. The attacker would not even need to steal the phone, but simply to install a listening device without Alice and Bob's knowledge.

In this scenario, the encryption was not defeated, but rather the encrypting device's security, or "key." In embedded systems, the key is almost always a large, secret number that can be used by a cryptographic routine to encrypt information or authenticate data. A secure embedded system's most important job is, therefore, to protect that secret key. If the system comes under attack, the key must be erased to prevent it from falling into the hands of an attacker. The destruction of the key renders the device inoperative, and prevents an attacker from gaining access to sensitive information such as bank account numbers and passwords.

Key protection requires that the secret key never leave the confines of the embedded system, as this

would provide an easy way for an attacker to defeat the device's security (**Figure 1**). A dedicated memory inside the design does not work, however, for key storage because transactions between the microcontroller and the memory could be watched. The best security is one that requires the key to stay inside the processor using it to encrypt or authenticate data. This means that the system's microcontroller needs internal, nonvolatile memory.

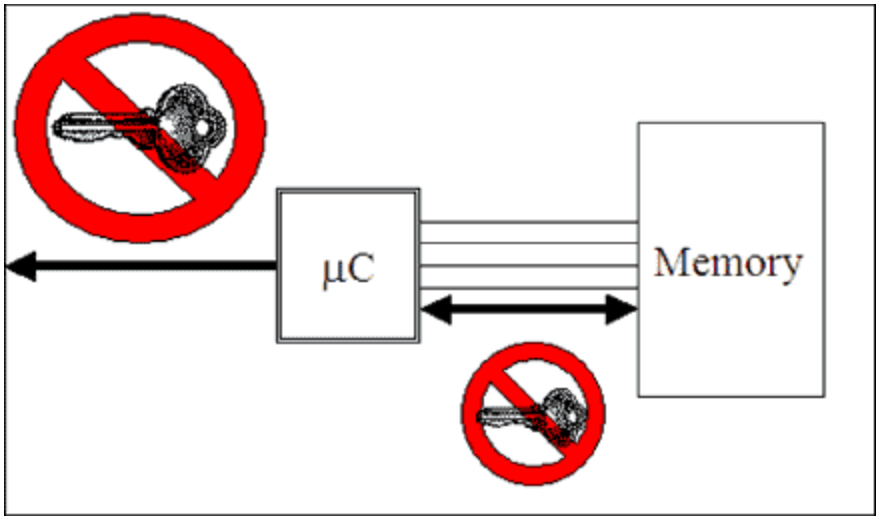

Figure 1. Secret key data should not leave the device or even be transferred between ICs.

Trip Wires and Plastic Protection

Even with key data stored only on the microcontroller, attackers can still discover secret information. For example, if an attacker can access the microcontroller's address and data buses, he or she could insert instructions to dump the key data to an external I/O port. A more sophisticated attacker could actually remove the plastic packaging from the microcontroller and use a microprobe to read the internal memory contents. A secure system, therefore, needs some way to impede this access, and even signal the microcontroller to erase its memory contents.

One simple approach to this security challenge is to seal the entire "sensitive area" (i.e., microcontroller, clocks, memories) in a tamper-evident material, perhaps by filling an area of the PC board with plastic or covering it with a metal box. Trip-wire devices can be used to detect high temperatures or the enclosure's removal. That detection would be useless, however, if the microcontroller is in a low-power state and cannot take action.

DS5250: Real Security for Key Protection

The DS5250, Dallas Semiconductor's state-of-the-art secure microcontroller solves these problems and helps systems achieve a high level of security, sufficient for use in government and financial applications. It all starts with memory.

NV SRAM

The DS5250's internal nonvolatile SRAM (NV SRAM) provides the perfect storage for sensitive information and encryption keys. This custom, low-leakage SRAM meets two critical requirements:

1. The data must be nonvolatile. Using a small, inexpensive battery, key data is maintained for several years.

2. The data must erase quickly. The DS5250's SRAM instantly erases when any of the chip's tamper-detection circuits are activated.

Battery-Powered Tamper Detection and Reaction

In addition to NV SRAM, a secure system requires sensors to detect an attack. The DS5250 has multiple battery-powered tamper sensors. The microcontroller does not need to be in an active state to react to a tamper event.

The DS5250 can detect fault-injection attacks with its on-chip temperature and voltage sensors. When the operating voltage or temperature passes outside the microcontroller's operation range, the DS5250 instantly erases its internal NV SRAM. This action eliminates the possibility of a hacker recovering any key data. To prevent microprobing of the NV SRAM cells, the top layer of the DS5250's silicon implements an ultra-fine mesh. If traces of that mesh are shorted, the DS5250 triggers a self-destruct and the key data is erased.

The DS5250 also has inputs that enable external circuits to trigger a self-destruct. This allows a system to implement multiple layers of security. The types of external circuits that can trigger a self-destruct are limitless. Some of the more common external sensors include:

1. Switches on an enclosure to detect entry.
2. PC board traces that are broken when a covering is removed.
3. Light sensors to detect when a case has been opened or is being examined.
4. Pressure sensors to detect that a pressurized seal has been broken.
5. Motion sensors for devices that should not be moved in their normal course of use (e.g., a usually stationary cash-dispensing machine put into motion).

Encrypted Code Space

During initial system loading, the DS5250 uses a random 3DES key to encrypt its instruction code before the code is stored in an external memory. This prevents an attacker from inserting malicious code into the DS5250 for execution, and also resists any attempt to reverse-engineer the application. Integrity checks can also be inserted in the code, thus detecting an attacker's attempt to alter the program code.

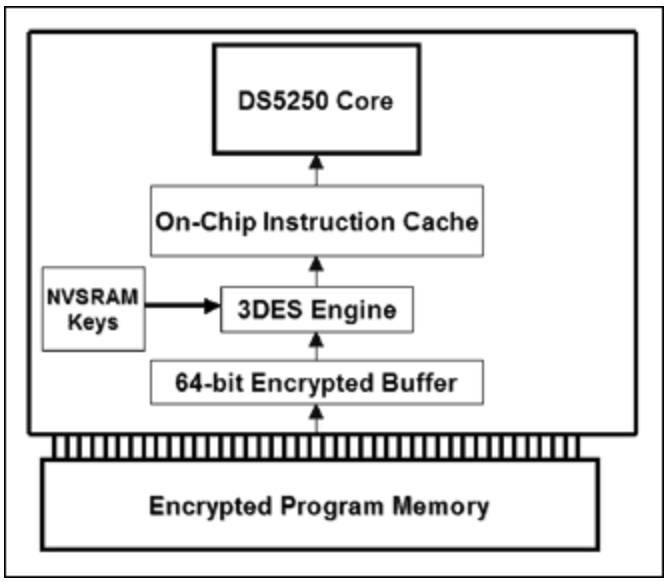

Figure 2. Encrypted code space safeguards the algorithms and data in an external memory space.

Encrypted code space not only prevents an attacker from reverse-engineering an application, but it also prevents someone from copying the device. Because the encryption keys are randomly generated on each DS5250, no two systems have the same data stored in their external flashes. The external flash would only be useful if an attacker knew the encryption key, but we have already seen that the DS5250 does not give up its secrets easily.

Engineering for Security

Designing secure systems is a challenging task. Trying to enhance existing designs is even more challenging. Key protection is the most critical part of a secure systems design. The DS5250 is designed specifically to safeguard the key, and thus provides the highest level of security for protecting any sensitive data. For more information about the DS5250 and our secure microcontrollers, go to Secure Microcontrollers.

Related Parts	
DS5230	IP Security Microcontroller
DS5250	High-Speed Secure Microcontroller

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 3824: http://www.maximintegrated.com/an3824
APPLICATION NOTE 3824, AN3824, AN 3824, APP3824, Appnote3824, Appnote 3824