Supply Chain Threats – Test, Provision, and Validation

White Paper, v1.0

April 2022

Author: Matthew Areno, PhD

Contents

1 Introduction

Supply Chain security is an area of significant focus and scrutiny today. A number of recent significant attacks against both hardware and software supply chains have further exposed the criticality of understanding the threats posed against manufacturers and providing sufficient mitigations. Understanding and mitigating supply chain threats starts with simply identifying the threats.

The purpose of this document is to further expand on prior work and dive into the testing, provisioning, and validation phases. These phases represent potentially the most significant threat against products as they are often conducted, at least in part, by external organizations. Without an understanding of all the known threats currently present, it would be impossible to make any assertions regarding the overall security of products and their associated supply chain.

As there is no singular process for these stages of the silicon manufacturing process, the process detailed in this document is still generic and represents a *common* approach to the test, provision, and validation phases. Readers should personalize the process presented to their own, while recognizing the threats presented may or may not be present, or may be slightly different, but still provide a good starting point for assessment of their own processes.

Various test/manufacturing processes and assets are described in this white paper. These are example processes and assets used in the industry and are described solely for the purposes of the discussion of supply chain issues in this white paper. These processing and assets are not meant to necessarily represent those processes and assets of Intel Corporation.

1.1 Acronyms

Term	Description
ATE	Automated Test Equipment
ATM	Assembly/Test/Manufacturing
BORE	Break Once, Run Everywhere
CA	Certificate Authority
DRNG	Deterministic Random Number Generator
FPF	Field Programmable Fuses
GKEK	Global Key-Encryption-Key
HVM	High Volume Manufacturing
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
PRQ	Product Readiness Qualifications
RNG	Random Number Generator

2 Test and Validation Overview

The Test and Validation phase of the Supply Chain Lifecycle is fairly complex and often occurs on multiple continents. There are several stages and transitions that occur during the Test and Validation phase and an understanding of each is critical in order to properly establish terminology and to identify threats that exist. This section provides an overview of each stage of the Test and Validation phase. Later sections will provide specific threats for each stage based off the efforts and events that occur in each stage identified in this section.

It should be noted that this section covers a single use case of test and validation. Companies may deviate from this process or use something completely unique. Further, this process may be different within a single company based on the type of product being tested and verified. As such, the information below should be considered generic, but fairly accurate for a large number of manufacturers.

2.1 Assembly Completed

Assembly Completed occurs once a part has, as the name indicates, been completely assembled. The effort here is mostly a validation of the assembly process to ensure that the part is structurally intact and is ready to begin formal testing. Because this is performed at the conclusion of assembly, it will most often occur overseas (outside of the United States) at an assembly/test/manufacturing (ATM) site.

At this stage, the parts are not yet provisioned. This means that fuses have yet to be programmed on the parts and that debug access is typically restricted (by default in most hardware implementations). However, in the restricted state a single unlock sequence is often used to unlock debug capabilities, and this sequence is identical across all products.

2.2 Class Testing

Class Testing represents the first round of tests performed on parts after they have left the Assembly stage. This stage will also likely occur at an overseas ATM site, and is performed exclusively through the use of automated test equipment (ATE). The testing performed at this point is mostly geared towards basic power sequencing to ensure that the part is electrically sound, can be turned on, and is ready for fusing.

The parts are in the same state from a provisioning and security standpoint during this phase as they were during Assembly Complete. No fuses have been blown yet and debug functionality is usually still restricted.

2.3 Fusing

The Fusing stage entails the initial provisioning of the majority of fuses within processing elements. The fuses are used to store critical information, such as unique identifiers and cryptographic material, as well as to control functionality of the processors, such as how many of the available cores may be used. Fuses are divided between high volume manufacturing (HVM) fuses that are programmed by the manufacturer, and field programmable fuses (FPF) that may be programmed by an original equipment manufacturer (OEM), equipment owner, or end customer.

Fusing of HVM fuses is performed via ATE and is also typically done at an overseas ATM. Once provisioned with the fuse values, the part is transitioned into a “post-manufacturing” state which enables traditional debug access in preparation for the start of system testing.

2.4 Manufacturing System Testing

Manufacturing System Testing encompasses a number of basic tests focused on proof of functionality. This stage is also referred to as Pre-Processor Validation or System-Level Test. Common tests during this stage include initial boot up of the part, complete boot of an Operating System, screening against defined test conditions, and verification of some security features.

It should be noted that this stage usually does not attempt to perform negative testing or exhaustively test all features. The majority of all testing is positive only, and centers around functionality of various technologies within the processing element. This stage, like all other prior stages, usually occurs at an overseas ATM via ATE.

2.5 Validation

The validation stage includes negative testing of technologies, capabilities, and features within a processing element, specifically designed to ensure they are functioning as expected. This stage is performed prior to the start of HVM to verify the functionality of technologies and capabilities integrated into the product. This is referred to as the Product Readiness Qualifications (PRQ). Once PRQ has been completed, it is assumed the outcome will be identical for all other components created during the HVM phase.

Based on the breadth of products manufactured, this stage may be performed by a separate party that has contracted with the original manufacturer to manufacture the product. As such, the validation of the product would fall under the responsibility of the contracting party.

In contrast with other stages, the validation stage may be conducted at a variety of manufacturer locations across the world or at customer locations. The determination of where this is performed is based on a number of different variables, although customers seeking specific accommodations can typically work with manufacturers to ensure performance at specific locations or at their own location.

2.6 Packing/Shipping

The final stage in the Test and Verification process is the Packing/Shipping phase. This phase is likely performed at the same location as the Validation phase and is responsible for packaging of products that have passed all required tests and shipping them to their appropriate distribution location. The ultimate location may be another distribution facility, an OEM, or another end customer.

Packaging and shipping are also responsible for record keeping and tracking of products both coming to and leaving from the facility. Products are recorded and customer information is maintained that is crucial to both warranty and return support, as well as overall supply chain security.

3 Protection of Assets

A large percentage of manufactured products contain high value assets, most often in the form of cryptographic key material that is used in the protection and/or authentication of proprietary and user data. It is well established that such assets are of significant interest to attackers around the world due to the capabilities they provide. This section provides information on these assets and existing procedures for handling such assets. Potential attacks to compromise these assets are addressed in a later section.

For the purposes of this section, the information provided will be associated with standard commodity processors. Other processing elements may also fall into this category. However, this section will not attempt to provide a comprehensive listing for all manufactured products. Other assets may exist for specific products and should be accounted for by their manufacturer.

3.1 Cryptographic Key Material

Many modern processors include a global key-encryption-key (GKEK) that is used to decrypt other encryption keys stored in some form of non-volatile memory, such as fuses or flash. This key may be stored directly in logic on the chip, generated at power-on via a physically unclonable function (PUF) or recreated using some other form of deterministic random number generator (DRNG). The GKEK may be used to decrypt the fuses themselves or may be used to instead decrypt a separate key (Fuse Key) that is then used to decrypt the fuses, hence the KEK title.

Depending on how the GKEK is stored or created, it may be unique per product type or unique per family. If the key is stored directly in the logic, it will be unique for each family or product version of processors. The GKEK may also be a combination of separate vectors stored in different types of non-volatile memory (NVM)[1]. This would allow some vectors to remain static across iterations of the device, while allowing the overall GKEK to change by device or device iteration by merely changing a vector stored in programmable, non-volatile memory, such as fuses.

Modern processors are also moving towards adoption of physical unclonable functions (PUFs). A PUF is a source of a unique-per-part collection of bits often resulting from measurable variations in manufacturing processes. These bits may then be used directly as a cryptographic key, may be feed into a random number generator (RNG) as a seed, or may be used in other ways to re-create a desired key value. Most PUFs require storage of static data referred to as *helper data* that is used to ensure the proper recreation of the collection of bits across reboots and environmental changes. While this helper data is intended to be non-sensitive, it is often treated with similar security concerns as other sensitive cryptographic material.

Symmetric keys, such as GKEKs or PUF-derived keys, are not the only cryptographic keys stored in hardware or fuses. Depending on the product, a number of asymmetric keys may also be included in fuses. Asymmetric keys are actually keypairs composed of a private and public component. It is commonplace for the public portion of the keypair, or a cryptographic hash of the public key, to be stored in fuses. A hash of the key is often used as it requires fewer bits and thus fewer fuses.

It is unusual to store the private portion of the keypair in hardware or fuses. The most likely implementation would be to store a secret *seed* that may be fed to a Key Generation Function (KGF) in order to regenerate

[1] https://csrc.nist.gov/publications/detail/sp/800-133/rev-1/archive/2019-07-23, section 6.6, pp 16

the asymmetric keypair upon each boot. In this scenario, the secret seed is protected in much the same manner as a standard symmetric key.

3.2 Fuse Provisioning

Due to the significant interest attackers have in extracting fuse values (which will be a major portion of the threat vector section), it is necessary to detail how the fuse values are generated, handled, and programmed. For this section, the fuses are divided between functional and security related, with the majority of the emphasis on the security fuses and how they are combined with the functional fuses. This should not be considered a judgment of the importance of functional fuses versus security fuses, but rather simply based on the timeline and associated threats.

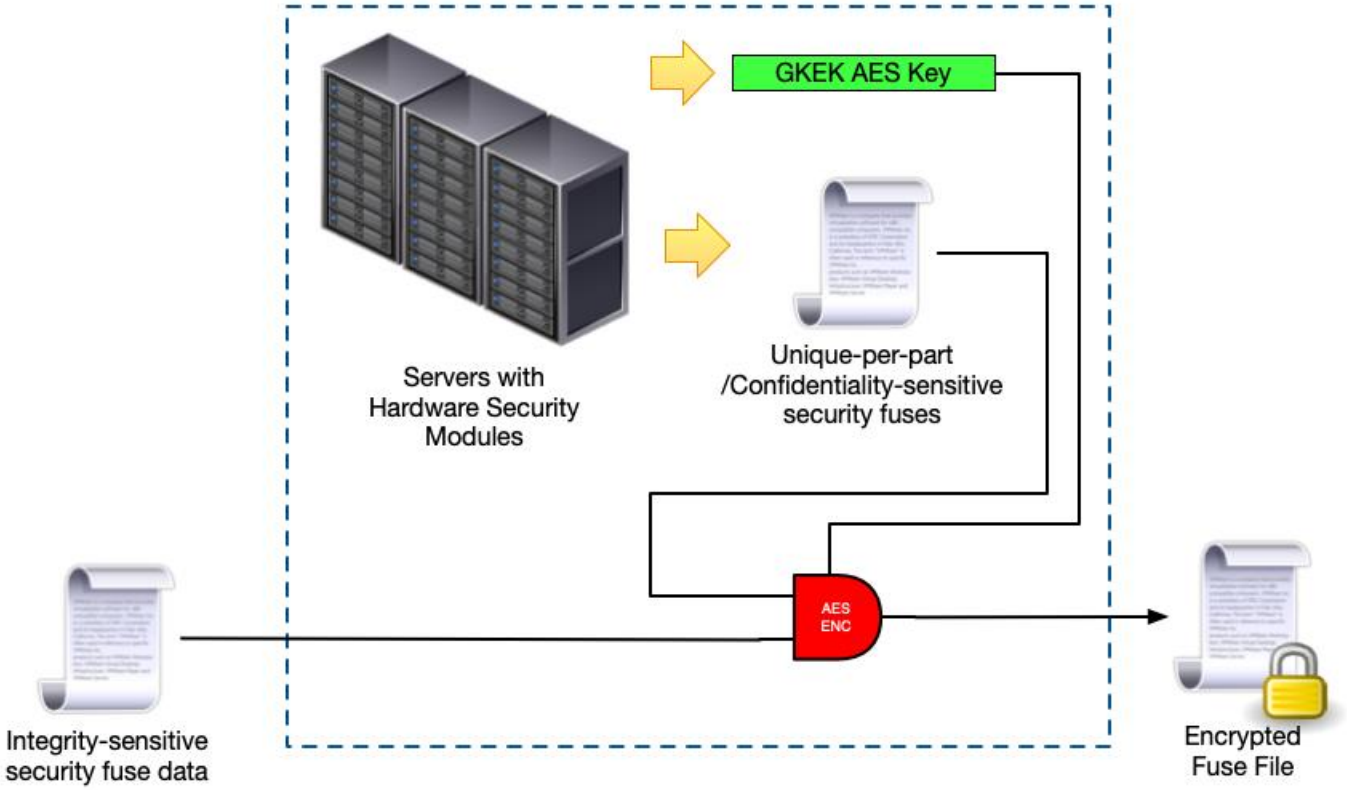

Figure 1 - Generation of Fuse File

3.2.1 Fuse Generation

The creation of the fuses is a process specific to each manufacturer. Typically, a request is made to utilize specialized hardware, called a Hardware Security Module (HSM), that is certified for the creation of cryptographically sound and unique values. These HSMs are used to create both device-unique and family-unique cryptographic keys. The output is what is referred to in this document as a *fuse file*.

The structure of the fuse file is not universally defined. However, the fuse file is expected to consist of both functional and security related fuses. For the purposes of this document, it is assumed that functional fuses are not encrypted and not generated by any specialized hardware. Security related fuses are assumed to be a combination of integrity-sensitive values and confidentiality-sensitive values. The confidentiality-sensitive fuse values may be generated by an HSM and never made available outside of the HSM in any unprotected

manner, whereas integrity-sensitive values are often established outside of an HSM but then sent to the HSM for cryptographic measurement and protection. Confidentiality-sensitive values are often encrypted, such as with the GKEK, and Integrity-sensitive values are included in a cryptographically signed portion of the overall fuse back. This process is illustrated in Figure 1.

A complete Fuse File consists of both the unencrypted functional fuse data combined with the encrypted security fuse data. Each generated Fuse File may be used to then provision one individual device, meaning a separate Fuse File should be generated for every product created, assuming the fuses contain unique-per-device values. Fuse Files are then packaged into a bundle of a predefined quantity and prepared for transmission to the associated ATM site.

3.2.2 Fuse Handling and Transportation

Once the fuse file bundles are created, they need to be transported to the ATM site. Depending on whether the manufacturer creates all of the fuse information themselves or contracts with a 3rd-party, the information may be submitted directly or may first need to be stored locally on the manufacturer site and then transmitted to the ATM site. Which of the two options happens likely depends on the structure of the Fuse File itself and how the functional and security fuses are combined. This is illustrated below in Figure 2.

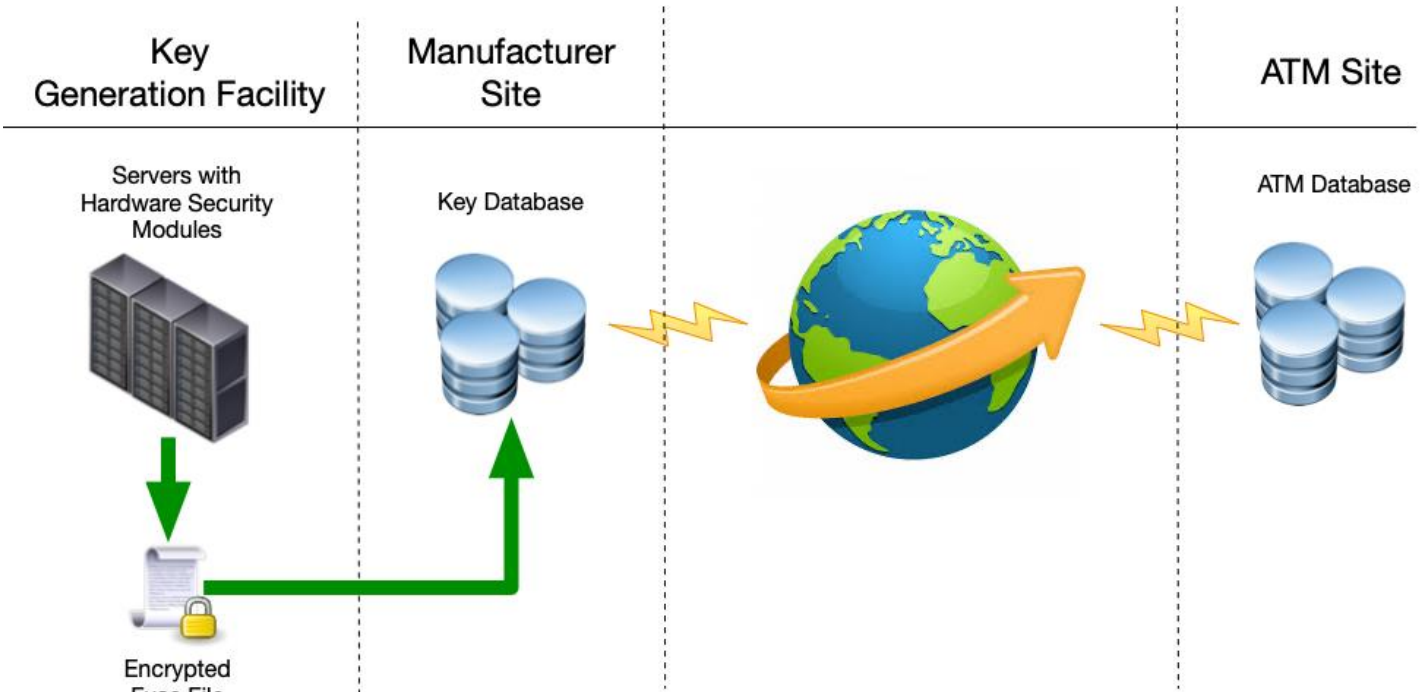

Figure 2 - Transmission of fuse files to ATM sites.

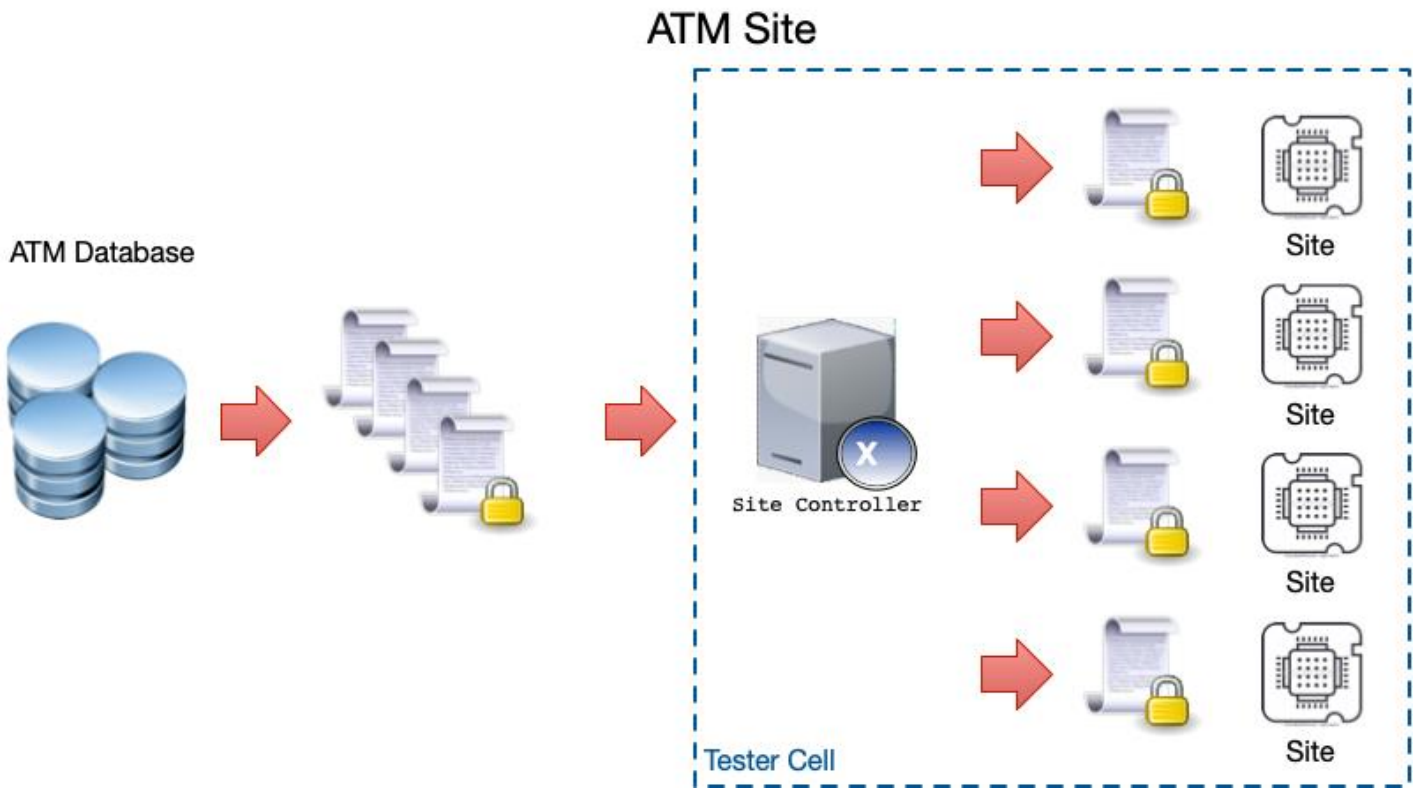

Figure 3 - Provisioning of fuse files into products

At the ATM site, a number of *Tester Cells* exist that are used to perform the testing and provisioning on each device. The Tester Cell consists of a *Site Controller* with one or more *Sites* for interacting with the individual devices (shown in Figure 3). The Site Controller interfaces directly with the local ATM database whenever it is ready to begin processing of a new batch of devices. A corresponding number of fuse files are requested and maintained on the Site Controller, then distributed to the individual sites in order to be programmed into the attached device. Once programming is completed, any necessary information is transferred back to the ATM database.

Maintenance of fuse files and any generated data during the process is subject to standard retention practices. This usually results in removal within a predefined window from the time of the fuse files was programmed into a device.

3.2.3 Fuse Programming

Once the fuse files are received by the site controller, the fuse data is written to the respective devices. This includes all functional fuses and usually all security fuses. How the programming happens depends on whether the device uses a GKEK or a fuse key for protection of the security fuses. Regardless, the fuses are programmed with the contents of the fuse file. Once completed, a validation process may be performed that would require the device to decrypt the encrypted fuses with the GKEK or some other fuse key and ensure certain functionality. This potential step will be very manufacturer specific.

Fuse provisioning of individual devices may also be divided and done in chunks rather than all at once. Many devices may include what are called *lockout* fuses or *end-of-manufacturing* fuses. This allows the device to support enhanced introspection, control, and validation while the device is inside the ATM, but then to disable such access once the associated work is completed, and before the device leaves the manufacturer facility. As a result, fuse provision may occur over a number of phases as opposed to a single event.

3.2.4 On-Device Certificate Authority

Recently developed specifications, such as the Trusted Computing Group's *Device Identifier Composition Engine*[2], are supporting the ability for components to be provisioned with their own certificate authority (CA) style capabilities. Devices are provisioned with a *unique device secret* (UDS) that can be used as a seed for generation of an asymmetric key pair. When this occurs, a certificate signing request (CSR) is generated with the public portion of that key pair. The CSR is created during the provisioning process and signed locally by the manufacturer or sent to a remote location and signed there.

This process will be heavily dependent on the manufacturer and owner of the ATM site. This process may also not happen at the ATM site, but instead occur after the part completes test and validation and is returned to the original manufacturer. This is often the case when the original manufacturer and the owner of the ATM are not the same organization or company.

[2] https://trustedcomputinggroup.org/work-groups/dice-architectures/

4 Threat Model

A threat model is meant to provide a method of identifying and illustrating potential attack vectors against a certain product, procedure, or methodology. A previous threat model was created to cover the entire Trusted Supply Chain (TSC) lifecycle at a fairly high level[3]. The output of that model is shown below in Figure 4.

As mentioned previously, the purpose of this document is to expand into the Testing and Validation, or Provisioning/Configuring, cycles. This section presents a more in-depth threat model with expanded information on potential threats to the Testing and Provision/Configuration phases. Each threat will be illustrated in a model along with a later definition of the threat and how it might be exploited by an attacker.

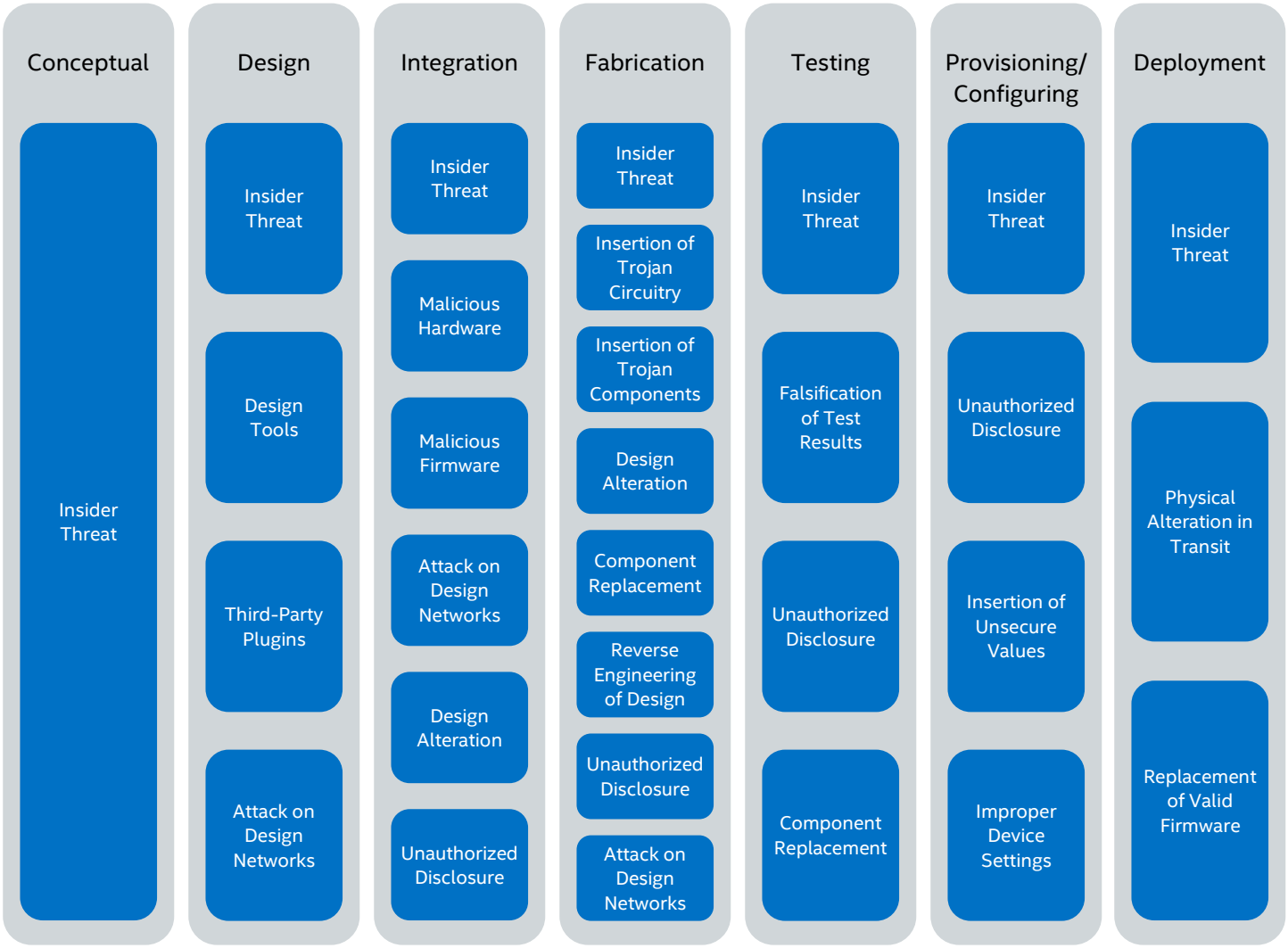

Figure 4 - High-level supply chain threat vectors.

[3] https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/supply-chain-threats-v1.pdf

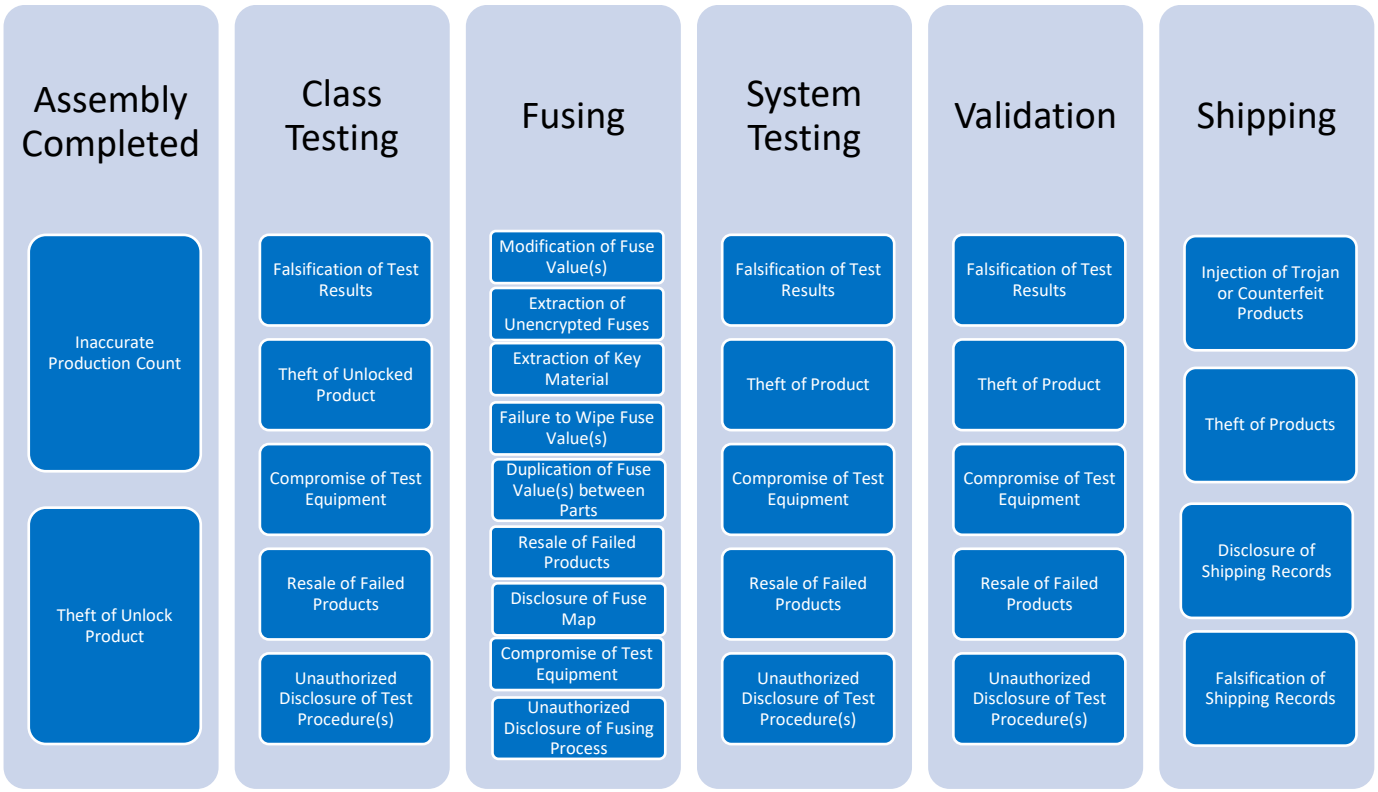

Figure 5 - Test and Validation Threat Model

4.1 Threat Definitions

The threat model for the Test and Validation cycles of the supply chain lifecycle is divided into six phases, as described previously in this document. The phases are: Assembly Completed, Class Testing, Fusing, System Testing, Validation, and Shipping. Each phase contains a number of potential threat vectors with some vectors existing across multiple phases. The resulting model is shown in Figure 5.

The remainder of this section is a listing of the definitions of each threat shown in the figure above. These definitions, many of which include examples of attacks, should be used to determine if sufficient mitigations already exist or if a gap has been discovered that will need to be addressed. The threat model presented is based off currently known or suspected attacks and is subject to future additions as new attack vectors are identified.

4.1.1 Inaccurate Production Count

Parts coming out of assembly are extremely sensitive until the provisioning process is completed. This is especially true during pre-HVM. Inaccurate production count could support untracked loss or theft of such critical components. Potential attacks could be performed on both valid and invalid parts. Although a part may not function fully for production usage, it could still be a candidate for reverse engineering or attempted exploitation of functional circuitry.

4.1.2 Theft of Unlocked Product

Parts that are not fully provisioned, in terms of fuses, represent the highest security risk as many attack mitigations are not yet functional on the component. The level of risk may be reduced at later stages of the

Test and Verification cycle. Until provisioning is completed, these parts should be handled, tracked, and accessed with the highest degree of scrutiny.

4.1.3 Falsification of Test Result(s)

Falsification of test results may result in functional units being declared as failed or may result in parts with compromised security features being declared valid and ready for use. The consequences of such falsification could result in a number of potential outcomes, including lost revenue and compromise of a company's proprietary information.

4.1.4 Compromise of Test Equipment

In many cases, the majority of direct access to products is through ATE. Compromise of the integrity of the execution of the ATE could result in potential Break-Once-Run-Everywhere (BORE) style attacks against hardware products. Additionally, such equipment often has access to sensitive values that are stored in Intel products, such as fuses with cryptographic material. Collection and subsequent extraction of this information may result in later compromise, or emulation, of valid fielded products.

4.1.5 Resale of Failed Product(s)

Any product that fails any stage of the Test and Validation cycle represents a potentially significant target for malicious parties. Factory workers are likely to be recipients of bribes meant to provide attackers with access to such parts with monetary compensation varying based on the stage at which the component failed. Such failures could be legitimate or may be the result of falsification of test results, as mentioned previously.

4.1.6 Unauthorized Disclosure of Test Procedure(s)

Access to test procedures could provide attackers with significant insight into what, and how, specific features or technologies are tested on Intel products. Test procedures, by nature, are not extensive enough to cover all possible scenarios. These documents would help to shrink attack vectors and help create or improve heuristics used by attackers.

4.1.7 Modification of Fuse Value(s)

Fuse values are used to control functionality and store critical data. Modification of such fuses, either pre-determined or arbitrary, can have a variety of problematic impacts on Intel products. This is especially true of any fuses that are not protected with any form of integrity protection. This problem may also exist on compromised ATE and may be done before the fuses are even transferred to the component.

4.1.8 Extraction of Unencrypted Fuse(s)

Unencrypted fuses consist of both fuses that are not encrypted at any time on the parts, as well as fuses that may be in an unencrypted state and potentially exposed to a malicious insider or compromised ATE. This will be especially true of any fuse material that is not encrypted beforehand due to the expectation of encrypting it later with a device specific value, such as a PUF-generated key. Keeping encrypted data together with an unencrypted key represents an obvious security risk, a risk that is further exacerbated if the data is not integrity protected in any manner.

4.1.9 Extraction of Key Material

Key material may be extractable on systems depending on the method used for generation of the GKEK. If the GKEK is stored in logic, the fuses file can be completely encrypted without ever being co-located, physically or logically, with the associated GKEK. The primary threat there would be during the Assembly process and not as relevant at the Test and Validation Phase. However, if the GKEK is generated on the device and thus inaccessible for encryption of the fuse file, a separate key is generated and used to encrypt the fuse file. This separate key then needs to be sent to its associated device and encrypted using the

generated GKEK for that device. Exposure of this separate key would allow for potential modification, or at a minimum introspection, of the associated fuse file and any key material contained therein.

4.1.10 Duplication of Fuse Value(s) Between Parts

Many products have a requirement of device-unique values that are often used for identification and attestation purposes. Any assumption based upon the uniqueness of these values could be violated if multiple parts were able to be provisioned using the same fuse file. This could be of particular interest to attackers who want to emulate a platform using virtualization technology or setup an alternate hardware environment that is seemingly identical to the expected environment of the user.

4.1.11 Disclosure of Fuse Map

There is an aspect of "security-through-obscurity" that is inherent to components with tens of thousands of fuses. Although this may not have been an intentional security feature, it nonetheless provides a level of security that would be lost if the fuse map were disclosed. Any information on the exact locations of high-value fuses would make an attacker's job significantly easier.

4.1.12 Unauthorized Disclosure of Fusing Process

Similar to the threat created by disclosure of the fuse map, disclosure of the fusing process itself is another significant threat. In conjunction with the threat of sale or re-sale of failing parts, a malicious party would know exactly where in the Test and Validation process the product is in its most vulnerable or most valuable state. Every compromise of information just allows attackers to be that much better and quicker at their work.

4.1.13 Theft of Product(s)

As mentioned previously, theft of products, especially prior to HVM or completion of provisioning, represents perhaps the most drastic of all security threats. Even if products are completely provisioned, theft reduces the cost impact to attackers and makes their job that much cheaper. A significant portion of mitigating threats is to make the cost of performing attacks targeting such threats as high as possible. The ability of an attacker to steal, or to purchase at a lower cost from someone else who stole the part, are both detriments to this mitigation.

4.1.14 Injection of Trojan or Counterfeit Product(s)

A malicious insider with access to products during the shipping phase may be able to inject trojan or counterfeit components into the supply chain. This could be done for a number of reasons, not all of which are malicious in nature. This could be done to tarnish the Intel brand-name or to try and get a malicious product into a specific targeted location or company, for instance. Regardless of the reason, it is imperative that end customers receive only authentic and accurate components for their orders.

4.1.15 Disclosure of Shipping Record(s)

Shipping records may not seem significantly sensitive from a security perspective, but they can provide a lot of information to potential attackers. It could allow them to determine which customers will be utilizing which products, thereby helping them to isolate attacks on specific products based upon their end target.

4.1.16 Falsification of Shipping Record(s)

Falsification of shipping records would allow a malicious insider to potentially record shipment of more products than was actually performed, creating an excess that could be sold on the side to a separate party. This could be to coverup a sale at any time during the Test and Validation cycle.

5 Conclusion

The threats identified in this document represent a significant risk to the product security of all manufacturers. While the threat landscape continues to evolve every day, these threats serve as a good starting point for any company or organization currently manufacturing products are intending to do so in the near future. However, any such threat document must be considered a living document and must evolve together with the threat landscape.

Supply chain security can be obtained and maintained. Organizations and companies must recognize the threat, identify gaps in mitigations, and create meaningful and achievable plans for addressing those gaps to provide the secure products all their customers deserve.