Improved data confidentiality: an overview of NGSCB

Mingfeng BAO
Department of Computer Science, University of Auckland
mbao002@ec.auckland.ac.nz

Abstract

The next-generation secure computing base (NGSCB) is the Microsoft® implementation of the TCPA/TCG specification. It is a combination of specialized hardware and software features aim to provide a trusted computing environment.
This term paper goes through the hardware modification and new features, illustrates how the nexus kernel cooperates with the specialized hardware platform to ensure curtained execution and isolation. After analysing the threat models, we get a conclusion that data confidentiality can be guaranteed in an NGSCB-enabled system.

1 Introduction

Nowadays, organizations expose to the Internet and provide on-line services to employees, customers, and business partners; personal users connect to the Internet for various purposes as well. Computers are provided with many valuable data, from privacy, financial information to commercial documents, which are very attractive to some people who are snooping on the network. They keep trying all kinds of tools to intrude these machines and get benefit from stealing data.

On the other side, most current operating systems are designed for providing feature-rich environment because users like to pay features rather than security. When operation systems are driven to be more functional and optimized for better performance, their kernels tend to be even larger, in which more security-related bugs will exist. While people are enjoying the great ease and convenience come with those flexible, extensible and feature-rich operating systems, they have to remember that general-purpose OS is vulnerable; its kernel is too large to ensure integrity. [1]

Even when a general-purpose system is perfectly protected and guaranteed that no malicious code is in execution, it is still not trustworthy. Thinking about we are sharing data with others. Can we trust the user who is downloading some copyrighted

materials from our machines, can we assume that the user will never archive or redistribute those copyrighted stuff without owner's permission?

2 NGSCB Architecture

Microsoft® Next-Generation Secure Computing Base (NGSCB) is a solution for trustworthy computing, which can keep stored data secret from unknown programs and make a user's machine environment (hardware and software stack) trusted by remote parties. NGSCB is a combination of new hardware and operating system features [2] that can be used in building "A trusted open platform [1]".

An NGSCB system requires support from hardware platform; the CPU and chipset need to be modified to enable NGSCB features, and the keyboard, mouse, video graphic card and graphics adaptor need function supports for encrypted data transfer [2,3,5]. A new hardware component is tightly attached to the motherboard, which is called Security Support Component (SSC) [2] or Trusted Platform Module (TPM) [3]. In addition, the motherboard should be adapted for the above changes [2,3,5].

Each SSC stored at least one privative key in its non-volatile memory, which never leaves the chip, accompanying with a public key certificate signed by its manufacturer [2]. It is responsible for storing keys and providing encryption, decryption and hashing services to authenticated nexuses. The NGSCB-enabled CPU has an extra mode flag, which distinguishes whether the CPU is running in standard or nexus mode. High-assurance software programs are supposed to run in nexus mode. An NGSCB-enabled chipset could recognize which part of memory is used in nexus mode, and protect them from bus mastering, including direct memory access (DMA). In order to build a trusted path in NGSCB system, some secure input devices are required; they are used to transfer encrypted keystrokes and mouse movement to the NGSCB-trusted drivers or applications, at the same time, video data from the nexus to the graphics adaptor should also be protected [2].

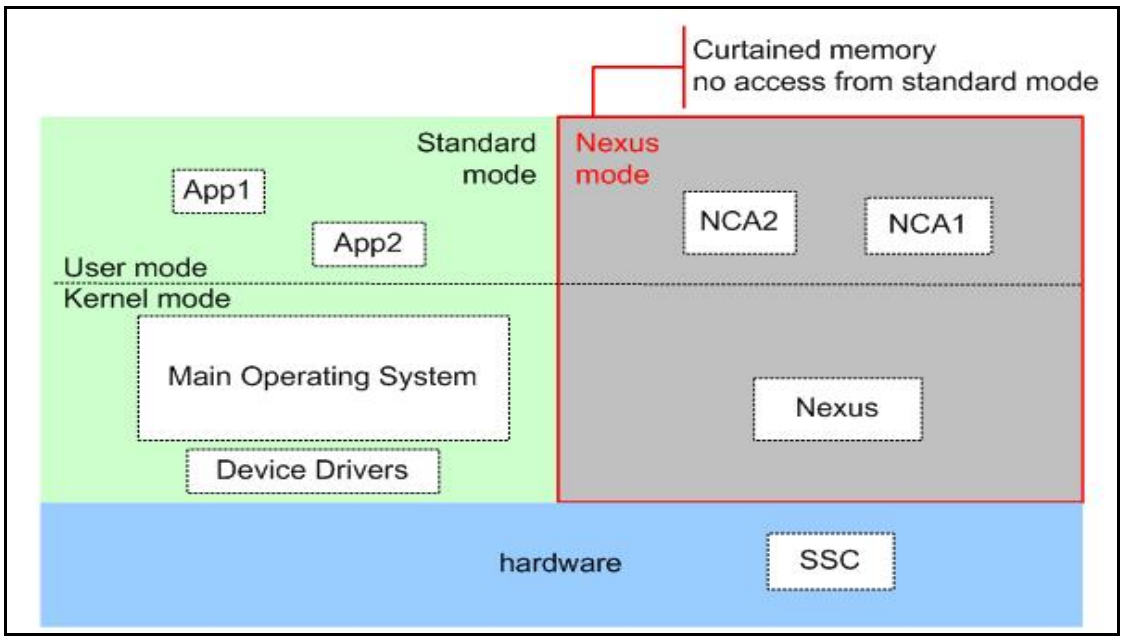

Figure 1: Curtained execution in NGSCB-enabled platform

Nexus and NCAs (Nexus Computing Agent) are software components running in protected environment in an NGSCB system (shown in Figure 1). A nexus is a small operating system kernel that is required to authenticate itself to the SSC when it starts up. When it is authenticated, it is allowed to access secrets (keys and functions) stored within the SSC. Then a special area of memory will be isolated for the nexus, which is called curtained memory because it is invisible to the main operating system or processes running in the user space. The nexus executes in the kernel mode within the curtained memory, it manages memory access, threading and secure IO devices and communicates with other non-SGSCB services. The nexus identifies and authenticates NCAs, which are trusted applications or services executing in user mode, and provides encryption and decryption service to them using the AES key stored in SSC. The nexus handles interrupts within the curtained memory as well [4].

3 Authenticated Start up for NGSCB & NCA

Although lots of hardware modification has been made in an NGSCB-enabled platform, the SSC is shipped with its features turned-off by default. Under this default setting, the system performs like a current general purpose OS. It is the machine owner's right to turn on/off the SSC through a secure UI [3,5]; NGSCB allows one nexus to run on its platform [5], it is also the owner's right to determine which nexus (identified by its hash) can execute [3,5]. When a nexus is started for the first time, the SSC randomly generates an Advanced Encryption Standard (AES) key, and stores the

key in the SSC associated with the nexus's hash. The AES key never goes out of the SSC, where it is used in serving nexus having the same hash digest as it is bound. [3]

An NGSCB system boots in standard mode, which is similar to current Windows® OS environment. In order to switch to nexus mode, an atomic nexus initialization process executes as follow:

- The standard OS prepares the nexus image in main memory and calls a CPU instruction to start nexus [2].
- The hardware send the nexus image to SSC, the SSC looks up the owner approved nexuses' hash, if it matches, its cryptographic hash is stored into a Process Control Register (PCR) [3], otherwise, it fails in initialization.
- The instruction changes hardware state and loads nexus into memory, denies all memory access (including DMA) to this part of curtained memory [2].
- Pass control to nexus, switch to nexus mode.

After the nexus is authenticated, it starts running in the curtained memory. In turn, it identifies and authenticates NCAs. The nexus has a secure UI available to its owner to grant execution permission to trusted NCAs [3]. NCA is identified by its code identity, which is the hash of the binary code [1] or a manifest of hash and public keys [3]. An NCA couldn't be loaded into and run within the curtained memory unless it has been permitted for execution. No program can modify the nexus or NCA that is running in curtained memory. Each NCA is well isolated; it has access only to memory behind the curtain allocated to it by the nexus [3,5].

4 Data protection

The authenticated start-up guarantees that only the original trusted code (both the nexus and NCA) can execute in the nexus mode. In this section, we will address how secrets are protected from malicious access.

NCAs are safe from hostile environment because they are running in the well-protected curtained memory, however their associated data stored on hard disks is visible to all programs running in standard mode. The nexus cryptographically protects its data before passing it to the file system. Sealed storage is a mechanism uses encryption to protect secrets from revealing by unauthorized programs.

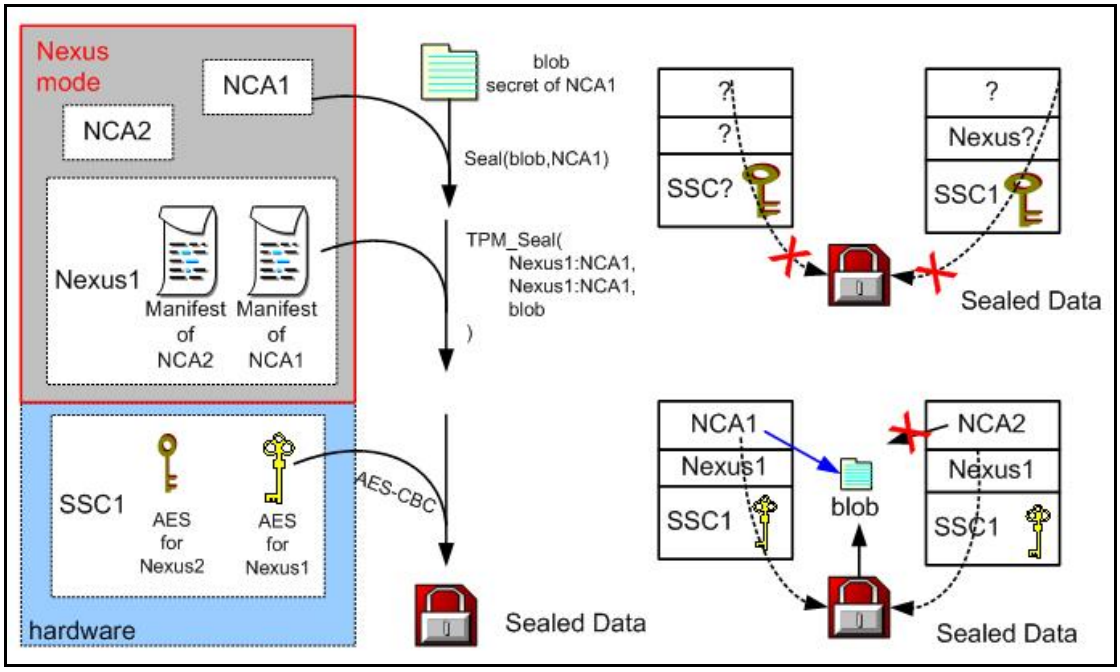

Figure 2: Seal and unseal primitives

The sealed data is tightly bound to the NCA, nexus and SSC, which means the sealed secret can only be read by the specified NCA program running on a certain machine with certain nexus kernel (illustrated in Figure 2). When an NCA wants to write some data to the disk after specifying which NCA can read it, the nexus calls the TPM_Seal service provided by the SSC. Then the SSC picks the AES key associated with the nexus and starts the AES-CBC algorithm to encrypt the data. When an NCA retrieves a sealed data, the main OS is required to read the encrypted file into the main memory and then the nexus sends an unseal request to the SSC after blocking that memory area. Only the original SSC and Nexus that sealed the data can get the proper AES key to access the original secret. The SSC decrypts the data if the integrity check succeeds, and the nexus compares the requester's identity with the stored one, only the authorized NCA can get the decrypted secret back [3].

With the support of SSC, which always hides the AES keys in its internal chips, the NGSCB system achieves a high confidentiality and integrity.

- No program running in standard mode can reveal the encrypted secret without the decryption key.
- Another SSC cannot decrypt the secret because the AES key is unique
- A tampered file will be detected by the SSC through an integrity check

- Another nexus on the same machine cannot read the secret because the AES key is nexus related
- An unauthorized NCA running on the same nexus won't get the secret because its code identity is different

In an extreme case, a malicious user could remove the SSC from the motherboard and get access to the AES key; however, this type of hardware attack only works on individual machine [2].

Sealed storage provides a strong protection over nexus's data; at the same time, the sealed data is bound to the specific software and hardware platform. The NGSCB system provides some mechanism for managing software upgrading and data migration.

- A newer version of nexus comes with a certificate that allows the older nexus to seal its secrets to the newer one [3].
- For upgrading NCAs, access rights can be transferred by reusing the original application manifest or signing a new manifest using the same key that signed the original one [3].
- In order to migrate sealed data from one machine to another, the user can ask the nexus on the source machine, through a secure UI, to encrypt its data using the public key of the destination nexus. When the destination nexus gets the data, it reseals the incoming secrets by calling TPM_Seal service provided by the SSC on the same machine [3].

5 Attestation

Comparing with a high quality program, people's behaviour is less trustworthy; Even a very honest person might make some inappropriate decisions occasionally. NGSCB is designed to authenticate hardware and software rather than user; it identifies software running on a particular remote machine while no assumption is made about that who is using the software [5]. Attestation is a mechanism that allows a NCA to authenticate itself to another NCA, local or remotely [1,3,5]. After the NCA has been authenticated, the NCA can run procedures to authenticate the actual user by obtaining the password or other attributes from the user through some secure input/output devices.

In a networked NGSCB environment, an NGSCB-enabled server can be well protected from malicious access made by unknown software. The server will only share data with an approved NCA running on recognizable SSC with a valid nexus kernel. When an NCA wants to talk to a remote NCA, the following operations are required:

- The user should explicitly allow the NCA to run attestation function, which will reveal platform information (public key) to remote parties [1,3].
- The NCA generates a private/public key pair, and passes the public key to the nexus during making an attest () request; the private key is kept secure using sealed storage [3].
- The nexus gets the NCA's identity and forwards the request to the SSC, the SSC uses the platform private key, which is a 2048-bit RSA key hidden within the SSC, to create a cryptographic identity of the NCA. The cryptographic identity contains the identity of the nexus and NCA, and the NCA generated public key as well [3].
- The nexus makes a call to the SSC and retrieves the public key certificate signed by the hardware manufacturer. Then sends the cryptographic identity with the certificate to the trusted remote party.
- The NGSCB-enabled server decrypts the identity and gets information about the client's hardware and software stack. If they are legitimate, the server will use the received public key for negotiating a session key [3].

Attestation provides a strong security foundation for the network environment, "it is a variant of public-key encryption [1]" and it is more reliable than the generic public-key encryption:

- An NGSCB platform can only have one signed certificate from its manufacturer, while a malicious person could obtain multiple certificates from CA(s) using different identity, in some cases, he / she could register a public key using another person's name.
- The NGSCB's private key never goes out of the SSC, its public key can be accessed only by authenticated nexus and authorized NCA, the NCA's private key is protected using sealed storage as well. On the other side, the current OS

doesn't provide strong protection on key management; users' private keys could be obtained and misused.

6 Threat Models

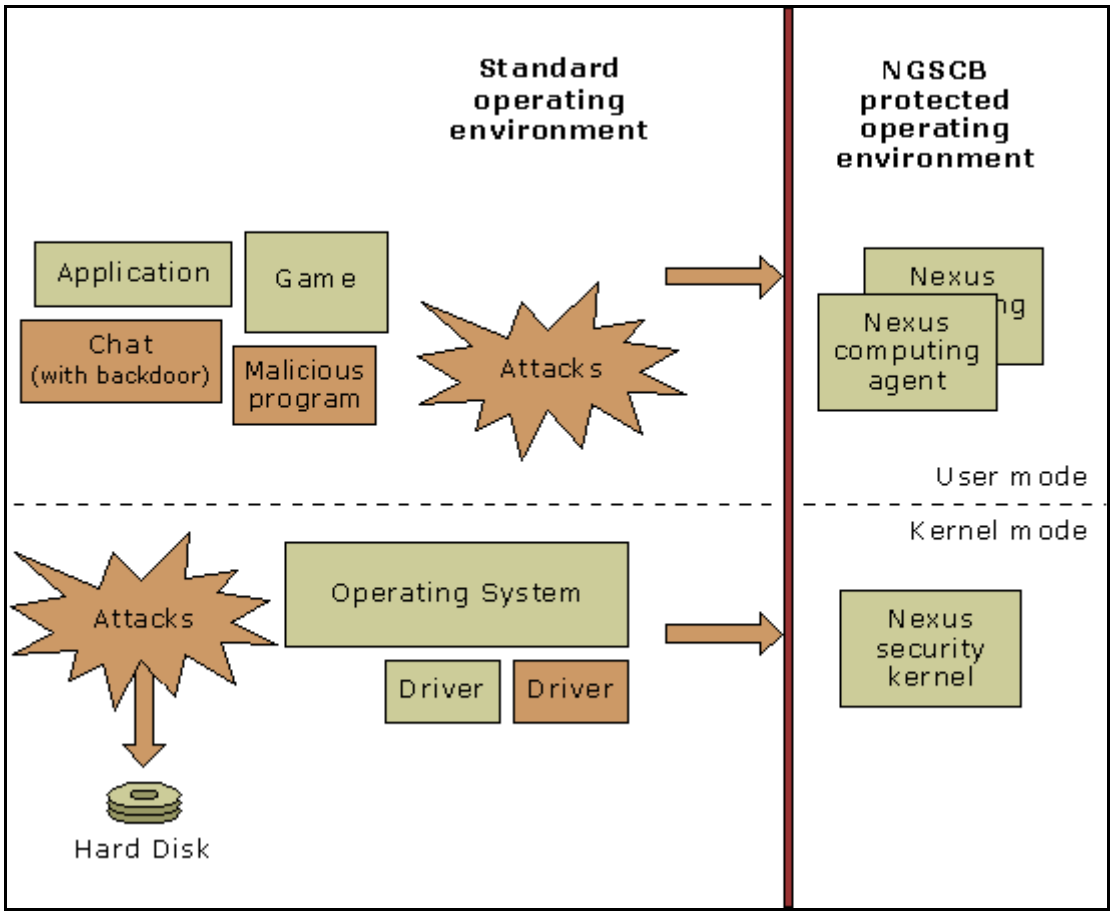

Figure 3: Typical NGSCB Configuration [4]

Figures 3 shows a typical NGSCB system where some malicious programs are running on the standard mode, in both the user and kernel space. All these attacks could harm the system but they cannot cross the boundary between the standard mode and the nexus mode.

It is possible that there is some bad code running in the nexus mode as well, in the case of the owner authorizes an inappropriate program to execute by mistake or being cheated by a third party, which is similar to downloading spy ware in current environment. Having been well isolated by the nexus, the bad NCA still cannot access other programs' memory space, including the nexus, other NCA and applications on the standard side.

However, in the presence of bugs or intentional backdoors within the nexus kernel, just like other operating systems, the NGSCB could be compromised. If it does have bugs, we should expect that a malicious NCA could exploit them to get privileged access without modifying the nexus. As the result, security boundary no longer exists and the bad NCA could reveal local sealed secrets to a remote hostile server. We cannot expect less skilled users always do the right thing in authorizing NCA and attestation; it's the nexus vendors' responsibility to offer bug free nexus kernel.

Since a nexus kernel is much smaller than a current Windows® kernel and only a few services are implemented, its quality could be guaranteed at a high level. However, the NGSCB architecture does not provide a full protection over the platform:

- Malicious code can execute as usual, which consumes CPU time.
- "Neither code nor data on the nexus side are paged out in the first version of NGSCB [3] ", so a bad NCA could exhaust the system memory.
- A malicious program cannot reveal sealed secrets, but it can delete or corrupt the encrypted file. The corresponding NCA will lose its data.
- The nexus can be blocked from accessing I/O [4].
- A malicious program can shut down the computer directly.
- An NGSCB system can be attacked from the hardware level.

7 Conclusion

Combining all security features offered by the curtained execution, sealed storage, attestation and secure IO, the protected environment is very reliable. Each NCA receives, processes, outputs and/or stores its own secrets separately, without any interference with other programs running on the same machine. The isolation is guaranteed and supported by the SSC. The NGSCB-enabled platform can also be deployed for digital rights management (DRM). For example, a data owner can specified that its data could only be read by an NCA named as p, the owner trusts p because he/she knows it will follow the owner's policy. By using the attestation mechanism, the owner only delivers data to machines with the software stack in the form of nexus: p. As the result, the usage of copyrighted stuff is under the owner's control; even it is on the remote machine.

Although the NGSCB does not guarantee the availability of service, its data confidentiality is improved to a very high level; we are promised that only authorized programs can get access to the confident valuable data.

References:

1 P. England et al., "A Trusted Open Platform", IEEE Computer 36:7, July 2003, pp.55-62

2 Microsoft White Paper, "Hardware Platform for the Next-Generation Secure Computing Base", Aug 11th, 2004,available at http://www.microsoft.com/resources/ngscb/NGSCB_Overview.mspx

3 Microsoft White Paper, "NGSCB: Trusted Computing Base and Software Authentication", Aug 11th, 2004,available at http://www.microsoft.com/resources/ngscb/NGSCB_Overview.mspx

4 Microsoft White Paper, "Security Model for the Next-Generation Secure Computing Base", Aug 11th, 2004,available at http://www.microsoft.com/resources/ngscb/NGSCB_Overview.mspx

5 Microsoft White Paper, (Published: November 2003),"Privacy-Enabling Enhancements in the Next-Generation Secure Computing Base", Aug 11th, 2004,available at http://www.microsoft.com/resources/ngscb/NGSCB_Overview.mspx