BIOS Update for PC Client Devices Protection Profile

Information Assurance Directorate

12 February 2013

Version 1.0

Table of Contents

List of Tables

Revision History

Version	Date	Description
1.0	4 May 2012	Initial release

1 Introduction to the PP

1.1 PP Overview of the TOE

1 This is the first generation of a Standard Protection Profile (PP) for PC client devices Basic Input/Output System (BIOS) firmware. The BIOS or the system BIOS is the primary firmware used to facilitate the hardware initialization process and transition control to the operating system on PC client devices. This PP addresses the primary threat that an adversary will modify or replace the BIOS on a PC client device and compromise the PC client environment in a persistent way.

2 The Target of Evaluation (TOE) defined in this PP is a PC client device. PC client devices are modern computers such as desktop and laptop computers. This PP focuses on current and future x86 and x64 client platforms, and is independent of any particular system design. Systems that should claim compliance to this PP include x86 and x86-64 desktop and laptop systems. Examples of devices that are not intended for evaluation against this PP, although the PP may be applicable since such devices may implement equivalent authenticated BIOS update protections, include mobile devices (such as tablets and smart phones) and servers.

3 There are several different types of BIOS firmware. Some computers use a 16-bit conventional BIOS, while many newer systems use boot firmware based on the Unified Extensible Firmware Interface (UEFI) specifications [12]. The BIOS is typically developed by both original equipment manufacturers (OEMs) and independent BIOS vendors, and it is distributed to end-users by motherboard (stored in non-volatile memory) or computer manufacturers. Manufacturers frequently update system firmware to fix bugs, patch vulnerabilities, and support new hardware. The system BIOS is stored on electrically erasable programmable read-only memory (EEPROM) or other forms of flash memory, and is modifiable by end users. System BIOS firmware is updated using a utility or tool that has special knowledge of the non-volatile storage components in which the BIOS is stored. The system BIOS is the first piece of software executed on the main central processing unit (CPU) when a computer is powered on. Its primary role on modern machines is to initialize and test hardware components and load the operating system. In addition, the BIOS loads and initializes important system management functions, such as power and thermal management. The system BIOS may also load CPU microcode patches during the boot process.

1.1.1 Usage and Major Security Features of the Target of Evaluation (TOE)

4 This PP provides Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) for preventing the unauthorized modification of BIOS firmware on PC client systems and is based on the recommendations provided in the NIST SP 800-147, *BIOS Protection Guidelines* [4]. The term BIOS in this PP includes the definition described in section 1.2 (page 1-1) of the NIST SP 800-147 [4]:

> *"As used in this publication, the term BIOS refers to conventional BIOS, Extensible Firmware Interface (EFI) BIOS, and Unified Extensible Firmware Interface (UEFI) BIOS. This document applies to system BIOS firmware (e.g., conventional BIOS or UEFI BIOS) stored in the system flash*

memory of computer systems, including portions that may be formatted as Option ROMs. However, it does not apply to Option ROMs, UEFI drivers, and firmware stored elsewhere in a computer system."

5 Although the TOE defined in this PP is the PC client device, the security functional requirements defined in this PP apply to the BIOS firmware stored in the system flash. This includes Option ROMs and UEFI drivers that are stored with the system BIOS firmware and are updated by the same mechanism. It does not apply to Option ROMs, UEFI drivers, and firmware stored elsewhere in a computer system.

6 The system BIOS is a critical component of a secure system. As the first code executed during the boot process, the system BIOS is implicitly trusted by hardware and software components in a system. Thus, PC client devices that conform to this PP must maintain the integrity of the BIOS after it has been provisioned by securing mechanisms used for updating the BIOS. The TOE contains an authenticated BIOS update mechanism. The authenticated BIOS update mechanism employs digital signatures to ensure the authenticity of the BIOS update image. To update the BIOS using the authenticated BIOS update mechanism, there shall be a Root of Trust for Update (RTU) that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image. The key store and the signature verification algorithm shall be stored in a protected fashion on the computer system and shall be modifiable only using an authenticated update mechanism or a secure local update mechanismIn the context of this PP, the RTU is an abstraction that contains the key store and cryptographic algorithm and is part of the TSF.

7 The key store and the signature verification algorithm shall be stored in a protected fashion on the computer system and shall be modifiable only using an authenticated update mechanism or a secure local update mechanism. In the context of this PP, the RTU is part of the TSF.

8 The update mechanism, which is also a part of the TSF, shall ensure that the BIOS update image has been digitally signed and that the digital signature can be verified using a key in the RTU before updating the BIOS. If recovery mechanisms are provided, the TOE shall also use the update mechanism unless the recovery process meets the requirements for a secure local update.

9 Another aspect of the TSF is the capability to control "writes" or modifications to the BIOS and entities that are stored in the same flash memory. This ensures that the only way in which to modify the BIOS requires the trusted update mechanism to be used.

10 BIOS implementations may optionally include a secure local update mechanism that updates the system BIOS without using the authenticated update mechanism. The secure local update mechanism, if it is implemented, should only be used to replace the manufacturer's original BIOS image or to recover from a corruption of a system BIOS that cannot be fixed using the authenticated update mechanism. A secure local update mechanism shall ensure the authenticity and integrity of the BIOS update image by requiring physical presence. Further protections may be implemented in the secure local update mechanism by requiring the entry of an administrator password or the unlocking of a physical lock (e.g., a motherboard jumper) before permitting the system BIOS to be updated.

1.1.2 Administration

11 The vendor is required to provide installation and configuration guidance (AGD_PRE, AGD_OPR) to correctly install and update the BIOS and administer the TSF (e.g., load new keys in the key store, configure the key sizes, algorithms or update the cryptographic algorithms) for every operational environment supported (for example, for every O/S supported by the product).

1.1.3 Available non-TOE Hardware/Software/Firmware

12 The TOE defined in this PP is a PC client device. All the devices connected to the PC client that are used to provide the BIOS image for updating (network-based system management tools, organization-maintained update server, manufacturer update server, etc.) are part of the IT operational environment. The TOE relies on the operational environment for providing an interface(s) for updating the system BIOS and the RTU. The vendor is expected to provide sufficient installation and configuration instructions to identify an operational environment with the necessary features and to provide instructions for how to configure it in a correct and secure manner.

13 In some cases the TOE vendor will have to provide specific configuration guidance for the operational environment to enable the TOE to meet its security objectives. Such guidance is considered operational guidance for the TOE and should be provided as part of the documentation available for end users of the TOE.

2 Security Problem Definition

14 The primary asset being protected is the system BIOS of a PC client device. Unauthorized modification of BIOS firmware by malicious software constitutes a significant threat because of the BIOS's unique and privileged position within the PC architecture. A malicious BIOS modification could be part of a sophisticated, targeted attack on an organization—either a permanent denial of service (if the BIOS is corrupted) or a persistent malware presence (if the BIOS is implanted with malware).

2.1 Threats

15 A threat consists of a threat agent, an asset, and an adverse action of that threat agent on that asset.

16 The primary threat agents are the entities that put the assets at risk if that threat agent can remotely modify or replace installed BIOS firmware. For instance, in the T.UNAUTHORIZED_BIOS_UPDATE in the table below, an attacker (the threat agent) attempts to remotely insert an unauthorized BIOS update (adverse action) to compromise the security features of the user's PC client device (the asset). The threat agent's objectives may include data compromise, user impersonation or denial of service.

17 For this PP, the TOE is not expected to defend against all threats related to a malicious BIOS update that may compromise the security features of the PC client device. For instance, the TOE is not responsible for detecting the integrity of the system BIOS while the device moves through the supply chain. Assuming that the PC client device arrives with the manufacturer's intended system BIOS installed, there are a still number of threats to the integrity of the system BIOS during the system's lifetime that are not covered by security functions included in this PP:

- Network-based system management tools could be used to launch an organization-wide attack on system BIOSs. For example, consider an organization-maintained update server for the organization's deployed system BIOS; a compromised server could push a malicious system BIOS to computer systems across the organization. If the BIOS update is signed by the update server this attack would succeed.

- Malicious installation of BIOS images can be achieved with physical access. The security functions included in this PP will not prevent users from installing unapproved BIOS images if they have physical access to the computer system and could replace the flash for example. Appendix C of this PP defines optional SFRs that partially remedy this threat by requiring a recovery process to restore to an approved BIOS when the updating process fails to completed a successful update of the BIOS.

- Any of the preceding update mechanisms that could be used to rollback to an authentic but vulnerable system BIOS. This is a particularly insidious attack, since the "bad" BIOS is authentic (i.e., shipped by the manufacturer).

Table 1: Threats

Threat	Description of Threat
T.UNAUTHORIZED_BIOS_UPDATE	An attacker attempts to replace the BIOS in the PC client with a malicious BIOS update that may compromise the security features of the TOE.
T.UNAUTHORIZED_BIOS_MODIFY	An attacker attempts to modify the BIOS in the PC client that may compromise the security features of the TOE.

2.2 Assumptions

18 This section of the security problem definition shows the assumptions that are made on the operational environment in order to be able to provide security functionality. If the TOE is placed in an operational environment that does not meet these assumptions, the TOE may not be able to provide all of its security functionality. Assumptions can be for physical, personnel and connectivity aspects of the operational environment.

Table 2: TOE Assumptions

Assumption	Description of Assumption
A.MANUFACTURER_BIOS	PC client system is delivered with the manufacturer's intended system BIOS installed.
A.AUTHORIZED_ADMINISTRATORS	Authorized administrators and local users of the TOE are trusted to follow and apply all administrator provided guidance.

3 Security Objectives

19 The Security Objectives are the requirements for the TOE and for the Operational Environment derived from the threats and the assumptions in Section 2. Section 4 restates the security objectives for the TOE more formally as SFRs. The TOE is evaluated against the SFRs.

3.1 Security Objectives for the TOE

20 Table 3 identifies the security objectives of the TOE. These security objectives reflect the stated intent to counter identified threats. The TOE has to meet these objectives by satisfying the SFRs. The objectives specified are primarily focused on verifying the source and integrity of the system BIOS update.

Table 3: Security Objectives for the TOE

Objective	Objective Description
O.BIOS_AUTHENTICATED_UPDATE	The TOE must provide a mechanism to ensure that any BIOS update to the TOE is verified to be trusted.
O.ROOT_OF_TRUST_FOR_UPDATE	The TOE must have a RTU that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image.
O.BIOS_INTEGRITY_PROTECTION	The TOE must implement mechanisms to prevent unintended or malicious modification of the system BIOS and RTU.
O.BIOS_NON-BYPASSABILITY	The TOE must ensure that the system BIOS is modified only by an authenticated BIOS update mechanism.

3.2 Security Objectives for the Operational Environment

21 The Operational Environment of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). The security objectives for the Operational Environment consist of a set of statements describing the goals that the Operational Environment should achieve.

22 This section defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. The assumptions identified in Section 2.2 are incorporated as security objectives for the environment. They levy additional requirements on the environment, which are

largely satisfied through procedural or administrative measures. Table 4 identifies the security objectives for the Operational Environment.

Table 4: Security Objectives for the Operational Environment

Objective	Objective Description
OE.MANUFACTURER_BIOS	Manufacturer must deliver the PC client system with the intended system BIOS installed.
OE.TRAINED_ADMINISTRATORS	Authorized administrators and local users of the TOE must be properly trained and follow all security guidance.

3.3 Security Objectives Rationale

23 This section describes the rationale for the Security Objectives as defined in the previous section. Table 5 illustrates the mapping from Security Objectives to the Threats.

Table 5: Security Objectives to Threats Mappings

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.UNAUTHORIZED_BIOS_UPDATE An attacker attempts to replace the BIOS in the PC client with a malicious BIOS update that may compromise the security features of the TOE.	O.BIOS_AUTHENTICATED_UPDATE The TOE must provide a mechanism to ensure that any BIOS update to the TOE is verified to be trusted. O.ROOT_OF_TRUST_FOR_UPDATE The TOE must have a RTU that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image. O.BIOS_NON-BYPASSABILITY The TOE must ensure that the system BIOS is updated only by an authenticated BIOS mechanism.	O.BIOS_AUTHENTICATED_UPDATE mitigates this threat by ensuring that the TOE will provide a secure BIOS update mechanism that verifies the authenticity and integrity of BIOS and RTU updates. O.ROOT_OF_TRUST_FOR_UPDATE ensures that the TOE will implement a RTU that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image. O.BIOS_NON-BYPASSABILITY counters this threat by ensuring that the system BIOS is updated only by an authenticated BIOS mechanism and thus does not allow a malicious BIOS update because it would not be authenticated.
T.UNAUTHORIZED_BIOS_MODIFY An attacker attempts to modify BIOS in the PC client that may compromise the security features of the TOE.	O.BIOS_INTEGRITY_PROTECTION The TOE must implement mechanisms to prevent unintended or malicious modification of the system BIOS and RTU.	O.BIOS_INTEGRITY_PROTECTION counters this threat by ensuring that the TOE will control the modification of the system BIOS and RTU such that unintended or malicious modification of the system BIOS and RTU is

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
	O.BIOS_NON-BYPASSABILITY The TOE must ensure that the system BIOS is updated only by an authenticated BIOS mechanism.	prevented. O.BIOS_NON-BYPASSABILITY ensures that the system BIOS is updated only by an authenticated BIOS mechanism such that other means to modify the BIOS would fail.

24 Table 6 illustrates the mapping from Security Objectives to Assumptions.

Table 6: Security Objectives to Assumptions Mappings

Assumption	Objectives Addressing the Assumption	Rationale
A.MANUFACTURER_BIOS The PC client system is delivered with the manufacturer's intended system BIOS installed.	OE.MANUFACTURER_BIOS The manufacturer will deliver the PC client system with the intended system BIOS installed in it.	OE.MANUFACTURER_BIOS ensures that the manufacturer will deliver the PC client system with the intended system BIOS installed in it.
A.AUTHORIZED_ADMINISTRATORS Authorized administrators and local users of the TOE are trusted to follow and apply all administrator provided guidance.	OE.TRAINED_ADMINISTRATORS Authorized administrators and local users of the TOE will be properly trained and follow all security guidance.	OE.TRAINED_ADMINISTRATORS ensures that the administrators and local users of the TOE will be properly trained on how to administer and use the TOE in a secure way for system BIOS updates.

4 Security Requirements and Rationale

25 The Security Requirements are divided into functional requirements and assurance requirements. The SFRs are a formal instantiation of the Security Objectives and are provided with application notes in Section 4.1. Section 4.2 is the required tracing of the SFRs to the Security Objectives.

26 The SARs are typically inserted into a PP and listed separately from the SFRs; the Common Criteria Evaluation Methodology (CEM) is then consulted during the evaluation based on the SARs chosen. Because of the nature of the Common Criteria SARs and the specific technology identified as the TOE, a more tailored approach is taken in this PP. While the SARs are still listed for context and completeness in Section 4.3, the majority of the activities that an evaluator needs to perform for this TOE with respect to each SFR and SAR are detailed in "Assurance Activities" paragraphs. Assurance Activities are normative descriptions of activities that must take place in order for the evaluation to be complete. Assurance Activities are located in two places in this PP; those that are associated with specific SFRs are located in Section 4.1, while those that are independent of the SFRs are detailed in Section 4.3. Note that the Assurance Activities are in fact a tailored evaluation methodology, presented in-line for readability, comprehension, and convenience.

27 For the activities associated directly with SFRs, after each SFR one or more Assurance Activities is listed detailing the activities that need to be performed to achieve the assurance required for conformant devices.

28 For the SARs that require activities that are independent of the SFRs, Section 4.3 indicates the additional Assurance Activities that need to be accomplished, along with pointers to the SFRs for which specific Assurance Activities associated with the SAR have been written.

29 Future iterations of the Protection Profile may provide more detailed Assurance Activities based on lessons learned from actual product evaluations.

4.1 Security Functional Requirements

30 The SFRs are a translation of the security objectives for the TOE. They are usually at a more detailed level of abstraction, but they have to be a complete translation (the security objectives must be completely addressed). The CC requires this translation into a standardized language for several reasons:

- To provide an exact description of what is to be evaluated. As security objectives for the TOE are usually formulated in natural language, translation into a standardized language enforces a more exact description of the functionality of the TOE.

- To allow comparison between two Security Targets (ST)s. As different ST authors may use different terminology in describing their security objectives, the standardized language enforces using the same terminology and concepts. This allows easy comparison.

4.1.1 Class: Cryptographic Support (FCS)

31 The primary threats addressed by these functional requirements are a brute force attack against the key space and the failure of a cryptographic component.

32 The cryptographic requirements make reference to standards describing the algorithms; most of these standards are available from US NIST as Special Publications (800-xxx) or Federal Information Processing Standards (FIPS). The assurance requirements detail how the implementation of these requirements is to be verified. Each scheme has the option of specifying the process under which the cryptographic assurance activities may be considered satisfied. All cryptographic functionality specified below must be implemented within the TOE.

Cryptographic operation (FCS_COP)

FCS_COP.1(1) **Cryptographic operation (Signature Verification)**

FCS_COP.1.1(1) **Refinement:** The TSF shall perform **cryptographic signature verification for the BIOS update image** in accordance with a **[selection:**

1) **Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater,**
2) **RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater, or**
3) **Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]**

that meets the following:

Case: Digital Signature Algorithm

- **[FIPS PUB 186-3, "Digital Signature Standard"]**

Case: RSA Digital Signature Algorithm

- **[FIPS PUB 186-3, "Digital Signature Standard"]**

Case: Elliptic Curve Digital Signature Algorithm

- **[FIPS PUB 186-3, "Digital Signature Standard"]**
- **The TSF shall implement "NIST curves" [selection: P-256, P-384 and P-512, no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard").**

Application Note:

33 The ST author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm. In particular, if ECDSA is selected as one of the signature algorithms, the key size specified must match the selection for the curve used in the algorithm.

34 For elliptic curve-based schemes, the key size refers to the binary logarithm (log_2) of the order of the base point. As the preferred approach for digital signatures, elliptic curves will be required after all the necessary standards and other supporting information are fully established.

Assurance Activities:

35 This requirement is used to verify digital signatures attached to BIOS updates from the TOE manufacturer/developer before installing those BIOS updates on the TOE.

36 For any algorithm, the evaluators check the TSS to ensure that it describes the overall flow of the signature verification. This should at least include identification of the format and general location of the data to be used in verifying the digital signature; how the data received from the operational environment are brought on to the BIOS; and any processing that is performed that is not part of the digital signature algorithm (for instance, hashing the public key provided with the BIOS update image, and ensuring that it matches a hash which appears in the key store, etc).

37

38 It should be noted that there are no key generation/domain parameter generation testing requirements. This is because it is not anticipated that this functionality would be needed in the end device, since the functionality is limited to checking digital signatures in delivered BIOS updates.

39 Tests used to verify that the TOE accepts correctly signed BIOS images and rejects incorrectly signed BIOS images are described in FPT_BUA_EXT.1 and can be considered as the minimum/basic tests for this requirement.

rDSA

There are two options for implementing the signature generation/verification function: ANSI X9.31 and PKCS #1 (Version 1.5 and/or Version PSS). At least one of these options must be implemented. Each implemented version must be tested as indicated below. If PKCS #1 Version PSS is chosen, then the evaluator shall check the TSS to ensure the length of the salt is specified.

If the TOE supports more than one modulus size, then the evaluator shall perform the following test for all modulus sizes. If the TOE supports more than one hash algorithm, the evaluator shall perform the following test for all hash algorithms. This means that if the implementation allows the choice of 2 modulus sizes and 2 hash algorithms, the evaluator would perform the following test 4 times.

The evaluator shall generate three groups of data. Each group of data consists of a modulus and 4 sets of test vectors consist with the modulus. The test vectors consist of a public exponent *e*; a pseudorandomly-generated message; and a signature for the message using the associated private key (consistent with *e* and the modulus *n*). This means that there will be a minimum of 12 test vectors for each modulus size/hash algorithm supported by the TSF.

In 3/4s of the test vectors after the correct signature has been generated (but not "fed" to the TSF), the evaluators will alter the public key, message, or signature (making sure to do at least 2 of each) so that the signature verification failure function will be tested. The evaluators shall then run the test vectors through the TSF and verify that the results are correct.

40 In addition, if the algorithm implemented is RSASSA-PKCS1-v1_5, as specified in *Public Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Standard-2002*, or the RSA algorithm described in X9.31, *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*, the appropriate additional test vectors from http://csrc.nist.gov/groups/STM/cavp/documents/dss/SigVer15EMTest.zip (for PKCS #1 Version 1.5 implementations) or http://csrc.nist.gov/groups/STM/cavp/documents/dss/SigVer931IRTest.zip (for X9.31 implementations) shall be used by the evaluators to verify that the implementation successfully passes these tests.

DSA

The evaluator examines the TSS to ensure that the values used for (L, N) are given, and the hash algorithm(s) used are specified. The evaluator verifies that the hash algorithm used for a specific (L,N) provides the requisite strength, as specified in Tables 2 and 3, Section 5.6.1 of SP 800-57, *Recommendation for Key Management --Part I: General (Revised)*. The evaluators shall also ensure that the (L,N) selected has comparable strength to the symmetric (data) encryption algorithm used on the USB Flash Drive (e.g., if 128-bit AES is used to encrypt the user data, then an (L,N) of at least (3072, 256) is required).

The evaluator performs the following test for each (L,N) and hash combination supported. The evaluator shall generate a key pair. The evaluators will then pseudorandomly generate 15 1024-bit messages and sign them with the private key. For about half of the messages--after the correct signature has been generated (but not "fed" to the TSF)--the evaluators will alter the public key, message, or signature (making sure to do at least 2 of each) so that the signature verification failure function will be tested. The evaluators shall then run the test vectors through the TSF and verify that the results are correct.

41

ECDSA

The evaluators shall examine the TSS to determine the curve or curves used in the implementation are specified and consistent with the requirement and the hash or hashes supported are specified. The evaluators shall conduct the following test for each curve, hash pair implemented by the TSF.

42 The evaluator generates 15 sets of data. Each dataset consists of a pseudorandom message; a public/private key pair (d,Q), and a signature (r,s). For about half of the messages--after the correct signature has been generated (but not "fed" to the TSF)--the evaluators will alter the public key, message, or signature (making sure to do at least 2 of each) so that the signature verification failure function will be tested. The evaluators shall then run the data through the TSF and verify that the results are correct.

FCS_COP.1(2)	**Cryptographic operation (Cryptographic Hashing)**
FCS_COP.1.1(2)	**Refinement:** The TSF shall perform **cryptographic hashing services** in accordance with **[selection: SHA-1, SHA 224, SHA 256, SHA 384, SHA 512]** and **message digest sizes [selection: 160, 224, 256, 384, and 512] bits** that meet the following**: FIPS Pub 180-3, "Secure Hash Standard".**

Application Note:

43 The intent of this requirement is to specify the hashing function used for digital signature algorithms specified in FCS_COP.1(1). The hash selection must support the message digest size selection. The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1(1).

44 Also if the TOE stores a hash value in the RTU, the ST author may iterate this requirement to specify the hashing function used to confirm the integrity of the key provided as part of a BIOS update. The TOE will hash the public key provided and compare the value with the hash value stored in the RTU.

Assurance Activities:

45 The evaluator checks the Guidance Documents to determine that any information required configuring the functionality for the required hash sizes is present. The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

46 The cryptographic hashing tests reference *The Secure Hash Algorithm Validation System (SHAVS)* [13], available from http://csrc.nist.gov/groups/STM/cavp/documents/shs/SHAVS.pdf.

The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-

oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented tests.

The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

- ***Short Messages Test - Bit-oriented Mode***

The evaluators devise an input set consisting of *m*+1 messages, where *m* is the block length of the hash algorithm. The length of the messages range sequentially from 0 to *m* bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- ***Short Messages Test - Byte-oriented Mode***

The evaluators devise an input set consisting of *m/8*+1 messages, where *m* is the block length of the hash algorithm. The length of the messages range sequentially from 0 to *m*/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- ***Selected Long Messages Test - Bit-oriented Mode***

The evaluators devise an input set consisting of *m* messages, where *m* is the block length of the hash algorithm. The length of the *i*th message is $512 + 99*i$, where $1 \leq i \leq m$. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- ***Selected Long Messages Test - Byte-oriented Mode***

The evaluators devise an input set consisting of *m/8* messages, where *m* is the block length of the hash algorithm. The length of the *i*th message is $512 + 8*99*i$, where $1 \leq i \leq m/8$. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- ***Pseudorandomly Generated Messages Test***

This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is *n* bits long, where *n* is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

47 In addition to the algorithm tests described above, tests used to verify that the TOE accepts correctly signed BIOS images and rejects incorrectly signed BIOS images are described in FPT_BUA_EXT.1 and can be considered as the minimum/basic tests for this requirement.

4.1.2 Class: Protection of the TSF (FPT)

48 The following functional requirements are related to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. It provides requirements that the TSF data cannot be tampered with and the TSF mechanisms cannot be bypassed.

Extended: BIOS Update Mechanisms (FPT_BUM_EXT)

FPT_BUM_EXT.1	**Extended: BIOS Update Mechanisms**
FPT_BUM_EXT.1.1	The TSF shall provide an authenticated BIOS update mechanism as described in FPT_BUA_EXT.1, and [selection, choose one of: a secure local update mechanism described in FPT_SLU_EXT.1, no other mechanism,] to update the BIOS system.

Application Note:

49 The secure BIOS update mechanism is used for verifying the authenticity and integrity of BIOS updates; and for ensuring that it is protected from modification outside of the secure update process. The authenticated BIOS update mechanism shall be protected from unintended or malicious modification by a mechanism that is at least as strong as that protecting the RTU and the system BIOS.

50 The intent of this requirement is to ensure that an authenticated BIOS update mechanism and (an optional) secure local update mechanism will be provided. Authentication verifies that a BIOS was generated by an authentic source and is unaltered. All updates to the system BIOS shall go through an authenticated BIOS update mechanism as described in FPT_BUA_EXT.1. Additionally if the TOE offers a secure local update mechanism, the appropriate selection is made and the ST author must select the FPT_SLU_EXT.1 specified in Appendix C.

Assurance Activities:

51 The evaluator shall consult the TSS of the ST in performing the assurance activities for this requirement. The evaluator focuses on ensuring that the description is comprehensive in how the BIOS update image is updated to the system and the point at which the authenticated BIOS update mechanism is applied.

52 In performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on process of updating the BIOS. When the RTU is not integrated in the BIOS, its update process shall also be explained in the TSS specifying the security mechanisms guaranteeing its integrity. The evaluators also shall check other evidences (AGD, ADV and ATE) for detailed descriptions

of authenticated BIOS update mechanism. If an optional secure local update mechanism is supported by the TOE, the evaluator shall check that the TSS and other evidences describe how this mechanism works and when this mechanism can be used.

Extended: BIOS Update Authentication (FPT_BUA_EXT)

FPT_BUA_EXT.1	**Extended: BIOS Update Authentication**
FPT_BUA_EXT.1.1	The TSF shall authenticate the source of the BIOS update using the digital signature algorithm specified in FCS_COP.1(1) using the key store that contains [selection: the public key, hash value of the public key].
FPT_BUA_EXT.1.2	The TSF shall only allow installation of updates if the digital signature has been successfully verified as specified in FCS_COP.1(1).

Application Note:

53 The authenticated BIOS update mechanism employs digital signatures to ensure the authenticity of the BIOS update image. The TSF provides a RTU that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image. The key store in the RTU shall include *a public key* used to verify the signature on a BIOS update image or *a hash of the public key* if a copy of the public key is provided with the BIOS update image. In the latter case, the update mechanism shall hash the public key provided with the BIOS update image, and ensure that it matches a hash which appears in the key store before using the provided public key to verify the signature on the BIOS update image. If the hash of the public key is selected, the ST author may iterate the FCS_COP.1(2) requirement - to specify the hashing functions used.

54 The intent of this requirement is to specify that the authenticated BIOS update mechanism shall ensure that the BIOS update image has been digitally signed; and that the digital signature can be verified by using a public key before updating the BIOS. The requirement also specifies that the authenticated BIOS update mechanism allows installation of only BIOS updates that digital signature has been successfully verified by the TSF.

Assurance Activities:

55 The evaluator shall consult the TSS of the ST in performing the assurance activities for this requirement. The evaluator focuses on ensuring that the description is comprehensive in how the RTU is implemented by the TSF (i.e. the RTU key storage includes the public key or the hash of the public key).

56 The TSS should cover the initialization process and the activities that are performed to ensure that the digital signature of the BIOS/RTU update image is verified before updating the BIOS/RTU. In performing

their review, the evaluators shall determine that the TSS contains a description of the digital signature verification process. The evaluators also shall look in other evidence (AGD, ADV and ATE) for descriptions of what will happen if the digital signature of the BIOS/RTU image cannot be verified successfully by the TSF (i.e. the BIOS/RTU image has not been signed, or signature cannot be verified using the key saved in the RTU, etc.).

57 The evaluators shall perform the following tests:

- Test 1: The evaluator determines the current version of the BIOS present in the TOE (system). After the update tests described in the following tests, the evaluator performs this activity again to verify that the version correctly corresponds to that of the updated BIOS.
- Test 2: The evaluator obtains or produces a legitimate update image of the BIOS using procedures described in the operational guidance and verifies that BIOS update is successfully loaded on the TOE. Perform a subset of other assurance activity tests to demonstrate that the update functions as expected.
- Test 3: The evaluator obtains or produces an illegitimate BIOS update image(s) for each of the following cases:
    a) unsigned image,
    b) signed with an incorrect key,
    c) signed with the correct key, but a corrupted BIOS image,

and attempts to install it on the TOE (system). The evaluator verifies that the TOE rejects all the attempted BIOS updates.

58 If the TSF can be updated separately from the system BIOS, the ST author should include the requirement FPT_TUD_EXT.1 from Appendix C.

Extended: Protection of the BIOS (FPT_PBR_EXT)

FPT_PBR_EXT.1	**Extended: Protection of the BIOS**
FPT_PBR_EXT.1.1	The TSF shall only allow modification of the BIOS by the update mechanisms described in FPT_BUM_EXT.1.

Application Note:

59 The BIOS (excluding configuration data used by the BIOS that is stored in nonvolatile memory) and software portions of TSF (e.g., RTU (key store and the signature verification algorithm)) shall be stored in a write protected area on the TOE. The BIOS shall be modifiable only using the authenticated update mechanism described in FPT_BUA _EXT.1 or the secure local update mechanism described in FPT_SLU

_EXT.1. The ST author would make the proper selection in FPT_BUM_EXT.1 to specify the exact mechanisms used. The TSF is modifiable only by using the mechanisms specified in FPT_TUD_EXT.1 (Appendix C).

Assurance Activities:

60 The evaluators shall check the TSS section to determine that it specifies that a key store and the signature verification algorithm are stored in write protected area. If the TSF can be updated, the ST Author will include the FPT_TUD_EXT.1 requirement and the associated assurance activates for TSF updates will be contained there. This will include updates to the cryptographic algorithms, as well as modifications to the key store.

61 Evaluators shall check the operational guidance to determine that there are instructions for how to securely modify the key store and signature verification algorithm within the RTU. The evaluators shall check in other evidences (ADV and ATE) for descriptions of the processes of updating (replacing) the key store and signature verification algorithm and explanation of what happens when an update is unsuccessful.

62 The evaluator shall perform the following tests:

- Test 1: Using the information contained in the TSS and in other evidences (AGD, interface specifications), the evaluator shall attempt to overwrite or modify the BIOS in the system while bypassing the authenticated update (e.g., using a modified Linux boot loader such as GRUB that attempts to write to the flash where the BIOS is stored).
- Test 2: Using the information contained in the TSS and in other evidences (AGD, interface specifications), the evaluator shall attempt to overwrite or modify the TSF components (key store, cryptographic algorithms) in the system.

4.2 Rationale for Security Functional Requirements

63 This section describes the rationale for the TOE SFRs as defined in Section 4.1. Table 7 illustrates the mapping from SFRs to Security Objectives with a corresponding rationale that the objective is addressed by the requirement. This table should be augmented by the ST author/vendor when they complete the selections and assignments for the requirements in Section 4.1, as well as (potentially) augment the baseline requirements with requirements from Appendix C.

Table 7: Rationale for TOE Security Functional Requirements

Objective	Requirement Addressing the Objective	Rationale
O.BIOS_AUTHENTICATED_UPDATE The TOE must provide a mechanism to ensure that any BIOS update to the	FPT_BUM_EXT.1	FPT_BUM_EXT.1 requires that an authenticated BIOS update mechanism and (an optional) secure local update

Objective	Requirement Addressing the Objective	Rationale
TOE is verified to be trusted.		mechanism will be provided. Authentication verifies that a BIOS update image is generated by an authentic source and is unaltered.
O.ROOT_OF_TRUST_FOR_UPDATE The TOE must have an RTU that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image.	FPT_BUA_EXT.1 FCS_COP.1(1) FCS_COP.1(2)	FPT_BUA_EXT.1 specifies that TSF will provide a RTU that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image. This requirement specifies that the authenticated BIOS update mechanism will ensure that the BIOS update image has been digitally signed; and that the digital signature can be verified by using a public key before updating the BIOS. The requirement also specifies that authenticated BIOS update mechanism will allow installation of only BIOS updates when the digital signature has been successfully verified by the TSF. FCS_COP.1(1) specifies the digital signature verification algorithms that are used by the TOE. FCS_COP.1(2) specifies the hashing function used for digital signature algorithms specified in FCS_COP.1(1).
O.BIOS_INTEGRITY_PROTECTION The TOE must implement mechanisms to prevent unintended or malicious modification of the system BIOS and RTU.	FPT_PBR_EXT.1	FPT_PBR_EXT.1 requires that the BIOS and RTU can be modified only by an authenticated update mechanism and/or secure local update mechanism. This requirement also specifies that the TSF shall protect the BIOS and RTU from unauthorized modification.

Objective	Requirement Addressing the Objective	Rationale
O.BIOS_NON-BYPASSABILITY The TOE must ensure that the system BIOS is updated only by an authenticated BIOS mechanism.	FPT_PBR_EXT.1	FPT_PBR_EXT.1 requires that the BIOS can be modifiable only by using the authenticated update mechanism or the secure local update mechanism.

4.3 Security Assurance Requirements

64 The Security Objectives for the TOE in Section 3.1 were constructed to address threats identified in Section 2.1. The Security Functional Requirements (SFRs) in Section 4.1 are a formal instantiation of the Security Objectives.

65 As indicated in the introduction to Section 4.1, while this section contains the complete set of SARs from the CC, the Assurance Activities to be performed by an evaluator are detailed both in Section 4.1 as well as in this section.

66 For each family, "Developer Notes" are provided on the developer action elements to clarify what, if any, additional documentation/activity needs to be provided by the developer. For the content/presentation and evaluator activity elements, additional assurance activities (to those already contained in Section 4.1) are described as a whole for the family, rather than for each element. Additionally, the assurance activities described in this section are complementary to those specified in Section 4.1.

67 The TOE security assurance requirements, summarized in Table 8, identify the management and evaluative activities required to address the threats identified in Section 2.1 of this PP. Section 4.4 provides a succinct justification for choosing the security assurance requirements in this section.

Table 8: TOE Security Assurance Requirements

Assurance Class	Assurance Components	Assurance Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
Tests	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability analysis
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage

4.3.1 Class ADV: Development

68 For TOEs conforming to this PP, the information about the TOE is contained in the guidance documentation available to the end user as well as the TOE Summary Specification (TSS) portion of the ST. The Assurance Activities contained in Section 4.1 should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

4.3.1.1 ADV_FSP.1 Basic functional specification

69 The functional specification describes the TSF Interfaces. It is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this PP will necessarily have interfaces to the Operational Environment that are not directly invokable by TOE users, there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. The activities for this family for this PP should focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional "functional specification" document should be necessary to satisfy the assurance activities specified.

70 In understanding the interfaces to the TOE, it is important to consider that the primary threat to be countered is user-initiated installation of a malicious system BIOS. The TOE update interface is of most importance. As described earlier, it is critical that any code that is replaced is properly signed and the signatures verified.

71 The interfaces that need to be evaluated are characterized through the information needed to perform the assurance activities listed, rather than as an independent, abstract list.

	Developer action elements:
ADV_FSP.1.1D	The developer shall provide a functional specification.
ADV_FSP.1.2D	The developer shall provide a tracing from the functional specification to the SFRs.
Developer Note:	As indicated in the introduction to this section, the functional specification is composed of the information contained in the AGD_OPR and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

Content and presentation elements:

ADV_FSP.1.1C	The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.2C	The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.3C	The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.
ADV_FSP.1.4C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_FSP.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Assurance Activities:

72 The assurance activities associated with these SARs will be those dedicated to checking the CC content requirements fulfillment. The functional specification documentation is provided to support the evaluation activities described in Section 4.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.

4.3.2 Class AGD: Guidance Documents

73 The guidance documents will be provided with the developer's security target. As indicated in the introduction, the duties of actual "administrators" are fairly restricted, so the guidance documents will contain information that is required by and used by all users of the TOE. To this end, "authorized user" is used in most cases in the text below; when "administrator" is used (except in the verbatim requirements from the CC) it is referring to the subset of users with responsibility for setting up the operational environment to allow a BIOS installation/update and any other TOE security functionality that requires human intervention.

74 Guidance must include a description of how the authorized user verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by an authorized user.

75 Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes:

- Instructions to successfully install or update the TOE in that environment; and
- Instructions to manage the security of the TOE as a product and as a component of the larger operational environment

76 Guidance pertaining to particular security functionality is also provided; requirements on such guidance are contained in the assurance activities specified in section 4.1.

4.3.2.1 AGD_OPE.1 Operational User Guidance

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Developer Note: The developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluators will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of

the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activities:

77 The documentation must describe the process for verifying that updates to the TOE come from the intended source (which in most cases will be the TOE vendor). This verification process may be initiated by the authorized user but performed by the TOE on the client machine. The evaluators shall verify that this process includes the following steps:

1. Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE.

2. Instructions for discerning whether the update process was successful or unsuccessful.

78 Considering that there are no explicit users/administrators related SFRs defined for this TOE, some requirements for the AGD_OPE may not be applicable (e.g., describing available functions and interfaces). Some of the AGD_OPE requirements however may become applicable if the FPT_SLU_EXT.1 defined in Appendix C is included. Thus the AGD_OPE evidence must provide more details for this SAR, as applicable.

4.3.2.2 AGD_PRE.1 Preparative procedures

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Developer Note: As with the operational guidance, the developer should look to the assurance activities to determine the required content with respect to preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

AGD_PRE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

Assurance Activities:

79 As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

4.3.3 Class ATE: Tests

80 Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through ATE_IND family, while the latter is through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

4.3.3.1 ATE_IND.1 Independent testing - Conformance

81 Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operation) documentation provided. The focus of the testing is to confirm that the requirements specified in Section 4.1 are being met, although some additional testing is specified for SARs in Section 4.3. The Assurance Activities identify the minimum testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this PP.

Developer action elements:

ATE_IND.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

ATE_IND.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2E The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

Assurance Activities:

82 The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the body of this PP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluators must document in the test plan that each applicable testing requirement in the ST is covered.

83 The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platform and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

84 The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluators are expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) is provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform.

85 The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was

a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a “fail” and “pass” result (and the supporting details), and not just the “pass” result.

4.3.4 Class AVA: Vulnerability assessment

86 For the first generation of this protection profile, the evaluation lab is expected to survey open sources to learn what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, evaluators will not be expected to test for these vulnerabilities in the TOE. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future protection profiles.

4.3.4.1 AVA_VAN.1 Vulnerability survey

	Developer action elements:
AVA_VAN.1.1D	The developer shall provide the TOE for testing.
	Content and presentation elements:
AVA_VAN.1.1C	The TOE shall be suitable for testing.
	Evaluator action elements:
AVA_VAN.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.1.2E	The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.1.3E	The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Assurance Activities:

87 As with ATE_IND the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in BIOS products in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-

applicability or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, for example, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires an electron microscope and liquid nitrogen, for instance, then a test would not be suitable and an appropriate justification would be formulated.

4.3.5 Class ALC: Life-cycle support

88 At the assurance level provided for TOEs conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation at this assurance level.

4.3.5.1 ALC_CMC.1 Labeling of the TOE

89 This component is targeted at identifying the TOE such that it can be distinguished from other products or version from the same vendor and can be easily specified when being procured by an end user.

	Developer action elements:
ALC_CMC.1.1D	The developer shall provide the TOE and a reference for the TOE.
	Content and presentation elements:
ALC_CMC.1.1C	The TOE shall be labeled with its unique reference.
	Evaluator action elements:
ALC_CMC.2.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activities:

90 The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the identifier is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

4.3.5.2 ALC_CMS.1 TOE CM coverage

91 Given the scope of the TOE and its associated evaluation evidence requirements, this component's assurance activities are covered by the assurance activities listed for ALC_CMC.1.

Developer action elements:

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

Evaluator action elements:

ALC_CMS.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activities:

92 The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.

4.4 Rationale for Security Functional Requirements

93 The rationale for choosing these security assurance requirements is that this is the first U.S. Government Protection Profile for this technology. If vulnerabilities are found in these types of products, then more stringent security assurance requirements will be mandated based on actual vendor practices.

5 Conformance Claims

94 The Conformance Claim indicates the source of the collection of requirements that is met by a PP or a ST that passes its evaluation. Application notes are provided in the SFR and SAR sections to further clarify specific requirements that must be met.

5.1 PP Conformance Claim

95 This PP is conformant to CC 3.1r4, CC Part 2 extended and CC Part 3 conformant.

96 STs that claim conformance to this PP shall meet a minimum standard of strict-PP conformance as defined in Section D3 of CC Part 1 (CCMB-2006-09-001).

97 Strict-PP conformance means the requirements in the PP are met and that the ST is an instantiation of the PP. The ST can be broader than the PP. The ST specifies that the TOE does at least the same as the PP, while the operational environment does at most the same as the PP. In this PP, application notes are provided to further clarify and explain the intent of the requirements specified and the expectation as to how the vendor will meet the requirements. It is expected that the evaluator of the ST will ensure strict-PP compliance by determining that the ST and its described TOE not only contain all the statements within this PP (and possibly more) but also meet the expectations as stated by the application notes.

5.2 PP Conformance Claim rationale

98 This PP does not claim conformance to another PP.

Appendix A: Supporting Tables and References

[1] Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program, July 2011

[2] Federal Information Processing Standards Publication (FIPS-PUB) 180-4, Secure Hash Standard, March, 2012

[3] Federal Information Processing Standard Publication (FIPS-PUB) 186-3, Digital Signature Standard (DSS), National Institute of Standards and Technology, June 2009

[4] NIST Special Publication 800-147, BIOS Protection Guidelines, April 2011

[5] NIST Special Publication 800-107, Recommendation for Applications Using Approved Hash Algorithms, February 2009

[6] NIST Special Publication 800-89, Recommendation for Obtaining Assurances for Digital Signature Applications, November 2006

[7] NIST Special Publication 800-102, Recommendation for Digital Signature Timeliness, September 2009

[8] ANS X9.31, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry, 1998

[9] RFC 3447, PKCS #1: RSA Cryptography Specifications Version 2.1, February 2003

[10] ANS X9.62, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 2005

[11] ITU-T Recommendation X.509, November 2008

[12] UEFI Specification Version 2.3. Unified EFI Forum. May 2009. http://www.uefi.org/specs/

[13] The Secure Hash Algorithm Validation System (SHAVS), by Lawrence E. Bassham III, July 2004 http://csrc.nist.gov/groups/STM/cavp/documents/shs/SHAVS.pdf.

Acronyms

ANSI	American National Standards Institute
BIOS	Basic Input/Output System

CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria
CM	Configuration management
CPU	Central Processing Unit
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
EEPROM	Electrically Erasable Programmable Read-Only Memory
EFI	Extensible Firmware Interface
FIPS	Federal Information Processing Standards
ISSE	Information System Security Engineers
IT	Information Technology
ITU	International Telecommunication Union
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
PC	Personal Computer
PP	Protection Profile
PKCS	Public-Key Cryptography Standards
PUB	Publication
RTU	Root of Trust for Update
RSA	R. Rivest, A. Shamir and L. Adleman.
SAR	Security Assurance Requirement
SF	Security Function
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm

SHS	Secure Hash Standard
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification
UEFI	Unified Extensible Firmware Interface

Appendix B: NIST SP 800-53/CNSS 1253 Mapping

99 Several of the NIST SP 800-53/CNSS 1253 controls are either fully or partially addressed by compliant TOEs. This section outlines the requirements that are addressed, and can be used by certification personnel to determine what, if any, additional testing is required when the TOE is incorporated into its operational configuration.

100 Application Note: In this version, only a simple mapping is provided. In future versions, additional narrative will be included that will provide further information for the certification team. This additional information will include details regarding the SFR to control mapping discussing what degree of compliance is provided by the TOE (e.g., fully satisfies the control, partially satisfies the control). In addition, a comprehensive review of the specified assurance activities, and those evaluation activities that occur as part of satisfying the SARs will be summarized to provide the certification team information regarding how compliance was determined (e.g., document review, vendor assertion, degree of testing/verification). This information will indicate to the certification team what, if any, additional activities they need to perform to determine the degree of compliance to specified controls.

101 Since the ST will make choices as far as selections, and will be filling in assignments, a final story cannot necessarily be made until the ST is complete and evaluated. Therefore, this information should be included in the ST in addition to the PP. Additionally, there may be some necessary interpretation (e.g. "modification") to the activities performed by the evaluator based on a specific implementation. The scheme could have the oversight personnel (e.g., Validators) fill in this type of information, or could have this done by the evaluator as part of the assurance activities. The verification activities are a critical piece of information that must be provided so the certification team can determine what, if anything, they need to do in addition to the work of the evaluation team.

Identifier	**Name**	**Applicable SFR**
AC-3	Access Enforcement	FPT_BUA_EXT.1.1, FPT_RTU_EXT.1.1
SC-3	Security Function Isolation	FPT_BUA_EXT.1.1, FPT_RTU_EXT.1.1
SC-13	Use of Cryptography	FPT_BUA_EXT.1.2, FCS_COP.1(1) FCS_COP.1(2)
SC-24	Fail In Known State	FPT_TEE.1.2, FPT_RCV.1.1

SI-7	Software and Information Integrity	FPT_BUA_EXT.1.2
SI-9	Information Input Restrictions	FPT_SLU_EXT.1 .1
SI-10	Information Input Validation	FPT_TEE.1.1
SI-11	Error Handling	FPT_RCV.1.1
SI-6	Security Function Verification	FPT_BUM_EXT.1
SI-3	Malicious Code Protection	FPT_PBR_EXT.1

Appendix C: Additional Requirements

102 For this version of the PP, this appendix contains additional components without supporting threats, objectives, rationale, or assurance activities (although some guidance is given for selected components). In tandem with the current review cycle, this supporting information will be developed and incorporated into the next release of the PP. Comments on the information contained in this section (both on whether the requirements contained are applicable to the potential conformant TOEs as well as requirements that are not contained in this appendix that are widely applicable BIOS products) are welcomed and solicited.

103 As indicated in the introduction to this PP, there are several capabilities that a TOE may implement and still be conformant to this PP. These capabilities are not required, creating a dependency on the Operational Environment (for instance, identification and authentication of administrators of the TOE). However, if a TOE does implement such capabilities, the ST author will take the following information and include it in their ST. Requirements not contained in this appendix may be included in the ST, but are subject to review and acceptance by the National Scheme overseeing the evaluation before a conformance claim to this PP can be made.

C.1 Protection of the TSF (FPT)

C.1.1 Manual recovery (FPT_RCV)

FPT_RCV.1	**Manual recovery**
FPT_RCV.1.1	**Refinement: If the BIOS fails to update or fails to successfully boot,** the TSF shall ~~enter~~ **provide** a maintenance mode where the ability to return to a secure state is provided **by using secure local update mechanism described in FPT_SLU_EXT.1.**

Application Note:

104 Providing a recovery mechanism for BIOS updates is optional. However, if the TOE provides recovery mechanisms then this requirement would be included by the ST.

105 The intent of this requirement is to enable the TSF to recover from a corrupted or malfunctioning system BIOS that can not be corrected using the authenticated BIOS update mechanism. The TSF will provide mechanisms to allow a user with physical presence during the boot process to replace the current system BIOS with a known good version and configuration. The administrator guidance would include instructions for the administrator to follow a recovery process. The ST author would include the FMT_SMF.1 to describe a security management function to provide the administrator the ability to manually recover the BIOS through this process.

106 Note that this requirement has a dependency on FPT_SLU_EXT.1, Secure Local Update.

Assurance Activities:

107 The evaluator shall examine the TSS section to confirm that it describes how the BIOS recovery mechanism works to ensure only the proper version of the BIOS is installed. The evaluator reviews the AGD guidance and shall determine that the instructions for using the secure local update process will allow manual recovery. The interface description in the guidance document is checked to ensure it is consistent with the details of how the mechanism works (e.g., there are no options or parameters of an interface that provide a function or feature not captured in the TSS). The evaluator verifies the process by following the instructions provided in the AGD guidance.

108 The evaluators shall perform the following test(s):

- Test 1: To test this requirement, the evaluator may stop the BIOS updating process by forcing shutdown of the PC system before the BIOS update is completed (for example by unplugging the PC), to trigger the recovery process.
- Test 2: The evaluator will use the secure local update mechanisms to update the BIOS with the good known version of the BIOS.

109 If this component is included, the following threat(s), objective(s) and rationale should be added to the ST.

T.UNAUTHORIZED_BIOS_RECOVERY	An attacker may attempt to initiate an authorized recovery process that will install a version of system BIOS (with potential vulnerabilities).

O.BIOS_MANUAL_RECOVERY	The TOE shall be able to return to a secure state if the BIOS update fails or fails to successfully boot by using secure local update mechanism.

T.UNAUTHORIZED_BIOS_RECOVERY An attacker may attempt to initiate an authorized recovery process that will install a version of system BIOS (with potential vulnerabilities).	O.BIOS_MANUAL_RECOVERY The TOE shall be able to return to a secure state if the BIOS update fails or fails to successfully boot by using secure local update mechanism.	O.BIOS_MANUAL_RECOVERY mitigates this threat by ensuring that the TOE will provide secure local update mechanism. This mechanism guards against attackers flashing old firmware with potential vulnerabilities.

O.BIOS_MANUAL_RECOVERY The TOE shall be able to return to a secure state if the BIOS update fails or fails to successfully boot by using secure local update mechanism.	FPT_RCV.1	FPT_RCV.1 specifies that the TOE shall provide the capability to recover if the BIOS update fails or fails to successfully boot.

C.1.2 Extended: Secure Local Update (FPT_SLU _EXT)

FPT_SLU_EXT.1	**Extended: Secure Local Update**
FPT_SLU_EXT.1.1	The TSF shall provide a secure local update mechanism [assignment: description of update mechanism(s)] that requires physical access to the TOE, by an authorized user, before permitting the system BIOS to be updated.
FPT_SLU_EXT.1.2	The secure local update mechanism shall be used only to [selection: replace the manufacturer's original BIOS image, recover from a corruption of a system BIOS that cannot be fixed using the authenticated update mechanism described in FPT_BUA_EXT.1].

Application Note:

110 This requirement identifies BIOS implementations that include a secure local update mechanism that updates the system BIOS without using the authenticated update mechanism. The secure local update mechanism shall ensure the authenticity and integrity of the BIOS update image by requiring a physical presence to the TOE. Examples of secure local updates mechanism include requiring entry of an administrator password or unlocking of a physical lock (e.g., a motherboard jumper) before permitting the BIOS to be updated. The use of the local update mechanism is limited by the selection in FPT_SLU_EXT.1.2.

111 The administrator guidance would include instructions for the administrator to use the local update mechanism interface. The ST author would add the FMT_SMF.1 SFR to include a management function to provide the administrator the ability to initiate the local update mechanism, as appropriate.

Assurance Activities:

112 The evaluator shall check the TSS section to confirm that it clearly and thoroughly describes how the secure local update functionality is implemented. The evaluator reviews the AGD guidance and shall

determine that the instructions for using the secure local update are clear and completely describe the actions required to complete a BIOS update including how to verify the update was successful. The evaluator tests the secure local update by following the instructions provided in the AGD guidance.

113 The evaluators shall perform the following test(s):

- Test 1: The secure local update will be used only in two occasions: (1) load the first BIOS image, and/or (2) recover from a corruption of a system BIOS that cannot be fixed using the authenticated update mechanism. The evaluator may be able to create or obtain a BIOS image and use this mechanism after the authenticated update mechanism was not able to install a BIOS update.

114 If this component is included, the following objective(s) and rationale must be added to the ST.

O.BIOS_SECURE_LOCAL_UPDATE	The TOE will implement a mechanism that installs a BIOS update image by requiring physical presence that can ensure the integrity of the update.

T.UNAUTHORIZED_BIOS_UPDATE An attacker attempts to replace the BIOS in the PC client with a malicious BIOS update that may compromise the security features of the TOE.	O.BIOS_SECURE_LOCAL_UPDATE The TOE will implement a mechanism that installs a BIOS update image by requiring physical presence that can ensure the integrity of the update.	O.BIOS_SECURE_LOCAL_UPDATE mitigates this threat by requiring physical presence to the TOE for ensuring the integrity of the BIOS update as an alternative to an authenticated BIOS update.

O.BIOS_SECURE_LOCAL_UPDATE The TOE will implement a mechanism that installs a BIOS update image by requiring physical presence that can ensure the integrity of the update.	FPT_SLU_EXT.1	FPT_SLU_EXT.1 requires that the TOE will provide the secure local update mechanism that requires a physical presence to the TOE in order to update the system BIOS.

C.1.3 Testing of external entities (FPT_TEE)

FPT_TEE.1 Testing of external entities

FPT_TEE.1.1 **Refinement:** The TSF shall run a suite of tests **before updating the BIOS** to check the fulfillment of **[assignment: method of verifying BIOS update version is later than the currently installed version].**

FPT_TEE.1.2 **Refinement:** If the test fails, the TSF shall **prevent the unauthorized rollback of the BIOS to an earlier authentic version and initiate a normal boot cycle using the currently installed BIOS except in the following conditions: [assignment: list conditions that rollback will be allowed].**

Application Note:

115 This requirement prevents an unauthorized rollback of the BIOS to an earlier authentic version. This eliminates the possibility that an earlier authentic BIOS version that may have a known security weakness unknowingly being installed. The selection in FPT_TEE.1.2 requires the ST author to specify conditions on which an earlier authentic version of BIOS can be installed.

116 The administrator guidance would include instructions for the administrator to configure the rollback prevention mechanism, if appropriate. The ST author would include the FMT_SMF.1 to describe a management function to provide the ability to initiate the rollback mechanism, as appropriate.

Assurance Activities:

117 The evaluator shall check the TSS section to confirm that it clearly and thoroughly describes how the TSF verifies the BIOS update version being installed is not an earlier authentic version if the conditions specified in the assignment of the FPT_TEE.1.2 are not satisfied. The evaluator shall check the TSS to verify that it describes conditions on which an early authentic version of BIOS can be installed. The evaluator tests the rollback mechanism by following the instructions provided in the AGD guidance and performing the following tests.

118 The evaluators shall perform the following test(s):

- Test 1: The evaluator attempts to install an earlier authentic BIOS version using the authenticated update mechanism when the conditions specified in the assignment of the FPT_TEE.1.2 are not satisfied. The evaluator verifies that the system will not allow the BIOS to be installed and reboots using the currently installed BIOS. During the reboot the evaluator should verify the BIOS version is the same as the last successful update.
- Test 2: The evaluator attempts to install an earlier authentic BIOS version when the conditions specified in the assignment of the FPT_TEE.1.2 are satisfied. The evaluator verifies that the system will allow the earlier BIOS version to be installed and reboots using the earlier BIOS version. During the reboot the evaluator should verify the BIOS version.

119 If this component is included, the following threat(s), objective(s) and rationale should be added to the ST.

T.UNAUTHORIZED_BIOS_ROLLBACK	An attacker may attempt to install an older version of system BIOS (with potential vulnerabilities).

O.BIOS_ROLLBACK	The TOE will implement mechanisms that prevent the unauthorized rollback of the BIOS to an earlier authentic version.

T.UNAUTHORIZED_BIOS_ROLLBACK An attacker may attempt to install an older version of system BIOS (with potential vulnerabilities).	O.BIOS_ROLLBACK The TOE will implement mechanisms that prevent the unauthorized rollback of the BIOS to an earlier authentic version.	O.BIOS_ROLLBACK mitigates this threat by ensuring that the TOE will implement mechanisms that prevent the unauthorized rollback of the BIOS to an earlier authentic version.

O.BIOS_ROLLBACK The TOE will implement mechanisms that prevent the unauthorized rollback of the BIOS to an earlier authentic version.	FPT_TEE.1	FPT_TEE.1 requires the TSF to check the version of the BIOS image and have a method of verifying BIOS update version is later than the currently installed version. It also requires the TSF to prevent the unauthorized rollback of the BIOS to an earlier authentic version and initiate a normal boot cycle using the currently installed BIOS with the exception of some other action(s) that would allow the rollback.

C.1.4 Trusted TSF Update (FPT_TUD)

Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1 Extended: TSF Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide the ability to query the current version of the TSF software.

FPT_TUD_EXT.1.2 The TSF shall provide the ability to initiate updates to TSF software.

FPT_TUD_EXT.1.3 The TSF shall only allow installation of TSF updates if the digital signature has been successfully verified as specified in FCS_COP.1(x).

Application Note: The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(x) in the body of the ST. The ST Author may use the first iteration if the algorithm employed is the same used for verifying the update to the BIOS, or they may iterate FCS_COP.1 if a different algorithm is used.

Assurance Activities: Updates to the TSF are signed by an authorized source. The definition of an authorized source is contained in the TSS, along with a description of how the keys used by the update verification mechanism are installed in the TOE. The evaluator ensures this information is contained in the TSS and that any instructions dealing with the installation of the update keys is detailed in the operational guidance. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature of the updates; and the actions that take place for successful (signature was verified) and unsuccessful (signature could not be verified) cases.

The evaluators shall perform the following tests:

- Test 1: The evaluator performs the version verification activity to determine the current version of the TSF. After the update tests described in the following tests, the evaluator performs this activity again to verify that the version correctly corresponds to that of the update.
- Test 2: The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Perform a subset of other assurance activity tests to demonstrate that the update functions as expected.
- Test 3: The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TSF rejects the update.

Appendix D: Document Conventions

120 Except for replacing United Kingdom spelling with U.S. English spelling, the notation, formatting, and conventions used in this PP are consistent with version 3.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP reader.

121 The notation, formatting, and conventions used in this PP are largely consistent with those used in version 3.1 of the CC. Selected presentation choices are discussed here to aid the PP user. The CC allows several operations to be performed on functional and assurance requirements; refinement, selection, assignment, and iteration are defined in Appendix C4 of Part 1 of the CC 3.1. Each of these operations is used in this PP.

Refinement Convention

122 The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by the word "Refinement" in **bold text** after the element number and the additional text in the requirement in bold text.

Selection Convention

123 The **selection** operation is used to select one or more options provided by the CC in stating a requirement (see appendix C.4.3 Part 1, CC 3.1). Selections that have been made by the PP authors show the selection in **bold** characters, the brackets and the word "selection" removed. Selections to be filled in by the ST author are shown in square brackets with an indication that a selection is to be made, [selection:].

Assignment Convention

124 The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a passphrase (see appendix C.4.2 Part 1, CC 3.1). Showing the value in **bold** characters denotes assignments that have been made by the PP authors, the brackets and the word "assignment" are removed. Assignments to be filled in by the ST author are shown in square brackets with an indication that an assignment is to be made [assignment:].

Iteration Convention

125 The **iteration** operation is used when a component is repeated with varying operations (see appendix C.4.1 Part 1, CC 3.1). The iteration number (iteration_number) is show in parenthesis following the component identifier.

126 The iteration operation may be performed on every component. The PP/ST author performs an iteration operation by including multiple requirements based on the same component. Each iteration of a component shall be different from all other iterations of that component, which is realized by completing assignments and selections in a different way, or by applying refinements to it in a different way.

127 For requirements stated in Appendix C, if the component has been used in the PP, the iteration number is shown as "(#)" to signify the ST author must replace the "#" with the appropriate iteration number.

Extended Requirement Convention

128 Extended requirements are permitted if the CC does not offer suitable requirements to meet the authors' needs. **Extended requirements** must be identified and are required to use the CC class/family/component model in articulating the requirements. Extended requirements will be indicated with the "EXT" inserted within the component.

Application Notes

129 Application notes contain additional supporting information that is considered relevant or useful for the construction of security targets for conformant TOEs, as well as general information for developers, evaluators, and ISSEs. Application notes also contain advice relating to the permitted operations of the component.

Assurance Activities

130 Assurance activities serve as a Common Evaluation Methodology for the functional requirements levied on the TOE to mitigate the threat. The activities include instructions for evaluators to analyze specific aspects of the TOE as documented in the TSS, thus levying implicit requirements on the ST author to include this information in the TSS section. In this version of the PP these activities are directly associated with the functional and assurance components, although future versions may move these requirements to a separate appendix or document.

Appendix E: Glossary of Terms

Basic Input/Output System (BIOS) - refers collectively to boot firmware based on the conventional BIOS, Extensible Firmware Interface (EFI), and the Unified Extensible Firmware Interface (UEFI).

Conventional BIOS - Legacy boot firmware used in many x86-compatible computer systems. Also known as the legacy BIOS.

Extensible Firmware Interface (EFI) - A specification for the interface between the operating system and the platform firmware. Version 1.10 of the EFI specifications was the final version of the EFI specifications, and subsequent revisions made by the Unified EFI Forum are part of the UEFI specifications.

Flash Memory - The non-volatile storage location of system BIOS, typically in electronically erasable programmable read-only memory (EEPROM) flash memory on the motherboard. While system flash memory is a technology-specific term, guidelines in this document referring to the system flash memory are intended to apply to any non-volatile storage medium containing the system BIOS.

Firmware - Software that is included in read-only memory (ROM).

Operational Environment – hardware and software that are outside the TOE boundary that support the TOE functionality and security policy, including the host platform, its firmware, and the operating system.

Option ROM - Firmware that is called by the system BIOS. Option ROMs include BIOS firmware on add-on cards (e.g., video card, hard drive controller, network card) as well as modules which extend the capabilities of the system BIOS.

Persistent memory – data storage that retains the data when power is turned off.

Protected Mode - An operational mode found in x86-compatible processors with hardware support for memory protection, virtual memory, and multitasking.

Root of Trust for Update (RTU) – As used in this document, it is a root of trust that contains: a. a signature validation algorithm, b. a key store that contains the public key needed to verify the signature on the BIOS update image, or c. instead of the public key, a hash of the public key if the public key is provided with the BIOS to be updated.

SAR (Security Assurance Requirements) – describes the development and evaluation methodologies for the developer and the lab to demonstrate compliance with the Security Functional Requirements.

SFR (Security Functional Requirement) – describes security functions that must be met by the TOE.

ST (Security Target) – describes and identifies the security properties of the TOE.

Target of Evaluation (TOE) – refers to a product or set of products that fulfill the requirements to encrypt/decrypt user data on a host machine. This includes all hardware, firmware and software used to satisfy the requirements of this PP.

TOE Security Functionality (TSF) – a set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TOE Security Policy (TSP) – a set of rules that regulate how assets are managed, protected and distributed within a TOE.

TOE Summary Specification (TSS) – a narrative describing how the TOE meets the SFRs in enough detail so that one can understand the operation of the TOE and the implementation of the security functional requirements.

Unified Extensible Firmware Interface (UEFI) - A possible replacement for the conventional BIOS that is becoming widely deployed in new x86-based computer systems. The UEFI specifications were preceded by the EFI specifications.

Volatile memory – memory that loses its content when power is turned off.

Appendix F: PP Identification

Tile:	Protection Profile for PC Client Devices
Version:	1.0
Sponsor:	National Security Agency (NSA)
CC Version:	Common Criteria for Information Technology Security Evaluation (CC) Version 3.1 Revision 3, July 2009
Keywords:	Authenticated BIOS update, BIOS, RTU