Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

IBM Offers Chip-Level Security

SecureBlue Technology Aims to Make Security Ubiquitous in SoCs

By Tom R. Halfhill {5/8/06-01}

In the digital age, embarrassing security breaches are becoming commonplace. A laptop computer with information about nearly 200,000 current and former Hewlett-Packard employees was stolen from Fidelity Investments. Flash-memory drives containing secret military intelligence were pilfered from a U.S. Army base in Afghanistan and openly sold in street bazaars. And, worst of all, Paris Hilton's cellphone address book was leaked on the Internet.

IBM's Technology Collaboration Solutions Unit has an answer: SecureBlue, a new security technology for system-on-chip (SoC) devices. SecureBlue uses industry-standard cryptography to protect the confidentiality of sensitive data, but it goes much further. It can also authenticate the integrity of confidential data and protect mission-critical systems by detecting and responding to many different kinds of tampering. Depending on how extensively an SoC developer implements SecureBlue, it can stop untrusted software from executing, detect whether a system has been compromised, and even delete sensitive information and disable the system if someone tinkers with the hardware or probes it with noninvasive tools such as x-rays.

SecureBlue isn't a CPU architecture or microarchitecture. Nor is it specific to any particular CPU architecture, such as PowerPC or x86. Instead, it's a processor-agnostic security framework that includes public-key cryptography, proprietary methods for accelerating block ciphers, synthesizable logic for implementing special hardware, analog sensors for detecting intrusions, and true random-number generators for creating keys and other secure data structures. IBM is willing to license all this technology to customers that want to implement SecureBlue on their own, but the company would rather bundle SecureBlue with IBM design services and manufacturing at an IBM foundry.

The long-term goal is to make SecureBlue nearly as ubiquitous as silicon. It's not just for protecting government computers or corporate records. IBM's vision is that someday virtually all SoCs, even those in mundane consumer-electronics products, will armor themselves with chip-level security like SecureBlue. A cellphone address book or a digital-music collection deserves protection, too.

Security for the Masses

Broadly speaking, encryption is a way of exchanging a large secret for a small secret. The large secret is the sensitive data, which could be anything from Paris Hilton's address book to the maintenance manual for a stealth bomber. Encrypting the data with public-key cryptography exchanges that secret for the smaller secret of a mathematical key.

A vital feature of SecureBlue is that it securely stores cryptographic keys inside an SoC and prevents anyone from tampering with them. Another important feature is that SecureBlue can use secure-key storage, hardware acceleration, and special metadata to guard virtually unlimited amounts of encrypted information in off-chip memory—with minimal storage overhead or loss of performance. So the large secret can be quite large indeed.

IBM derived the chip-level version of SecureBlue from an existing board-level module for PCs, workstations, and servers. As Figure 1 shows, IBM's 4758 is a sealed PCI card that plugs into an industry-standard PCI slot. The card adds secure data storage and cryptographic acceleration to a system

that's otherwise unsecure. The U.S. and Canadian governments have certified the 4758 card for FIPS (Federal Information Processing Standards) 140-1 Level 4, the highest level of the FIPS 140-1 cryptographic standard. This security level is sufficient for sensitive but unclassified information.

Inside the 4758 card is a 99MHz 486-compatible processor, an IBM UltraCipher Engine (a proprietary IBM cryptographic coprocessor), some memory (RAM, ROM, flash ROM, and battery-backed RAM), a real-time clock/calendar, a hardware random-number generator, and special hardware that detects and responds to tampering. The card is powered by long-life batteries, so it doesn't depend on the host system's main power supply. Power independence is crucial, because it allows the card to react to tampering, even if someone disconnects the system's power source. Internal sensors can detect many kinds of attempts to penetrate the card or analyze its operation. It can detect changes in temperature as well as attempts to probe the card electrically or with radiation (such as x-rays). It even notices if the host system's power supply is being suspiciously manipulated. If the card detects an intrusion, it can instantly erase all the sensitive information it stores.

IBM's challenge was to reduce the features of the 4758 card to a chip-level technology suitable for integration into any ASIC or SoC. In theory, the chip-level version of SecureBlue can protect data inside a cellphone, PDA, laptop computer, digital camera, flash-memory drive, automobile, antitank missile, or anything else containing a highly integrated microprocessor.

IBM says the chip-level version of SecureBlue can duplicate virtually all the features of the 4758 card and even provide additional features. Of course, some of the more exotic security features don't make sense in everyday applications, so they are optional. A processor that automatically deletes sensitive information when x-rayed wouldn't be desirable in a laptop computer or any other consumer product that must pass through airport x-ray machines. Such a processor would also be easy prey for malicious hackers who are satisfied to see valuable data deleted, instead of stolen—merely tampering with the device would accomplish their goal. However, protection against things like x-ray exploration might be useful for a military device prone to loss on a battlefield.

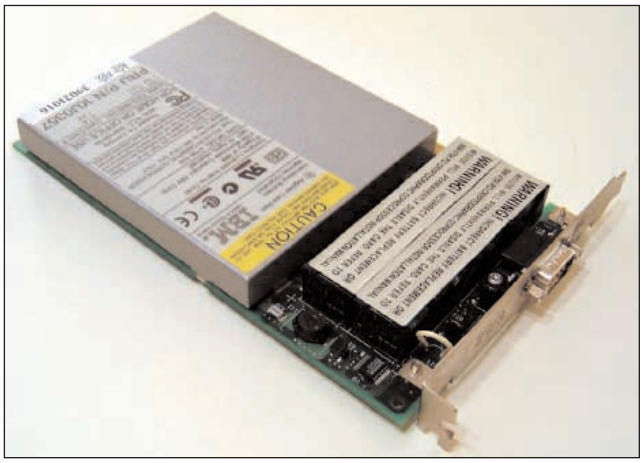

Figure 1. IBM's new SecureBlue technology adapts the security features of this IBM 4758 PCI card into licensable IP for SoCs. The PCI card is sealed inside a tamper-resistant metal enclosure. (Photo: IBM)

The most sophisticated intrusion detection requires analog sensors on chip—which, of course, necessitates a more costly mixed-signal design. For example, a temperature sensor could detect someone's trying to freeze the chip. IBM says frozen SRAM can sometimes retain data for hours after the power supply is disconnected, so a malicious hacker might try to disable and analyze a chip in this manner. But a temperature sensor could trigger a SecureBlue safety mechanism that flushes the SRAM when the temperature drops below a certain threshold. Additional sensors can detect x-rays, power interruptions, and even attempts to force a malfunction by manipulating the chip's supply voltage or clock frequency. All these SecureBlue features are optional.

Secure Memory Guards Data

One of the most important differences between SecureBlue and the 4758 card is their capacity for secure data storage. The 4758 can protect only as much data as will fit in memory enclosed inside the card, whereas a SecureBlue-enabled chip can securely store a virtually unlimited amount of data in external memory.

Figure 2 shows a block diagram of the SecureBlue architecture. One important component is a hard-wired cryptography engine, which runs independently of the operating system and offloads encryption and decryption from the SoC's main processor core. The engine varies in size, according to the type of cryptography and level of performance it supports. For example, an Advanced Encryption Standard (AES) engine would be larger than a Data Encryption Standard (DES) engine, and a faster AES engine would be larger than a slower AES engine. For now, the only off-the-shelf option is an AES engine, because IBM doesn't consider the 30-year-old DES cipher to be secure enough for SecureBlue. (For a price, IBM will adapt the engine to any block cipher, including proprietary and top-secret ones.)

In addition to the cryptography engine, SecureBlue has safe storage for master keys and other critical data (described below). The type of storage is an implementation option. It could be ROM, SRAM, flash memory, or fuses in the logic itself. In any case, once the chip stores the master keys, they are beyond the reach of external software. Even the system's owner or user can't retrieve them without triggering SecureBlue's tamper-detection mechanisms.

Another type of critical data that a SecureBlue-enabled chip can store is the root of an "integrity tree" that further guards the encrypted information. This feature is optional. Although encryption protects the confidentiality of information, it doesn't prevent tampering with the

encrypted data. Someone could try to crack the encryption by manipulating the data in various ways. To detect those attempts, SecureBlue offers the option of integrity protection, which tags the encrypted data with special metadata stored in a hierarchical tree structure.

SecureBlue keeps the root of this integrity tree in secure memory on chip, but the tree can spill over into external memory if necessary. Although the external metadata is "in the clear" (unencrypted), it contains no clues for cracking the encryption. Instead, it contains information about the formatting of the encrypted data. When the processor reads the data, SecureBlue will notice if the data was altered in any fashion, or even if it was relocated in memory. How SecureBlue responds to tampering depends on the implementation. It could throw a security exception for the operating system to handle, or even delete all the data and shut down the system.

Note that there's nothing special about the hardware of protected external memory. If the external memory is DRAM, it consists of ordinary DRAM chips. Anyone can tamper with the memory, read its data, or write new data. But the master keys for decrypting the data are safely stored inside the SecureBlue-enabled chip. And when the chip accesses the manipulated memory, SecureBlue will check the integrity tree, notice that something has changed, and react accordingly.

IBM Says Memory Overhead Is Minimal

Although the integrity tree can authenticate information stored in on-chip memory—even in the processor's data cache—the ability to protect information stored in external memory is what makes SecureBlue practical as a chip-level technology. Without this capability, SecureBlue would be limited to protecting information only within its own domain (the chip), just as the 4758 card can protect information only within its domain (the sealed PCI card). The integrity tree allows SecureBlue to protect a virtually unlimited amount of information. The same mechanism can stop untrusted software from running.

One question is how much overhead the metadata in the integrity tree adds to the actual data. IBM says the overhead varies: protecting the integrity of larger amounts of data is more efficient than protecting smaller amounts of data. Some of the metadata is for rigorous error correction, not just security, because SecureBlue would interpret ordinary bit errors as evidence of tampering. In general, IBM says the overhead is about 14.2%. That is, protecting 1MB of encrypted data would require about 145KB of metadata in the integrity tree. But remember, integrity protection is optional, and SecureBlue needn't store the entire tree on chip.

Another consideration is the effect of all this integrity checking on memory performance. SecureBlue must reference the metadata to verify all information the processor reads from secure memory. Conversely, SecureBlue must generate metadata for all information the processor writes to secure memory. It would seem that these additional steps would slow down read/write operations by at least the amount of overhead (14.2%).

However, IBM says the effect on memory latency and throughput is minimal. IBM has invented an algorithm for block ciphers that operates on data in a parallel fashion instead of a serial fashion, which significantly improves performance. The patented algorithm speeds up encryption and decryption as well as integrity checking. (See the "Pricing & Availability" box for a reference to a white paper about this algorithm, which is called integrity-aware parallel mode.) There are so many other factors involved—the design of the chip, the likelihood of cache misses, the size of the integrity tree, how much of the tree fits in on-chip memory, and so on—that IBM says it's impractical to state a useful estimate of memory performance.

Although data traffic between the SoC and external memory flows through an unsecure datapath—the chip's I/O bus—IBM says a hacker can't reverse-engineer the security algorithms by analyzing the traffic or the encrypted data in memory. These algorithms don't generate predictable output. If someone writes the same data thousands of times to the same memory location, the data will be different each time. If someone writes the same data thousands of times to different memory locations, it will also be different. In addition, the algorithms whiten the data (add noise) to obscure it. Without any patterns to discern, there are few clues to help crack the encryption.

Some Questions Remain Unanswered

IBM designed the synthesizable-logic components of SecureBlue to work with CoreConnect, an on-chip bus that IBM introduced in 1999. (See *MPR 7/12/99-03*, "PowerPC 405GP

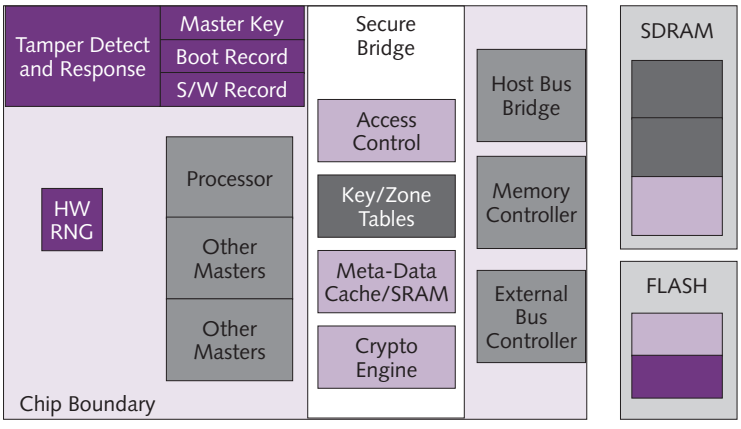

Figure 2. This high-level diagram shows IBM's SecureBlue technology in an SoC. The "secure bridge" is the boundary between encrypted data in external memory and unencrypted data inside the chip. The main SecureBlue components are a cryptographic engine, safe storage for cryptographic keys, and the root of an integrity tree (special metadata that verifies information retrieved from secure memory). Optional SecureBlue components include a hardware random-number generator and analog sensors for detecting attacks.

Price & Availability

SecureBlue technology is available for licensing now. The synthesizable-logic components are in Verilog format, but IBM hasn't released a detailed list. Licensing fees and royalties are undisclosed. A small amount of information about SecureBlue is available on IBM's website at *http://domino.watson.ibm.com/comm/pr.nsf/pages/news.20060410_security.html*.

An IBM white paper about the parallel algorithm that SecureBlue uses on block ciphers is available from the National Institute of Standards and Technology (NIST) Computer Security Resource Center. The paper, "Integrity Aware Parallelizable Mode," by IBM researcher Charanjit S. Jutla, is posted online at *http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/*.

Has CoreConnect Bus.") Although CoreConnect is openly licensed intellectual property that's especially popular in PowerPC-based designs, it's not as widespread as AMBA, which is predominant in ARM-based chips. IBM says it's willing to adapt SecureBlue to AMBA or any other on-chip bus—if the customer provides sufficient motivation in the form of cash. But if IBM is serious about making SecureBlue ubiquitous, porting the cores to AMBA should be a high priority.

IBM's claim that SecureBlue's integrity checking has a minimal effect on memory performance would be stronger if supported with benchmark tests. IBM says a lead customer is already making a chip with SecureBlue technology, so it should be possible to run some comparison tests with and without integrity checking on various amounts of data. Of course, testing one chip would provide only one data point—SecureBlue's performance depends greatly on the chip implementation—but it would be better than nothing.

For chip developers, three more vital questions remain. How much will SecureBlue enlarge the die, and therefore the chip's manufacturing cost? What will be the effect on power consumption? And how much money will it cost to license SecureBlue?

Speaking to the first question, IBM estimates that a SecureBlue implementation aiming for minimum performance would occupy about 2.5% of the die in a current process technology, which *MPR* assumes is 90nm CMOS. If the SecureBlue implementation includes intrusion-detection sensors, which are on-chip analog components, then the design will require a mixed-signal process, which would inflate the cost.

Without knowing more about the die-area overhead, it's fruitless to estimate the effect on power consumption. Gate counts for the synthesizable components would be useful, but IBM hasn't released that information. And IBM isn't providing much guidance on licensing costs or design fees, either. For developers, these are make-or-break details.

SecureBlue is an intriguing concept that addresses a genuine need in the marketplace. By now, everyone should recognize the value of universal security. But to make SecureBlue universally successful, IBM needs to release more technical details, benchmarks, and case studies. Developers need enough information to make a first-order decision without engaging IBM to obtain deeper disclosures.