Full Drive Encryption Security Problem Definition – Authorization Acquisition

Introduction for the FDE Collaborative Protection Profiles (cPPs) Effort

The purpose of the first set of Collaborative Protection Profiles (cPPs) for *Full Drive Encryption (FDE): Authorization Acquisition (AA)* and *Encryption Engine (EE)* is to provide requirements for Data-at-Rest protection for a lost device. These cPPs allow FDE solutions based in software and/or hardware to meet the requirements. The form factor for a storage device may vary, but could include: system hard drives/solid state drives in servers, workstations, laptops, mobile devices, tablets, and external media. A hardware solution could be a Self-Encrypting Drive or other hardware-based solutions; the interface (USB, SATA, etc.) used to connect the storage devices to the host machine is outside the scope.

Full Drive Encryption encrypts all data (with certain exceptions) on the storage device and permits access to the data only after successful authentication to the FDE solution. The exceptions include the necessity to leave a portion of the storage device (the size may vary based on implementation) unencrypted for such things as the Master Boot Record (MBR) or other AA/EE pre-authentication software. These FDE cPPs interpret the term "full drive encryption" to allow FDE solutions to leave a portion of the storage device unencrypted so long as it contains no user or authorization data.

Since the FDE cPPs support a variety of solutions, two cPPs describe the requirements for the FDE components shown in Figure 1.

The *FDE cPP - Authorization Acquisition* describes the requirements for the Authorization Acquisition piece and details the necessary security requirements and assurance activities necessary to interact with a user and result in the availability of a data encryption key (DEK).

The *FDE cPP - Encryption Engine* describes the requirements for the Encryption Engine piece and details the necessary security requirements and assurance activities for the actual encryption/decryption of the data by the DEK. Each cPP will also have a set of core requirements for management functions, proper handling of cryptographic keys, updates performed in a trusted manner, audit and self-tests.

Figure 1: FDE Components

This SPD defines the scope and functionality of the Authorization Acquisition as well as the assumptions made about the operating environment and the threats to the AA that the cPP requirements address.

Implementations

Full Disk Encryption solutions vary with implementation and vendor combinations.

Therefore, vendors will evaluate products that provide both components of the Full Disk Encryption Solution (AA and EE) against both cPPs. A vendor that provides a single component of a FDE solution would only evaluate against the applicable cPP. The FDE cPP is divided into two documents to allow labs to independently evaluate solutions tailored to one cPP or the other. When a customer acquires an FDE solution, they will either obtain a single vendor product that meets the AA + EE cPPs or two products, one of which meets the AA and the other of which meets the EE cPPs.

The table below illustrates a few *examples* for certification.

Table 1: Examples of cPP Implementations

Implementation	cPP	Description
Host	AA	Host software provides the interface to a self-encrypting drive
SED	EE	A self-encrypting drive used in combination with separate host software
Software FDE	AA + EE	A software full drive encryption solution
Hybrid	AA + EE	A single vendor's combination of hardware (e.g. hardware encryption engine, cryptographic co-processor) and software

Target of Evaluation (TOE) Description

The Target of Evaluation (TOE) for this cPP (Authorization Acquisition) may be either a Host software solution that manages a HW Encryption Engine (e.g. a SED) or as part of a combined evaluation of this cPP and the Encryption Engine cPP for a vendor that is providing a solution that includes both components.

The following sections provide an overview of the functionality of the FDE AA as well as the security capabilities.

Authorization Acquisition Introduction

The Authorization Acquisition sends Key Encryption Keys (KEKs) and/or Key Releasing Keys (KRKs) to the Encryption Engine. The EE does not have to use this value directly as the key to decrypt or release the DEK. It may use it as part of a scheme that uses other intermediate keys to eventually protect the DEK. Key Encryption keys (KEKs) wrap other keys, notably the DEK or

other intermediary keys which chain to the DEK. Key Releasing Keys (KRKs) authorize the EE to release either the DEK or other intermediary keys which chain to the DEK.) Figure 2 illustrates the components within AA and its relationship with EE.

Authorization factors may be unique to individual users or may be used by a group of individuals. In other words, the EE requires authorization factors from the AA to establish that

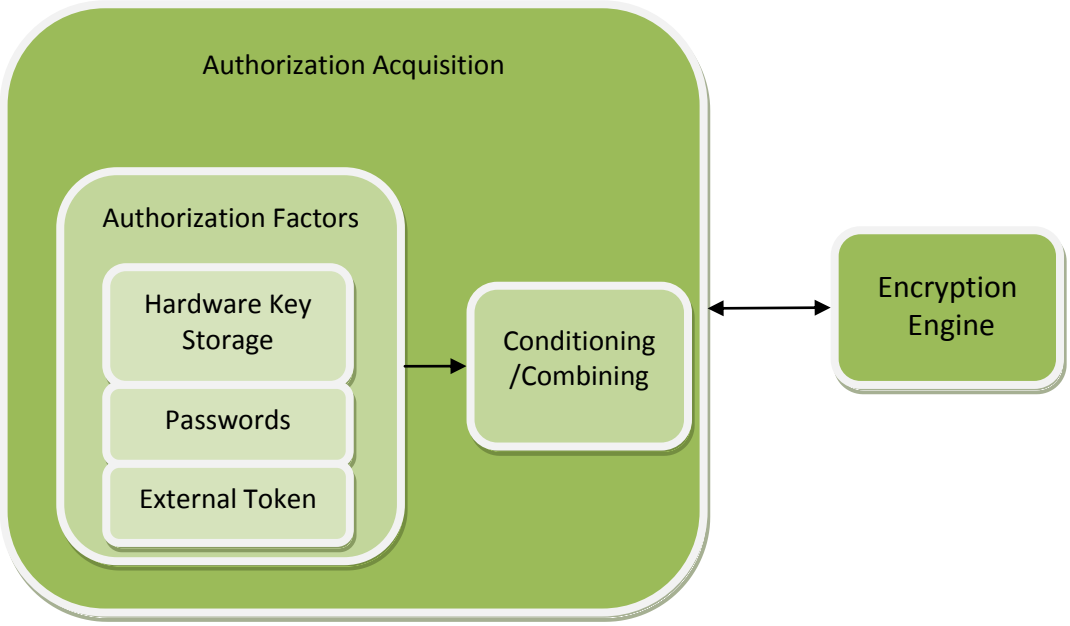

Figure 1: Authorization Acquisition Details

the possessor of the authorization factors belongs to the community of users authorized to access information stored on the storage device (and does not require specific user authorization). Examples of authorization factors include, but are not limited to, passwords, passphrases, or randomly generated values stored on USB tokens or a pin to release a key on hardware storage media such as a Trusted Platform Module (TPM).

Authorization Acquisition Security Capabilities

The AA collects authorization factors which the EE uses to access data on the storage device and perform a variety of management functions. Depending on the type of authorization factor, the AA may condition them further. For example, it may apply an approved password-based key derivation function (e.g. PBKDF2) on passwords. An external token containing a randomly generated value of sufficient strength may require no further conditioning on the authorization factors. The AA may then combine one or more authorization factors in such a way that maintains the strength of both factors.

The AA serves as the main management interface to the EE. However, the EE may also offer management functionality. The requirements in the EE cPP address how the EE should handle these features. The management functionality should include the ability to send commands to the EE such as changing a DEK, setting up new users, managing KEKs and other intermediate

keys, and performing a cryptographic erase (e.g. overwrite of the DEK). It may also forward commands that partition the drive for use by multiple users. However, this document defers the management of partitions and assumes that administrators and users will only provision and manage the data on whole drives.

Interface/Boundary

The interface and boundary between the AA and the EE will vary based on the implementation. If one vendor provides the entire FDE solution, then it is may choose to not implement an interface between the AA and EE components. If a vendor are provides a solution for one of the components, then the assumptions below state that the channel between the two components is sufficiently secure. Although standards and specifications exist for the interface between AA and EE components, the cPP does not require vendors to follow the standards in this version.

The TOE and the Operational/Pre-Boot Environments

The environment in which the AA functions may differ depending on the boot stage of the platform in which it operates, see Figure 3. Aspects of provisioning, initialization, and perhaps authorization may be performed in the Pre-Boot environment, while encryption, decryption and management functionality are likely performed in the Operating System environment.

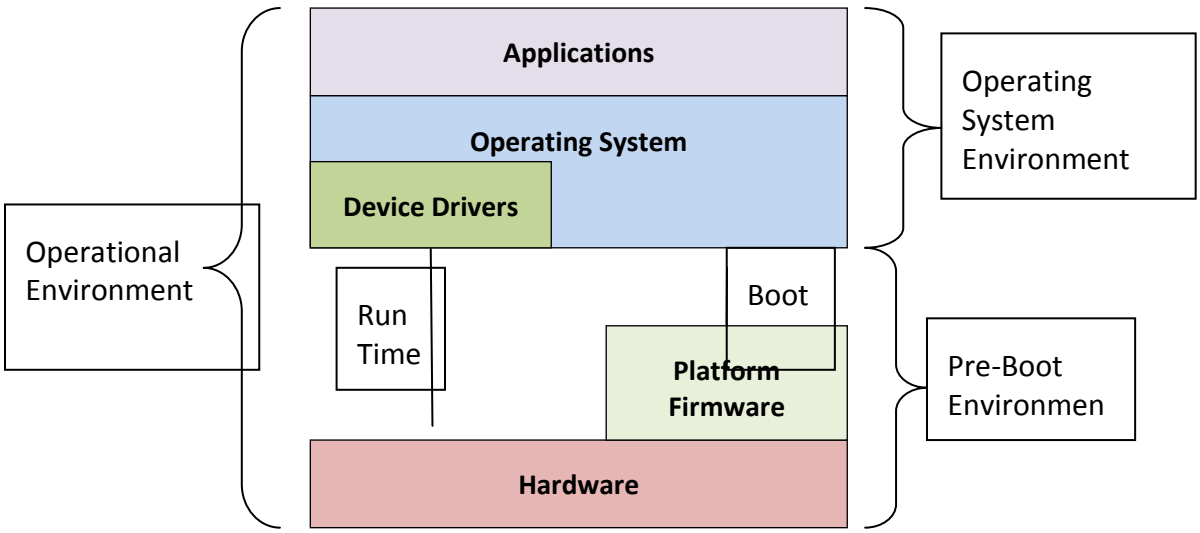

Figure 3: Operational Environment

In the Operating System environment, the Authorization Acquisition has the full range of services available from the operating system (OS), including hardware drivers, cryptographic libraries, and perhaps other services external to the TOE.

The Pre-Boot environment is much more constrained with limited capabilities. This environment turns on the minimum number of peripherals and loads only those drivers

99 necessary to bring the platform from a cold start to executing a fully functional operating
100 system with running applications.

101 The AA TOE may include or leverage features and functions within the operational
102 environment.

103 Functionality Deferred until Next cPP Version

104 Due to time constraints, this SPD defers requirements for some important functionality until
105 the next version of the cPP. These include requirements for partition/volume management,
106 remote management, and power management (requirements for power state protection).

107 Threats

108 This section provides a narrative that describes how the requirements mitigate the mapped
109 threats. A requirement may mitigate aspects of multiple threats. A requirement may only
110 mitigate a threat in a limited way.

111 A threat consists of a threat agent, an asset and an adverse action of that threat agent on that
112 asset. The threat agents are the entities that put the assets at risk if an adversary obtains a lost
113 or stolen storage device. Threats drive the functional requirements for the target of evaluation
114 (TOE). For instance, one threat below is T.UNAUTHORIZED_DATA_ACCESS. The threat agent is
115 the possessor (unauthorized user) of a lost or stolen storage device. The asset is the data on the
116 storage device, while the adverse action is to attempt to obtain those data from the storage
117 device. This threat drives the functional requirements for the storage device encryptor (TOE) to
118 authorize who can use the TOE to access the hard disk and encrypt/decrypt the data. Since
119 possession of the KEK, DEK, intermediate keys, authorization factors, submasks, and random
120 numbers or any other values that contribute to the creation of keys or authorization factors
121 could allow an unauthorized user to defeat the encryption, this SPD considers keying material
122 equivalent to the data in importance and they appear among the other assets addressed below.

123 It is important to reemphasize at this point that this Collaborative Protection Profile does not
124 expect the product (TOE) to defend against the possessor of the lost or stolen hard disk who
125 can introduce malicious code or exploitable hardware components into the Target of Evaluation
126 (TOE) or the Operational Environment. It assumes that the user physically protects the TOE and
127 that the Operational Environment provides sufficient protection against logical attacks. One
128 specific area where a conformant TOE offers some protection is in providing updates to the
129 TOE; other than this area, though, this PP mandates no other countermeasures. Similarly, these
130 requirements do not address the “lost and found” hard disk problem, where an adversary may
131 have taken the hard disk, compromised the unencrypted portions of the boot device (e.g., MBR,
132 boot partition), and then made it available to be recovered by the original user so that they
133 would execute the compromised code.

(T.UNAUTHORIZED_DATA_ACCESS)

The cPP addresses the primary threat of unauthorized disclosure of user data stored on a storage device. If an adversary obtains a lost or stolen storage device (e.g., a storage device contained in a laptop or a portable external storage device), they may attempt to connect a targeted storage device to a host of which they have complete control and have raw access to the storage device (e.g., to specified disk sectors, to specified blocks).

(T.KEYING_MATERIAL_ COMPROMISE)

Possession of any of the keys, authorization factors, submasks, and random numbers or any other values that contribute to the creation of keys or authorization factors could allow an unauthorized user to defeat the encryption. The cPP considers possession of keying material of equal importance to the data itself. Threat agents may look for keying material in unencrypted sectors of the storage device and on other peripherals in the operating environment (OE), e.g. BIOS configuration, SPI flash, or TPMs.

(T.AUTHORIZATION_GUESSING)

Threat agents may exercise host software to repeated guess authorization factors, such as passwords and pins. Successful guessing of the authorization factors may cause the TOE to release DEKs or otherwise put it in a state in which it discloses protected data to unauthorized users.

(T.PERSISTENT_INFORMATION)

As a courtesy to the user, the TOE and/or the Operational Environment goes into power saving mode in which it leaves the data or key material unencrypted in persistent memory to facilitate a speedy recovery upon powering up. Threat agents look for unencrypted keying material and data giving them unauthorized access to data.

(T.UNAUTHORIZED_UPDATE)

Threat agents may attempt to perform an update of the product which compromises the security features of the TOE. Poorly chosen update protocols, signature generation and verification algorithms, and parameters may allow attackers to install software and/or firmware that bypasses the intended security features and provides them unauthorized to access to data.

(T.KEY_RECOVERY)

Threat agents may attempt to access an archive of keys and perform an attack or brute force exhaust to recover those keys, thus providing them with unauthorized access to the data. A poorly chosen key recovery process using cryptographically weak algorithms and protocols could give threat agents access to the user's data.

Assumptions

Assumptions that must remain true in order to mitigate the threats appear below:

(A. INITIAL_DRIVE_STATE)

Users enable Full Drive Encryption on a newly provisioned or initialized storage device free of user data. The cPP does not intend to include requirements to find all the areas on storage devices that potentially contain user data. In some cases, it may not be possible - for example, data contained in “bad” sectors.

While inadvertent exposure to data contained in bad sectors or un-partitioned space is unlikely, one may use forensics tools to recover data from such areas of the storage device. Consequently, the cPP assumes bad sectors or un-partitioned space contains no user data.

(A.SECURE_STATE)

Upon the completion of proper provisioning, the drive is only assumed secure when in a powered off state up until it is powered on and receives initial authorization.

(A.TRUSTED_CHANNEL)

Communication among and between product components (e.g., AA and EE) is sufficiently protected to prevent information disclosure. In cases in which a single product fulfills both cPPs, then it assumes that the communication between the components does not breach the boundary of the TOE. In cases in which independent products satisfy the requirements of the AA and EE, the physically close proximity of the two products during their operation means that the threat agent has very little opportunity to interpose itself in the channel between the two without the user noticing and taking appropriate actions.

(A_AUTHORIZED_USER)

Authorized users follow all provided user guidance, including keeping password/passphrases and external tokens securely stored separately from the storage device and/or platform.

(A.PLATFORM_STATE)

The platform in which the storage device resides (or an external storage device is connected) is free of malware that could interfere with the correct operation of the product.

(A.EXTERNAL_AUTH_USE)

External tokens that contain authorization factors are used for no other purpose than to store the external token authorization factors.

(A.MEMORY_REMNANCE)

The user does not leave the platform and/or storage device unattended until FDE solution clears all volatile memory after a power-off, so memory remnant attacks are infeasible.

Authorized users do not leave the platform and/or storage device in a mode where sensitive information persists in non-volatile storage (e.g., Lockscreen). Users power the platform and/or storage device down or place it into a power managed state, such as a “hibernation mode”.

(A.PASSWORD_STRENGTH)

Authorized administrators ensure password/passphrase authorization factors have sufficient strength and entropy to reflect the sensitivity of the data being protected.

(A.PLATFORM_I&A)

The product does not interfere with or change the normal platform identification and authentication functionality such as the operating system login. It may provide authorization factors to the Operating system's login interface, but it will not change or degrade the functionality of the actual interface.

(A.STRONG_CRYPTO)

All cryptography implemented in the Operational Environment and used by the product meets the requirements listed in the cPP. This includes generation of external token authorization factors by a RBG.

(A.TRAINED_ADMINS)

Authorized administrators are appropriately trained and follow all administrator guidance.