



**Table of Contents**

BACKGROUND AND THE ASSERTED PATENTS ..... 3

APPLICABLE LAW ..... 5

DISPUTED TERMS..... 8

1. “an asymmetric key pair” (’399 Patent Claims 1, 10, 29, 37) .....8

2. “predetermined data” (’399 Patent claims 1, 29).....15

3. “executable tamper resistant key module”/“executable tamper resistant code module”/“tamper resistant key module” (’399 Patent claims 1, 9, 29, 37) .....18

4. “selected program”/“program” (’399 Patent claims 1, 29, 37).....27

5. “including the generated private key and the encrypted predetermined data” (’399 Patent Claims 1, 29).....30

6. “integrity verification kernel” (’399 Patent Claims 9, 10; ’550 Patent Claim 15) .....36

7. “integrity verification kernel code” (’399 Patent Claim 10).....43

8. “manifest” (’399 Patent Claim 10; ’550 Patent Claim 16) .....46

9. “manifest parser generator code” (’399 Patent Claim 10).....51

10. “address space” (’550 Patent Claim 14) .....53

11. “first process” (’550 Patent Claims 14–17) / “second process” (’550 Patent Claim 14)....57

12. “embedded” (’550 Patent Claim 14).....61

13. “challenge” (’550 Patent Claims 14, 17) .....63

## **BACKGROUND AND THE ASSERTED PATENTS**

Plano Encryption Technologies, LLC (“PET”) brought actions against American Bank of Texas, Independent Bank, Guaranty Bank & Trust, N.A., Citizens Nation Bank, and Broadway National Bank (collectively, “Defendants”) alleging that Defendants infringe U.S. Patent Nos. 5,974,550 (the “’550 Patent”) and 5,991,399 (the “’399 Patent”), (collectively, “the Asserted Patents”).

The ’550 Patent and the ’399 Patent are based on separate patent applications each filed in December 1997. Though both patents generally relate to security techniques for computer systems, the patents do not stem from a common patent family. Thirteen groupings of claim disputes are presented to the Court. Three of those claim dispute groupings include terms that are found in both patents. The disputed terms of the ’550 Patent are found in claims 14–17 and the disputed terms of the ’399 Patent are found in claims 1, 9, 10, 29, and 37. (Dkt. Nos. 88–1, 88–2.)

In general, the ’550 Patent relates to a remote security protocol in computer systems in which two processes are operating. Techniques are described for authenticating a first process that operates in an address space different than that of a second process. ’550 Patent 1:7–10, 1:44–46. As an example, the processes may relate to electronic commerce such as a consumer purchase, where one process asks another process for a service to be performed. *Id.* at 2:20–22. The authenticating techniques described include the use of a challenge-response protocol. The second process may create a module containing a secret and send the module and a challenge to the first process. The first process executes the module and recovers the secret when the first process is verified by the module. The first process uses the secret to produce a response and then sends the response to the second process. *Id.* at 1:44–54, FIG. 2.

The '550 Patent abstract recites:

Authenticating a remote process operating in an address space different than that of a local process includes the steps of creating, by the local process, a tamper resistant module containing a temporary secret, sending the tamper resistant module and a challenge from the local process to the remote process, executing the tamper resistant module by the remote process and recovering the secret when the integrity of the remote process is verified by the tamper resistant module, encoding the challenge using the secret to produce a response, sending the response to the local process, and decoding the response by the local process. Optionally, the tamper resistant module includes a request for information from the second process and the response includes the answer to the request for information.

'550 Abstract.

In general, the '399 Patent relates to digital content protection in computer systems. Techniques are described for distributing a private key over a network so only a specific trusted player can use the private key to access encrypted digital content. '399 Patent 1:7–11. More particularly, an asymmetric key pair having a public key and private key may be generated. Data may be encrypted with the public key. An executable tamper resistant key module is built, which includes the private key and the encrypted data. The executable tamper resistant key module may then be sent to a remote system. The tamper resistant key module is then executed on a remote system as part of a validation process. If the validation process is successful, then the data is decrypted with the private key included in the tamper resistant key module. *Id.* at 3:5–20. As an example, an embodiment of the techniques can be used to provide a private key to a DVD player. Verification of the DVD player's integrity and authenticity allows the DVD player to use the private key to decrypt digital content. *Id.* at 3:53–61. The '339 Patent abstract recites:

Secure distribution of a private key to a user's application program (also called a "trusted player" such as a DVD player or CD-ROM player) with conditional access based on verification of the trusted player's integrity and authenticity is provided. Once validated, the trusted player uses the private key to decrypt encrypted digital content. The private key is dynamically generated, associated with specific digital content, and communicated in real-time from a server to the

trusted player in a secure manner, thereby controlling access to encrypted digital content. The key is wrapped into an executable tamper resistant key module in which the key can only be used by the right trusted player as determined by the server based on user requests and payment. The key module plugs in to the trusted player and executes to validate the player and decrypt the content. The integrity of the trusted player is correlated to its ability to perform a cryptographic operation using an asymmetric key pair in a manner that is tamper resistant, thereby preventing an unencrypted copy of digital content to be made.

'399 Abstract.

### **APPLICABLE LAW**

“It is a ‘bedrock principle’ of patent law that ‘the claims of a patent define the invention to which the patentee is entitled the right to exclude.’” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)).

Claim construction is a legal issue that may be based on underlying findings of fact. *Teva Pharm. USA, Inc. v. Sandoz, Inc.*, 135 S. Ct. at 841. “In cases where [ ] subsidiary facts are in dispute, courts will need to make subsidiary factual findings about that extrinsic evidence. These are the ‘evidentiary underpinnings’ of claim construction that we discussed in *Markman*, and this subsidiary factfinding must be reviewed for clear error on appeal.” *Id.* (citation omitted).

To determine the meaning of the claims, courts start by considering the intrinsic evidence. *Id.* at 1313; *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 861 (Fed. Cir. 2004); *Bell Atl. Network Servs., Inc. v. Covad Commc’ns Grp., Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). The intrinsic evidence includes the claims themselves, the specification, and the prosecution history. *Phillips*, 415 F.3d at 1314; *C.R. Bard, Inc.*, 388 F.3d at 861. Courts give claim terms their ordinary and accustomed meanings as understood by one of ordinary skill in

the art at the time of the invention in the context of the entire patent. *Phillips*, 415 F.3d at 1312–13; *Alloc, Inc. v. International Trade Comm’n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

The claims themselves provide substantial guidance in determining the meaning of particular claim terms. *Phillips*, 415 F.3d at 1314. First, a term’s context in the asserted claim can be very instructive. *Id.* Other asserted or unasserted claims can also aid in determining the claim’s meaning, because claim terms are typically used consistently throughout the patent. *Id.* Differences among the claim terms can also assist in understanding a term’s meaning. *Id.* For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id.* at 1314–15.

“[C]laims ‘must be read in view of the specification, of which they are a part.’” *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)). “[T]he specification ‘is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.’” *Id.* (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). This is true because a patentee may define his own terms, give a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor’s lexicography governs. *Id.* The specification may also resolve ambiguous claim terms “where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone.” *Teleflex, Inc.*, 299 F.3d at 1325. But, “[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.” *Comark Commc’ns, Inc. v. Harris Corp.*, 156 F.3d

1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988)); *see also Phillips*, 415 F.3d at 1323. The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) (“As in the case of the specification, a patent applicant may define a term in prosecuting a patent.”).

Although extrinsic evidence can be useful, it is “less significant than the intrinsic record in determining the legally operative meaning of claim language.” *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc.*, 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id.* at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert’s conclusory, unsupported assertions as to a term’s definition are entirely unhelpful to a court. *Id.* Generally, extrinsic evidence is “less reliable than the patent and its prosecution history in determining how to read claim terms.” *Id.*

## DISPUTED TERMS

### 1. “an asymmetric key pair” (’399 Patent Claims 1, 10, 29, 37)

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
one or more asymmetric key pairs, which are two related keys, a public key and a private key that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification	a matched pair of complementary cryptographic keys, wherein data encrypted by one of the keys can only be decrypted by the other key

The term before the Court is “an asymmetric key pair.” The dispute presented by the parties for resolution, however, encompasses the entire phrase “an asymmetric key pair having a public key and a private key” and the relationship of the public and private keys to the asymmetric key pair. PET contends that the claim phrase may encompass a plurality of key pairs such that the claimed public key and the claimed private key do not have to come from the same asymmetric key pair. In contrast, Defendants contend that the claimed public key and claimed private key come from the same asymmetric key pair. As to the meaning of any one key pair, the parties are generally in agreement.<sup>1</sup> Thus, the dispute presented squarely to the Court is whether the claimed public key and claimed private key emanate from the same key pair.

### Positions of the Parties

PET contends that the Federal Circuit has repeatedly emphasized the rule that an indefinite article “a” or “an” in patent parlance carries the meaning of “one or more.” (Dkt. No. 87 at 1–2 (citing *KJC Corp. v. Kinetic Concepts, Inc.*, 223 F.3d 1351, 1356 (Fed. Cir. 2000); *Baldwin Graphic Systems, Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1342–43 (Fed. Cir. 2008)).) To limit “a” or “an” to “one,” PET contends that the patentee must “evinced a clear intent.” (Dkt. No.

---

<sup>1</sup> Generally, the parties agree that for any particular asymmetric key pair, the public key and the private key are matched keys that perform complementary functions such as encryption and decryption or signature generation and signature verification. (Dkt. No. 78 at 7–8; Dkt. No. 86 at 4–6.)

87 at 2 (quoting *01 Communique Lab., Inc. v. LogMeIn, Inc.*, 687 F.3d 1292, 1297 (Fed. Cir. 2012)).) PET contends that subsequent use of definite articles “the” or “said” in a claim refer back to the same claim term and does not change the general plural rule, but simply re-invokes that non-singular meaning. PET contends that here, there is no claim language that requires that the public key and the private key “correspond to” a single asymmetric key pair.

PET contends that its construction comes straight from a definition of “asymmetric key pair” from a glossary document found in a 2016 webpage which PET alleges cites to a federal government computer security publication (FIPS) definition for “asymmetric key pair.”<sup>2</sup> (Dkt. No. 78 at 7.)

PET also contends that the ’399 Patent specification specifically teaches that, in addition to encryption, private keys can be used to sign code, as well as encrypt data: “[t]he second use is digital signatures where the public key is used to verify the digital signature while the private key is used to create the signature.” ’399 Patent 1:59–62. PET contends that Defendants’ construction excludes this second use. PET contends that claim 10 explicitly calls out the second usage: “a manifest of the program signed by an asymmetric private key of the predetermined asymmetric key pair.” PET further contends that the specification teaches that multiple keys can be generated for the module: “[t]he key module contains a plurality of keys.” ’399 Patent 7:47. At the hearing, PET emphasized that the process of Figure 4 describes the use of two separate asymmetric key pairs and a symmetric key pair. PET further contends that the key pair of step 100 of Figure 4 utilizes a private key for signature purposes. With regard to the “including” term

---

<sup>2</sup> PET cites the Glossary of Key Information Security Terms, [http://infohost.nmt.edu/~sfs/Regs/NISTIR-7298\\_Glossary\\_Key\\_Infor\\_Security\\_Terms.pdf](http://infohost.nmt.edu/~sfs/Regs/NISTIR-7298_Glossary_Key_Infor_Security_Terms.pdf) (last visited May 15, 2016) (citing FIPS 201). (Dkt. No. 78 at 7.) FIPS appears to be an acronym for National Institute of Standards (NIST) Federal Information Processing Standard (“FIPS”). PET neither provides FIPS 201 nor a date for FIPS 201. In PET’s Reply, PET provides portions of a 2006 version of the Glossary as Exhibit E but that exhibit does not reference “asymmetric key.” (Dkt. No. 87 Ex. E.)

discussed below in which the same issue was raised, PET also noted that another embodiment of the '399 Patent describes the use of a private key for signature purposes. '399 Patent 10:30–39.

Defendants contend that whether a claim term preceded by an indefinite article in a “comprising” claim is limited to the singular depends on the context of the claim language in light of the specification. (Dkt. No. 86 at 4 (citing *Enfish LLC v. Microsoft Corp.*, 822 F.3d 1327, 1341–42 (Fed. Cir. 2016); *In re Varma*, 816 F.3d 1352, 1362–63 (Fed. Cir. 2016); *Convolve, Inc. v. Compaq Computer Corp.*, 812 F.3d 1313, 1321 (Fed. Cir. 2016)).) Defendants contend that, here, the context of the claim language at issue is consistent only with a singular construction. Specifically, Defendants note that each asserted claim requires “an asymmetric key pair” having “a private key and a public key” and later the claims require that data be encrypted with “*the* generated public key” and a module includes “*the* generated private key.” Defendants contend that the context of the claim is clear in that the keys, being preceded by the definite article “the,” must be the public and private keys of the asymmetric key pair that were previously generated. (*Id.*)

Defendants contend that this conforms to the specification, which shows that there is a single asymmetric key pair having the generated public key and the generated private key. '399 Patent 8:20–28, Fig. 4A. Defendants contend that the specification teaches using the generated private key of the pair to decrypt the data. (*Id.* at 5 (citing '399 Patent 3:17–20, 7:55–58, 10:4–7, 8:61–63).) Defendants contend that, as taught in the specification, it is only possible for the included generated private key to decrypt the encrypted predetermined data if the public key and private key are part of the same asymmetric key pair. Defendants contend that PET’s inclusion of “complementary operations” in PET’s construction is consistent with this requirement.

Defendants contend that PET's citation to the specification's description that "[t]he key module contains a plurality of keys" does not support PET's construction because the specification is clear that the other keys are in addition to the claimed private key. '399 Patent, 8:20–28, 7:46–57. Defendants contend that inclusion of other symmetric and asymmetric keys does not change the conclusion that the private key and public key recited in the claims must form a single pair. (*Id.* at 6 (citing *Enfish*, 822 F.3d at 1341–42, n.4 (“To be clear, we do not hold that the claims are directed exclusively to a database with a single, self-referential table. Rather, the claims recite a single, self-referential table, regardless of any other tables that may be present in the same database.”))).)

Defendants also object to PET's inclusion of the language “such as . . . signature generation and signature verification.” Defendants contend that PET's sole support for this inclusion is extrinsic evidence published nearly nine years after the '399 Patent was filed. (*Id.*) Defendants contend that the relationship between the public key and the private key is defined by the fact that data encrypted by one of the keys can only be decrypted by the other key. Defendants contend that the encryption/decryption relationship can be used for various purposes, including signature generation and verification, but that does not further define an asymmetric key pair itself. Defendants agree that asymmetric key pairs can be used in the context of a digital signature, but contend that the inclusion of “use” by PET only causes confusion in the context of these claims. Specifically, Defendants note that each claim includes limitations related to encrypting data with the generated public key and all claims recite “including” the private key. Defendants state that the only reference to a digital signature in an asserted claim does not relate to the claim term “an asymmetric key pair” but, instead relates to another key pair: “a

predetermined asymmetric key pair associated with a manifest of the program.” ’399 Patent at claim 10.

### Analysis

PET seeks to have the claim interpreted such that the claimed “key pair” may be multiple key pairs such that the claimed “public key” and “private key” do not have to be a part of the same asymmetric key pair.<sup>3</sup> PET is correct that generally “a” or “an” carries the meaning of “one or more.” *KJC Corp.*, 223 F.3d at 1356. Though PET contends there is no claim language that requires the public key and the private key to correspond to the same key pair, the Court disagrees.

The context of the claim and specification must be considered. *Enfish*, 822 F.3d at 1341–42; *In re Varma*, 816 F.3d at 1362–63; *Convolve, Inc.*, 812 F.3d at 1321. For example, in *Enfish*, the claim required “a logical table.” The Federal Circuit noted that the remainder of the claim “describes rows and columns without providing any suggestion that a second table has been introduced. The specification makes clear that the invention is directed to the arrangement of a single, logical table, particularly, a row defining a column in that table.” *Enfish*, 822 F.3d at 1341–42. Similarly, in *In re Varma*, the claim limitation at issue was “a statistical analysis request corresponding to two or more selected investments.” The Federal Circuit noted that the claim language “on its face” indicates that “[a] single request must correspond to at least two investments.” *In re Varma*, 816 F.3d at 1362. The Federal Circuit noted that the use of “a” cannot “serve to negate what is required by the claim language.” *Id.* at 1363. The Federal Circuit

---

<sup>3</sup> As an initial matter, the Court notes that for more than half of the terms at issue, PET relies on 2016 online dictionaries, glossaries, etc. for supporting a plain and ordinary meaning. When pressed at the hearing as to what is the relevance of sources that are more than eighteen years after the priority date, PET acknowledged that evidence from at or before the priority date would be more relevant. PET stated, though, that its evidence shows that the plain and ordinary meaning has been long in existence. (Dkt. No. 98 at 26–27, 40–41.) The Court’s analysis provided herein for this term and the remaining terms does not rely on PET’s extrinsic evidence online sources which are dated well after both the filing and issue dates of the Asserted Patents. See *Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1299–1300 (Fed. Cir. 2003).

analogized that for one “to have ‘a dog that rolls over and fetches sticks’ it does not suffice that he have two dogs, each able to perform just one of the tasks.” *Id.* In *Convolve*, a claim preamble called out a “user interface for . . . working with a processor” and the claim body recited “means for causing the processor to . . . .” *Convolve*, 812 F.3d at 1321. The Federal Circuit found that the use of “the processor” in the body of the claim indicated that the user interface was working with the same processor to perform the recited steps. *Id.*

The Court finds that the claim language and specification here fall under the rationale of *Enfish*, *In re Varma*, and *Convolve, Inc.* Here, the claim language references “pair,” language that inherently implies that the two keys are from the same pair, just as “table” in *Enfish* implies the use of rows and columns in one table. Further, the claim language describes the pair as “having” both keys. Such language is similar to the “corresponding” language of *In re Varma*. Further, the language of ’399 Patent claim 37 recites the private key of “an asymmetric key pair” and then “a public key of the asymmetric key pair.” Such language is similar in regards to use of “the” as in *Convolve*. Taken together, the claim language teaches that the claimed public key and the claimed private key are keys of the same key pair, not keys of different key pairs. Claims 1 and 29 explicitly call out a “key pair” and then states that the “pair” has “a public key and a private key.” Similarly, ’399 Patent claim 37 recites “a private key of an asymmetric key pair and . . . a public key of the asymmetric key pair.” The use of “pair” identifies the existence of two keys, and the claims themselves state that the pair has a public key and a private key. Further, a key pair having a public key and a private key, in context of the specification, does not mean two key pairs, one providing the public key and one providing the private key. ’399 Patent 1:50–62, 7:46–58, 8:20–28.

PET makes much of the fact that additional key pairs may be utilized for other purposes. The Court's construction provided herein does not prevent the use of additional key pairs. Similar to the guidance in *Enfish*, the Court's holding does not mean that additional key pairs cannot exist, but merely that the claimed public and private keys of the pair ("key pair having a public key and a private key") are part of the same pair. For example, claim 1 recites "generating an asymmetric key having a public key and a private key." Dependent claim 10 then adds another key pair: "an asymmetric public key of a predetermined asymmetric key pair" and "an asymmetric private key of the predetermined asymmetric key pair." The predetermined keys are used for signature purposes. This other key pair is consistent with the Court's construction in that the "predetermined" keys are from the same "predetermined" key pair.

Further, the Court's construction does not exclude any embodiments. The claims include, for example, (1) generating a key pair having a public key and a private key, (2) using the public key for encrypting predetermined data, and (3) including a private key with the encrypted predetermined data. The only teaching of this in the specification has the public key and the private key coming from the same key pair. The examples PET cites to relate to other key pairs that do not meet the surrounding claim language of the key pair in question.

As described in the specification, public and private keys of the same key pair perform matching complimentary operations. Both parties acknowledge this. (Dkt. No. 78 at 7-8; Dkt. No. 86 at 4-6.) Thus, both parties include the concept of matched complementary keys in their constructions. As to the stated uses of the key pairs, the specification provides that the key pairs may have two uses: encryption and digital signature uses. '399 Patent 1:50-62. The claims themselves describe the function of a particular key pair and the corresponding public and

private keys. '399 Patent claims 1, 29, 37. As construed herein, and agreed to by both parties, the keys of a given key pair are “complementary.”

For claims 1, 10, and 29, the Court construes “an asymmetric key pair having a public key and a private key” to mean “one or more asymmetric key pairs, one of the asymmetric key pairs having the claimed public key and claimed private key, the asymmetric keys of an asymmetric key pair being complementary by performing complementary functions, such as encrypting and decrypting data or creating and verifying signatures.”

For claim 37, the Court construes “a private key of an asymmetric key pair and data encrypted by a public key of the asymmetric key pair,” to mean “a private key of an asymmetric key pair and data encrypted by a public key of the same asymmetric key pair such that the claimed public key and the claimed private key come from the same asymmetric key pair, an asymmetric key pair being complementary by performing complementary functions, such as encrypting and decrypting data or creating and verifying signatures, the use of additional asymmetric key pairs is not excluded.”

**2. “predetermined data” ('399 Patent claims 1, 29)**

<b>PET's Proposed Construction</b>	<b>Defendants' Proposed Construction</b>
data determined in advance of the encryption	data that has been selected for secure distribution before [generating an asymmetric key pair (claim 1) / executing the programming instructions (claim 29)]

The parties dispute whether or not the data has to be determined prior to both generating the key pair and executing the programming instructions.

## **Positions of the Parties**

PET states that the relevant claims require “predetermined data” to be encrypted by a public key, and the ordinary meaning of this term requires only that the data be determined in advance of the encryption. PET contends that there is no claim language that requires that the data must be determined prior to the generation of asymmetric key pairs or prior to the execution of any code. (Dkt. No. 78 at 9.) PET further contends that though it does not believe the steps of the claim have a particular order, if an order is required, it normally is the order as claimed. Here, PET notes that claim 1 recites the generating step before the “encrypting the predetermined data step.” PET contends that all that is required by the claims is that the data be determined in advance of the encryption with a public key. As to claim 29, PET contends that since the claim is an apparatus claim, including Defendants’ order requirement is even more inapt. PET further contends that claim 29 includes no limitation at all that the predetermined data be “distributed” in any fashion, further counseling against Defendants’ construction.

Defendants contend that PET’s construction effectively reads out the word “predetermined.” Defendants contend their construction is required in order to give meaning to “predetermined.” Defendants contend that PET bases its arguments on its erroneous contention that the steps of the ’399 Patent do not require any particular order. The Defendants contend that the steps must be performed in the claimed order, simply because each step requires the results from the last. (Dkt. No. 86 at 7.) Defendants contend that the second step of encrypting predetermined data with the generated public key cannot occur before an asymmetric key pair having a public key and a private key is generated in the first step. Defendants contend that although claim 29 is an apparatus claim, for the same reasons, the programming instructions must be performed in this same order.

Defendants further contend that the only process disclosed in the specification teaches that the data is determined before the key pair is generated. '399 Patent 7:59–8:31, Fig. 4A. In particular, Defendants contend that, in Figure 4A, the asymmetrical key pair is generated in step 108 after the data is stored at step 102. (Dkt. No. 86 at 8.) Defendants contend that nowhere in the specification is it suggested that the data can be determined *after* the asymmetric key pair is generated.

### Analysis

Ordinarily, a method claim is not construed to require that its constituent steps be performed in a particular order unless the claim recites an order. *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1342 (Fed. Cir. 2001). A method claim that does not recite an order may nonetheless be construed to require that the claim's steps be performed in a particular order if (1) the claim language, "as a matter of logic or grammar" requires that the steps be performed in a particular order, or (2) the specification "directly or implicitly requires such a narrow construction." *See Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369–70 (Fed. Cir. 2003).

Here, Defendants not only seek an order to the claim, but they seek an order that does not match the order of the steps as recited in the claim. In particular, Defendants seek that the "predetermined data" of the second limitation is determined before the generation of the first limitation. But, the claim merely first recites the "generating an asymmetric key pair" step and the second step is "encrypting the predetermined data with the generated public key." To the extent an order is required, the order in the claim would contradict what Defendants seek. Defendants may be correct that inherently the generation step would be performed before the encrypting step because the encrypting is done with the generated public key. However, this

speaks nothing to the order of when the data is determined with reference to the timing of the generation of the key pair. At the hearing, Defendants emphasized that the Defendants’ construction matches the order of the flow chart in Figure 4A. However, as mentioned above, mere teaching of an embodiment in the specification is not enough to require the embodiment’s order in the claims. *See Altiris, Inc.*, 318 F.3d 1370–71 (the specification must require the order).

**The Court construes “predetermined data” to mean “data determined in advance of the encryption.”**

**3. “executable tamper resistant key module”/“executable tamper resistant code module”/“tamper resistant key module” (’399 Patent claims 1, 9, 29, 37)**

**“tamper resistant module”/“module” (’550 Patent claims 14, 15, 16)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
<p>As to ’399 Patent terms:  software that is designed to work with other software, is resistant to modification and that includes a plurality of keys used for secure communication</p> <p>As to ’550 Patent terms:  software that is designed to work with other software and that is resistant to modification</p>	<p>a self-contained unit of software that is capable of being communicated independently from the selected program, comprising instructions to check the integrity of the selected program and itself, that is compiled to be resistant to observation and modification such that attempts to decipher what the software is doing, or modifications made to the software, will result in the software being unable to execute</p>

Defendants seek to incorporate a number of terms from the specification. The key disputes include (1) the meaning of “tamper resistant,” (2) whether the module has to be communicated independent from a “selected program,” and (3) whether any attempts to evade the security will result in the software “being unable to execute.”

## **Positions of the Parties**

PET notes that the parties all agree that the terms should be construed consistently and all terms relate to software. PET contends that “tamper resistant” also has a well-known ordinary meaning relating to “resistant to modification.” (Dkt. No. 78 at 10 (citing 2016 general dictionaries).) PET contends that the claim language has an ordinary meaning and that unless the specification provides disclaimer or redefinition, the construction should stay true to the claim language. (*Id.*)

PET contends that the patent discloses many methods for making the module tamper resistant. (Dkt. No. 78 at 11 (citing ’399 Patent at 6:7–16 (“Detailed methods for creating the tamper resistant module . . . are disclosed in pending US patent applications . . . and are incorporated herein by reference.”))).) PET contends there is no reason to read limitations from the specification embodiment into the plain claim language.

PET also objects to Defendants’ use of “unable to execute.” PET contends this limitation is directly contrary to the specification of the patent. ’399 Patent 5:52–55 (“[Tamper resistant software] can be trusted, within certain bounds, to operate as intended even in the presence of a malicious attack.”). PET thus contends that “tamper resistance” does not require inoperability in the presence of a detected alteration or modification. PET contends that Defendants’ construction would improperly read out of the claim this preferred embodiment.

PET contends that the term “key module” is used broadly in the specification: “[t]he key module contains a plurality of keys.” ’399 Patent 7:46. Further, PET states that during the prosecution of the ’399 Patent, it was specifically argued that the meaning of the word “module,” in the specification and the claims, “is consistent with common usage of the word by those

skilled in the art” (Dkt. No. 78 Ex. C at FH0084.)<sup>4</sup> PET contends that the common usage is that it is software designed to work with other software. (Dkt. No. 78 at 11 (citing ’399 Patent 7:40–41 (“The key module is forwarded over communications network 34 to client 32. It is a ‘plug-in’ to executable 44 of trusted player 42.”))).)

PET further objects to Defendants’ proposed construction of a module as a “self-contained unit of software” as having no actual technical meaning in the context of software. PET also contends that the added limitation that the module must be “capable of being communicated independently from the selected program” improperly incorporates limitations not provided by the claims. PET contends that the claim language explains which keys are necessary to be included in the module and, thus, PET’s proposed construction of the term is clear.

As to the ’550 Patent, PET contends that there is no discussion of particular cryptographic keys in the module, but rather an embedded “secret” that is recovered when the module is executed by a processor. PET contends that otherwise the arguments set forth for the ’399 Patent apply equally. PET contends that the ’550 Patent also discloses that tamper resistant software modules do not necessarily stop executing because tampering is detected. ’550 Patent at 2:49–52 (“Tamper resistant software . . . can be trusted, within certain bounds, to operate as intended even in the presence of a malicious attack.”).

Defendants contend their construction is fully supported by the specification and that PET’s proposal is so short it provides no aid for actually understanding these terms. Defendants contend that PET ignores the specifications and improperly relies on extrinsic, dictionary definitions for some of the individual constituent terms and links those definitions to arrive at a purported construction for the entire claim term. (Dkt. No. 86 at 10.)

---

<sup>4</sup> File history citations for the ’399 Patent are made to Dkt. No. 78 Ex. C and for the ’550 Patent are made to Dkt. No. 78 Ex. D.

Defendants contend that defining “tamper” or “tamper resistant” by relying on non-technical, present-day online dictionaries fails to take into account what an executable tamper resistant key module is in the context of the Asserted Patents. Defendants contend that the concept of a tamper resistant module was a term coined prior to issuance of the Asserted Patents, as the patentees acknowledged by expressly incorporating by reference other patents’ disclosures of the term. ’399 Patent 6:7–16. Defendants contend that their construction defines these terms based on their functionality as used by the patents. (*Id.* (citing *Phillips*, 415 F.3d at 1315 (“The best source for understanding a technical term is the specification from which it arose, informed, as needed, by the prosecution history.”))).)

Defendants contend that because the terms are “coined” terms, reference to the specification is required. Defendants further state that each element of their construction is found in and supported by the specification of the ’399 Patent. Defendants contend that PET provides no basis whatsoever for its omission of the fact that these claim terms are about a “module,” which, as understood in the field, is a piece of self-contained software. (*Id.* at 11 (citing technical dictionary).) Defendants assert that PET’s “software that works with other software” is meaningless as all software is designed to work with other software. Defendants contend that the specification further establishes that the module is self-contained by the fact that it is both independently communicated and acts as a plug-in. ’399 Patent 7:40–41.

Defendants further assert that in addition to the module being a separate component, the patents disclose that the tamper resistant module is capable of being communicated separately from the selected program:

The storage device reader 16 interacts with a key module 18, which is downloaded from a communications network or otherwise accessed by the storage device reader. ’399 Patent 4:45–48.

Preferably, key module 18 is provided dynamically 60 by a content provider from a remote system over a communications network such as the Internet. *Id.* at 4:59–61.

The key module is forwarded over communications network 34 to client 32. *Id.* at 7:41–42.

The tamper resistant module is a small piece of executable software that can be easily communicated from one process to another and executed by a receiving system. *Id.* at 2:34–37.

The tamper resistant module can be sent to Process B before the challenge, after the challenge, or simultaneously with the challenge. *Id.* at 3:36–38.

(Dkt. No. 86 at 11–12.) Defendants contend that the '550 Patent file history likewise supports this communication aspect by asserting the tamper resistant module is a “software agent sent to the remote process.” (Dkt. No. 78 Ex. D at FH0196–97.)

Defendants further state that the tamper resistant module is comprised of instructions to check the integrity of the selected program and itself: “[t]he tamper resistant key module is then executed on the remote system to check the integrity and authenticity of the program and the integrity of the tamper resistant key module itself.” '399 Patent 3:14–17. Defendants state that the '399 Patent further teaches that the key module “ensures that the party requesting” decryption “is authentic and its integrity is verified.” '399 Patent 4:46–59.

As to the '550 Patent, Defendants contend that one process seeks to establish the authenticity and integrity of another process. (Dkt. No. 86 at 12 (citing '550 Patent, 2:26–28).) Defendants state that authenticity means the process is actually what it purports to be and integrity confirms the process has not been altered or hacked. (*Id.* (citing '550 Patent, 2:28–31).) Defendants further state that the tamper resistant module “is capable of verifying integrity of” the exemplary Process B. '550 Patent 2:33–34.

Defendants contend that “tamper resistant” in the module terms means the module is compiled to be resistant to observation and modification such that interference with the module will yield it unable to execute. Defendants contend that the specification teaches this: “[t]amper resistant software is software which is resistant to observation and modification” (’399 Patent, 5:52–53) and “[t]his self-decrypting software will only execute properly if no part of the image has been altered from the time it was compiled by the tamper resistant compiler” (’399 Patent 5:59–62).

Defendants contend that contrary to PET’s assertions, these characteristics of the terms are not an improper importation of one embodiment. Rather, Defendants contend that the patents carry them, in the context of these modules, from one embodiment to another for all applications of the claimed inventions. Defendants further contend that PET cites to no actual embodiment to the contrary.

Defendants contend that PET provides no intrinsic evidence to limit “tamper resistant” to “resistant to modification” despite the specifications’ clear explanation that tamper resistant software “is software which is resistant *to observation* and modification.” ’399 Patent 5:52–53 (emphasis added). Defendants assert that even as to just “modification,” “resistant to” by itself is also insufficient to properly define this term. Defendants contend that not only do the patentees distinguish between digital signatures and tamper resistance, but it is the tamper resistant module that is using the digital signature of the selected program to determine if the program has been tampered with. (Dkt. No. 86 at 13 (citing ’399 Patent 4:17–20; 5:11–14 (“The present invention is designed to prevent or obstruct all of these attacks by the combined methods of tamper resistance, authentication, and verification of integrity.”); 8:39–46 (“The IVK in the key module

verifies that the signature of the trusted player corresponds to the manifest.”)).) Defendants contend that for this reason alone, PET’s proposed construction should be rejected.

In reply, PET contends that Defendants propose reading “self-contained” into the claim language based on a definition from a computer encyclopedia and that “self-contained” is found nowhere in the intrinsic evidence. PET states that Defendants do not explain what “self-contained” is supposed to mean, and thus, a term not found in the specification will require further explanation. (Dkt. No. 87 at 3.) PET contends that the Defendants admit that the specification of the ’399 Patent is replete with references to the “key module” working in connection with other software, and thus there is no reason to define it narrowly.

As to the ’550 Patent, PET contends that the Defendants ignore that the two patents are separate patents. PET contends that because there is no intrinsic evidence supporting Defendants’ restrictive reading in the ’550 Patent, Defendants do not address that patent separately. (Dkt. No. 87 at 9.) PET contends that the lack of any evidence supporting a restrictive reading in the ’550 Patent is strong evidence that the limitations should not be read into either patent. (*Id.*)

### **Analysis**

Defendants seek to incorporate a number of specific embodiments from the specifications into the claim terms. Moreover, Defendants seek to include specific embodiments from one asserted patent into the other patent. Defendants even include certain ’399 Patent claim limitations such as “the selected program” into the ’550 Patent construction, even though “selected program” is wholly absent from the claims and specification of the ’550 Patent. Defendants’ constructions are not proper in light of the intrinsic evidence, and further, fail to acknowledge that the two patents are independent patents.

The '399 Patent specification provides a broad general meaning as to what is meant by “tamper resistant:” “[t]amper resistant software is software which is resistant to observation and modification.” '399 Patent 5:52–53. The '399 Patent specification further explicitly states that detailed methods for creating tamper resistant modules were already known and incorporates by reference U.S. Patent No. 5,892,899 (the “'899 Patent”) entitled “Tamper Resistant Methods and Apparatus.” '399 Patent 6:7–16. The '899 Patent clearly states that there is a wide range of techniques for rendering a software program tamper resistant: “distributing” the program in “time and space” into “a number of subprograms” that are executed “over a period of time” ('899 Patent 1:35–42, Fig. 1); “obfuscating the program” (*Id.* at 1:46–48, 2:5–6, Fig. 4); “distributing the secret private key in time as well as space” (*Id.* at 1:65–66, 2:5–6); “by providing a system integrity verification program having tamper resistant integrity verification kernels, that jointly deploy an interlocking trust mechanism . . . .” (*Id.* at 2:14–17, Fig. 16.) Defendants seek to incorporate specific embodiments of tamper resistance disclosed in the '399 Patent. However, as noted, the specification describes the term in a broader context. Absent more, specific embodiments will not be read into the claim limitations. *See Phillips*, 415 F.3d at 1323. The more general broad statements of the '399 Patent provide the proper context of “tamper resistant.”

As to Defendants’ requirement relating to the result of an attack on a tamper resistant module (“attempts to decipher or modify the software will result in the software being unable to execute”), again the '399 Patent provides explicit teaching otherwise: “[t]amper resistant software is software which is resistant to observation and modification. It can be trusted, within certain bounds, to operate as intended even in the presence of a malicious attack.” '399 Patent 5:52–55. Defendants’ construction would render the software inoperable, even if an attack was

repelled. Finally, Defendants seek to include a “self-contained” limitation; again, the intrinsic evidence indicates otherwise as the ’899 Patent, explicitly incorporated by reference into the ’399 Patent, teaches that the tamper resistant modules can be partitioned in time and space into sub-programs. ’899 Patent 1:35–42, 3:48–4:20; Fig. 1.

The embodiments Defendants seek to incorporate primarily find support from the ’399 Patent. But as to those ’550 Patent limitations to which Defendants cite, just as with the ’399 Patent, the ’550 Patent intrinsic evidence provides a broader description countering Defendants’ limitations. Just as in the ’399 Patent, the ’550 Patent states: “[t]amper resistant software is software which is resistant to observation and modification. It can be trusted, within certain bounds, to operate as intended even in the presence of a malicious attack.” ’550 Patent 2:49–53. Further, the ’550 Patent likewise incorporates by reference the patent application that became the ’899 Patent (application Ser. No. 08/662,679). ’550 Patent 3:21–29. For the reasons recited above with regard to the ’399 Patent, the inclusion of Defendants’ limitations in the ’550 Patent claims is improper.

The Court also notes that PET’s construction does not totally conform to the specification and claims. First, it is clear that tamper resistance, as described in both specifications, provides resistance to observation and modification. ’399 Patent 5:52–53; ’550 Patent 2:49–53. Second, PET recites “a plurality of keys” for the ’399 Patent terms. However, certain claims only reference inclusion of the private key in the module. ’399 Patent 3:5–20, Claim 1.

Finally, it is noted that the terms of the ’399 Patent and corresponding asserted claims include “key” whereas the ’550 Patent terms to be construed and the corresponding asserted claims do not reference “key.”<sup>5</sup>

---

<sup>5</sup> ’399 Patent claim 9 which uses the term “the tamper resistant code module” depends from claim 1, which references “an executable tamper resistant key module.”

The Court construes “executable tamper resistant key module” / “executable tamper resistant code module” / “tamper resistant key module” (’399 Patent) to mean “software that is designed to work with other software, that is resistant to observation and modification, and that includes a key for secure communication.”

The Court construes “tamper resistant module” / “module” (’550 Patent) to mean “software that is designed to work with other software and that is resistant to observation and modification.”

**4. “selected program”/“program” (’399 Patent claims 1, 29, 37)**

PET’s Proposed Construction	Defendants’ Proposed Construction
ordinary meaning; a “program” is a series of computer instructions; selected program means a “particular series of computer instructions”	an application program

The parties dispute whether or not a prosecution statement limits “program” to “application program.”

**Positions of the Parties**

PET contends that the term has a clear ordinary meaning, and that PET’s construction is supported by both non-technical and technical dictionaries. (Dkt. No. 78 at 13 (citing 2016 Internet sources).) PET objects to Defendants’ construction for limiting the plain meaning of “program” to a particular type of program. PET contends that the specification indicates distinctions between the usage of “application program” and simply “program:” “[c]onsider the situation where an application program running on a user’s PC accesses encrypted digital content on a storage medium” (’399 Patent 2:35–37) versus “[a]n embodiment of the present invention is a method of securely distributing data to a program on a remote system” (*Id.* at 3:5–6). PET contends that Defendants are attempting to read an embodiment into the claims.

Defendants contend that during prosecution, the Examiner objected that “throughout claims 1–33, the uses of the word ‘process’ are indefinite and unclear.” (Dkt. No. 78 Ex. C at FH0072.) In response, the patentees changed “process” to “program” in the claims, and asserted that “[s]upport for this change may be found at page 7, lines 10–15, where the entity for receiving the distribution of the data is discussed as a program.” (*Id.* at FH0084.) Defendants quote passage of the specification:

An embodiment of the present invention includes a method of securely distributing a private key to a user’s application program (also called a “trusted player” such as a digital versatile disk (DVD) player, compact disk read only memory (CD-ROM) player, or floppy disk device driver, and the like) with conditional access based on verification of the trusted player’s integrity and authenticity.

’399 Patent, 3:54–59. Defendants further contend that in prosecution arguments responding to the Examiner’s indefiniteness objection to the word “player,” the patentees asserted that “at page 7, lines 10–15, the entity receiving the private key for use in decrypting an encrypted digital object is an application program. When the application program has certain functional capabilities, it may also be called a player.” (Dkt. No. 78 Ex. C at FH0085.) Defendants contend that the prosecution thus indicates that the term “selected program” means “an application program.”

### **Analysis**

Because the file history “represents an ongoing negotiation between the PTO and the applicant, rather than the final product of that negotiation, it often lacks the clarity of the specification and thus is less useful in claim construction proceedings.” *Phillips*, 415 F.3d at 1317. Statements will constitute disclaimer of scope only if they are “clear and unmistakable statements of disavowal.” *See Cordis Corp. v. Medtronic Ave, Inc.*, 339 F.3d 1352, 1358 (Fed. Cir. 2003). An “ambiguous disavowal” will not suffice. *Schindler Elevator Corp. v. Otis*

*Elevator Co.*, 593 F.3d 1275, 1285 (Fed. Cir. 2010) (citation omitted). On balance, neither the specification nor the prosecution history contain any definitive statement or disclaimer mandating that “program” must be “application program.” See *Omega Eng. v. Raytek Corp.*, 334 F.3d 1314, 1324 (Fed. Cir. 2003) (“As a basic principle of claim interpretation, prosecution disclaimer promotes the public notice function of the intrinsic evidence and protects the public’s reliance on definitive statements made during prosecution.”). Here, the prosecution history indicates that the patentees merely pointed to a portion of the specification to indicate that the specification provided support for use of the term “program.” Merely because that portion of the specification referenced a particular program provides no indication that the general meaning of “program” was being disavowed. Defendants are utilizing the prosecution history to improperly import embodiments from the specification. Moreover, the specification repeatedly references the more general usage of just a “program.” ’399 Patent 3:5–20, 4:35–37, 5:18, 10:40–45.

Having resolved the parties’ dispute over whether “program” is limited to “application program,” the Court finds that the term needs no further construction. See *O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1362 (Fed. Cir. 2008) (“[D]istrict courts are not (and should not be) required to construe every limitation present in a patent’s asserted claims.”); *Finjan, Inc. v. Secure Computing Corp.*, 626 F.3d 1197, 1207 (Fed. Cir. 2010) (“Unlike *O2 Micro*, where the court failed to resolve the parties’ quarrel, the district court rejected Defendants’ construction.”).

**The Court finds that “program” has its plain and ordinary meaning and that no further construction is necessary.**

**5. “including the generated private key and the encrypted predetermined data” (’399 Patent Claims 1, 29)**

**“including a private key of an asymmetric key pair and data encrypted by a public key of the asymmetric key pair” (’399 Patent Claim 37)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
including one or more of the private keys of the one or more asymmetric key pairs and encrypted predetermined data (Claims 1, 29)	has compiled within itself the encrypted predetermined data, and the private key that can be used to decrypt the encrypted predetermined data (Claims 1, 29)
including one of the private keys of one or more asymmetric key pairs and data encrypted by one of the public keys of one or more asymmetric key pairs (Claim 37)	has compiled within itself data encrypted by a public key, and the private key that can be used to decrypt said data (Claim 37)

The parties present several issues for resolution. First, the parties raise the “one or more” issue discussed above with regard to “asymmetric key pair.” Included with that first dispute is Defendants’ contention that the private key must be used to decrypt the predetermined data. Second, the parties dispute whether “including” must reference “compiled.” Third, Defendants object to PET’s omission of the use of “the” with reference to the encrypted predetermined data.

**Positions of the Parties**

PET contends that the Federal Circuit has held that “including” is synonymous with “comprising,” and should be given a broad construction. *Hewlett-Packard Co. v. Repeat-O-Type Stencil Mfg. Corp.*, 123 F.3d 1445, 1451 (Fed. Cir. 1997). PET contends that nothing in the ’399 Patent’s specification or prosecution history explicitly defines the term “including” to mean “compiled within,” or disavows its normal broad scope.

As to the private key, PET contends that the claims require nothing more than including a private key. PET contends that multiple uses for private keys are described in the specification, and there is no reason to read a particular embodiment or function into this claim language. (Dkt.

No. 87 at 5–6.) PET contends that including a private key to sign the key module as part of the building step is specifically disclosed as preferred embodiments in the specification. (Dkt. No. 78 at 15 (citing '399 Patent 7:46–51 (“The key module contains a plurality of keys. It contains an asymmetric public key for verifying the digital signature of the manifest. The digital signature was created using an asymmetric private key by the manufacturer of the trusted player.”); *Id.* at 10:35–39 (“The signature verification engine function is performed on the digest of the specified object using the generated asymmetric private key to generate a signature, which can be validated by the trusted player or other application on the client.”))).) PET contends that the use of a private key of a generated asymmetric key pair to sign the software is specifically described by dependent claims such as: “[t]he method of claim 1, wherein building the executable tamper resistant code module comprises generating an integrity verification kernel” (claim 9) and “generating an integrity verification kernel” may comprise “a manifest of the program signed by an asymmetric private key of the predetermined asymmetric key pair” (claim 10). PET contends that while not required by claim 1, certainly using a private key (of one of the generated asymmetric key pairs) to sign a manifest of the program is within the scope of the claims.

PET contends that there are other dependent claims that cover using a private key to decrypt the encrypted predetermined data. (Dkt. No. 78 at 16 (citing dependent claim 4).) PET contends there is no reason to read the decryption limitation into claim 1, much less exclude the embodiments found in claim 10. PET contends that the doctrine of claim differentiation further supports its position.

With respect to “including,” Defendants contend that both sides agree that: (1) the tamper resistant key module is executable software; and (2) the private key (and the encrypted data) is included “in” the tamper resistant key module. Defendants contend that their construction

correctly explains that to be included in the software means to be “compiled within,” consistent with how the patentees describe building the tamper resistant key module:

A key compiler is a program that takes an asymmetric key pair, which is represented as data, and turns it into a piece of executing code such as the key module 18. In this way, the entire key is never assembled at one place in a program at one point in time. Instead, pieces of the key are revealed as they are needed. Thus, the key is distributed in program space. This makes it hard for an attacker to find and change the key.

’399 Patent 5:44–51. Defendants also cite to the passage: “[i]n parallel with IVK generation function processing, the generated asymmetric private key 202 for use in decrypting the encrypted symmetric keys is processed by another instance of key compiler 208.” *Id.* at 9:55–58, Fig. 5; Fig. 4A.

Defendants contend that PET’s argument that including a private key to sign the key module as part of the building step is specifically disclosed as a preferred embodiment in the specification is incorrect. Defendants contend that the quoted portion of the specification, relied on by PET, is actually referring to the digital signature of the program (the “trusted player” in the preferred embodiment) to which the tamper resistant key module is being sent in order to check the integrity and authenticity of such program. (Dkt. No. 86 at 17.) Defendants contend that the entirety of the paragraph shows the incorrectness of PET’s position:

The key module contains a plurality of keys. It contains an asymmetric public key for verifying the digital signature of the manifest. The digital signature was created using an asymmetric private key by the manufacturer of the trusted player. To create a key module capable of verifying the manifest, key module generation function 50 needs to obtain the corresponding asymmetric public key. The key module also contains one or more symmetric keys for decrypting the encrypted digital content. ***Finally, the key module includes an asymmetric private key for decrypting the encrypted symmetric public keys*** when the validity of the trusted player on the client is assured.

’399 Patent 7:46–57 (emphasis added). Defendants contend that the patentees are describing the keys included in the tamper resistant key module and that none of the keys described are used to

“sign” the tamper resistant key module. Defendants contend that PET’s arguments demonstrate the need for a construction, rather than relying on the ordinary meaning of the word “including.”

Defendants contend that the digital signature of the program (the “trusted player” in the preferred embodiment) to which the tamper resistant key module is being sent in order to check the integrity and authenticity of such program is described and claimed elsewhere. (Dkt. No. 86 at 17, n.12 (citing ’399 Patent 8:34–37, claims 3, 10, 11).) Defendants also contend that PET’s reliance on the embodiment described at 10:35–39 of the ’399 Patent is similarly misplaced. Defendants assert that such embodiment does not supplant the use of the asymmetric key in the tamper resistant key module but merely describes an unclaimed way of also allowing the “trusted player” to confirm the validity of the tamper resistant key module. (*Id.*) Defendants contend this is just another instance showing that the use of digital signatures (or “code signing”) is not the claimed tamper resistance.

As to the usage of the private key, Defendants contend their construction explains that the private key compiled within the tamper resistant key module is the key that is able to decrypt the encrypted predetermined data which is also compiled within the tamper resistant key module. Defendants contend that the private key is from the same asymmetric key pair (generated in the first limitation) as the public key used to encrypt the predetermined data (recited in the second limitation). Defendants contend that PET’s proposed construction for this term further evidences the weakness of PET’s proposed construction for “an asymmetric key pair.” (*Id.* at 18.)

Defendants further note that PET omits the usage of “the” with reference to the claimed “the encrypted predetermined data” in claims 1 and 29. Defendants contend the data is the prior referenced “encrypted predetermined data” and the use of “the” should be maintained in the construction. (*Id.* at 16, n.7.)

## Analysis

PET seeks to allow the particular claimed private and public keys to come from different pairs and not themselves be a “pair.” Again, PET seeks to ignore that the claim links a “pair” having a private key and a public key. For the reasons described above with reference to the “asymmetric key pair” term, the Court rejects PET’s position. PET argues that dependent claims, such as claim 10, indicate that the private key may be used for signature purposes, thus, contemplating a private key from another pair. Again, as noted above, the Court’s construction does not preclude the existence of additional key pairs. In fact, as noted, claim 10 specifically recites another key pair: “a predetermined asymmetric key pair associated with the manifest.” This conforms to the specification. ’399 Patent Figure 4A, 7:59–66. Finally, PET points to an alternative embodiment described at ’399 Patent 10:30–39. However, that embodiment does not include the claimed “encrypting predetermined data with the generated public key.” Thus, the use of the generated private key that is complementary to the generated public key is not implicated by that embodiment. ’399 Patent 10:30–39.

Similarly, as described above, the Court rejects Defendants’ addition of a decryption step to the “including” term. That issue has been resolved in the construction of “an asymmetric key pair.” The claim term in question here only references “including” the private key and the encrypted data. Defendants’ construction, in this regard, goes beyond the meaning of “including” and instead seeks to add details that are unrelated to the meaning of the “included” phrase. *See Phillips*, 415 F.3d at 1323. Again, the complementary nature of key pairs is described in the construction of “asymmetric key pair.”

As to what it means to “include” a key and data in a tamper resistant module, Defendants point to an embodiment in the specification in which a key compiler is used to turn the key into a

piece of executing code that may be included in the module. (Dkt. No. 86 at 16–17 (citing ’399 Patent 5:44–56).) The term in question, though, is much broader, merely requiring “including.” Defendants have not pointed to any disclaimer or disavowal limiting the meaning of “including.” Rather, Defendants merely point to an embodiment of the specification. However, even a single embodiment is not necessarily enough to read a limitation into the claim from the specification. *Arlington Indus., Inc. v. Bridgeport Fittings, Inc.*, 632 F.3d 1246, 1254 (Fed. Cir. 2011) (“[E]ven where a patent describes only a single embodiment, claims will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using words of expressions of manifest exclusion or restriction.”) (citation omitted). In addition, there is at least one usage in the specification referencing “including” the private key and the data without reference to the compiling limitation. ’399 Patent 3:7–14.

As to PET’s exclusion of “the” with regard to “the predetermined data” of claims 1 and 29, Defendants raise a proper objection. PET did not provide rebuttal to Defendants’ objection in the briefing or at the hearing. The claims recite: “encrypting predetermined data . . . the executable tamper resistant key module including the generated private key and the encrypted predetermined data.” ’399 Patent claims 1 and 28. What is “included” is the prior referenced encrypted predetermined data. *See NTP, Inc. v. Research in Motion, Ltd.*, 418 F.3d 1282, 1306 (Fed. Cir. 2005) (use of a definite article “the” refers to the antecedent usage of “a”). This is clear from the claim language itself and does not appear to be contested.

The claim terms are found in the context of the surrounding claim language of the “executable tamper resistant key module including . . . .” Having resolved the disputes regarding the parties’ additional limitations added to the clear concept of a module “including” the private key and the encrypted data, no further construction is necessary. *See O2 Micro Int’l Ltd. v.*

*Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1362 (Fed. Cir. 2008) (“[D]istrict courts are not (and should not be) required to construe every limitation present in a patent’s asserted claims.”); *Finjan, Inc. v. Secure Computing Corp.*, 626 F.3d 1197, 1207 (Fed. Cir. 2010) (“Unlike *O2 Micro*, where the court failed to resolve the parties’ quarrel, the district court rejected Defendants’ construction.”). The constructions of “asymmetric key pair” and the module terms, in conjunction with the Court’s rejection of limiting “including” to compiling, resolves the disputes.

The Court finds that “including the generated private key and the encrypted predetermined data” (’399 Patent Claims 1, 29) and “including a private key of an asymmetric key pair and data encrypted by a public key of the asymmetric key pair” (’399 Patent Claim 37) need no further construction.

**6. “integrity verification kernel” (’399 Patent Claims 9, 10; ’550 Patent Claim 15)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
software that can be used in conjunction with other software to determine that code has not been altered through the use of a digital signature	small code segment that is compiled in a manner to make it resistant to observation and modification, such that attempts to decipher what the software is doing, or modifications made to the software, will result in the software being unable to execute and that verifies that the image of the program corresponds to a supplied digital signature for that program

The parties contest whether “integrity verification kernel” should be construed based on general ordinary meanings of the constituent elements of the term or based upon the specification disclosures. PET objects to Defendants’ inclusion of details of disclosed embodiments.

## **Positions of the Parties**

PET contends that “integrity” has a well-known meaning in the art of encryption relating to the concept that data not being altered or modified as evidenced by 2016 Internet sources. (Dkt. No. 78 at 17 (citing Glossary of Key Information Security Terms, [http://infohost.nmt.edu/~sfs/Regs/NISTIR-7298\\_Glossary\\_Key\\_Infor\\_Security\\_Terms.pdf](http://infohost.nmt.edu/~sfs/Regs/NISTIR-7298_Glossary_Key_Infor_Security_Terms.pdf) (last visited May 15, 2016) (citing FIPS 140-2); Netlingo Dictionary, <http://www.netlingo.com/word/integrity.php>)).<sup>6</sup> PET contends this meaning is supported by the specification. (*Id.* (citing ’399 Patent at 1:27–29; 2:30–35; 3:14–17; 3:53–59; 4:8–14; 4:18–20; 4:56–59; 5:15–17).) PET also contends that “verification” also has a well-known meaning in the art related to the confirmation of the truth. (*Id.* at 17–18 (citing a variety of 2016 non-technical dictionaries).)

PET contends that although various embodiments are described in the specification for the “integrity verification kernel” (“IVK”), the specification describes an IVK as “software that verifies that a program image corresponds to the supplied digital signature.” ’399 Patent 5:18–20. PET further asserts that the specification also explains that an IVK “can be used alone . . . or it can be used in conjunction with other software. . . .” *Id.* at 5:22–24. PET contends that these aspects of the IVK are incorporated in PET’s construction. PET objects to Defendants’ construction as incorporating Defendants’ construction for the different claim term “tamper resistant” into the construction for the IVK.

Further, PET contends that the remainder of Defendants’ proposed construction reads limitations regarding specific IVKs described in the specification into the claims. PET contends there is nothing that requires that the code be unable to operate in the event of a change or “an

---

<sup>6</sup> In its reply, PET contends that the technical meaning of “integrity,” proposed by Plaintiff, was supported by a Glossary of Key Information Security Terms compiled by the federal government in 2006, (Dkt. No. 87 Ex. E), but that the definition comes from a federal government information security standard, FIPS 140-2, which was published in May of 2001. (Dkt. No. 87 at 6.)

attempt to decipher what the software is doing.” (Dkt. No. 78 at 18.) PET contends that, to the contrary, the specification teaches that “tamper resistant” software is often designed to continue to operate even in the event of a detected attack, much less when one attempts to decipher what the software is doing. (*Id.* (citing ’399 Patent 5:52–55 (“Tamper resistant software . . . can be trusted, within certain bounds, to operate as intended even in the presence of a malicious attack.”))).)

PET contends that similar arguments apply to the usage of the term in the ’550 Patent. PET points to the ’550 Patent specification as describing an IVK as “software that verifies that a program image corresponds to the supplied digital signature.” ’550 Patent 3:4–6. PET also points to the prosecution history as being consistent: “[t]he present invention also determines the integrity of the remote process, to detect if it has been tampered with, through the use of an integrity verification kernel.” (Dkt. No. 78 at 27–28 (quoting Dkt. No. 78 Ex. D at FH0197).)

Defendants contend that PET tries to define the coined claim term “integrity verification kernel” by individually defining words that make up the term. Further, Defendants contend that PET then looks to non-contemporaneous dictionary definitions for the meanings of the words “integrity” and “verification.” Defendants contend that non-contemporaneous extrinsic evidence can provide no support for PET’s construction. (Dkt. No. 86 at 19 (citing *Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1300 (Fed. Cir. 2003))).) Defendants contend that for a coined term like “integrity verification kernel,” the patent specifications are the correct first source for IVK’s proper construction. (*Id.* (citing *Honeywell Int’l Inc. v. Universal Avionics Systems Corp.*, 488 F.3d 982, 991 (Fed. Cir. 2007))).)

Defendants point to the ’550 Patent specification: “[i]ntegrity verification means that [the process] has not been altered or ‘hacked’ in any way.” ’550 Patent 2:30–31. Defendants contend

that there is no need to look to any extrinsic evidence. Defendants also assert that the patentees argued, during prosecution of the '550 Patent, that the addition of an IVK to determine the integrity of the remote process and to detect if it has been tampered with distinguished the claimed inventions over prior art. (Dkt. No. 86 at 19–20 (citing Dkt. No. 78 Ex. D at FH0197–98).)

Defendants further contend that their proposed construction is derived directly from the specifications:

An integrity verification kernel (IVK) is software that verifies that a program image corresponds to the supplied digital signature. An IVK is a small code segment that has been “armored” using methods to ensure that it is not easily tampered with. An IVK can be used alone, to ensure that its tasks are executed correctly, or it can be used in conjunction with other software to provide the assurance that the other software has executed correctly (that is, they can be used as verification engines).

'399 Patent 5:18–26.

An IVK is software that verifies that a program image corresponds to a supplied digital signature. This provides a robust mechanism for detecting changes made to executing software, where those changes might be caused by transmission errors or malicious attacks to the software. Any change to the software results in a failure in the verification process. IVKs for tamper resistant software are constructed to perform self-checks of object code, bilateral authentication of partner modules, and checks on local and remote data to verify the integrity of a software module.

'550 Patent, 3:4–13.

Defendants contend that PET's construction, to the contrary, includes that the IVK can be used with other software but then selectively excludes that it can be used alone. Defendants contend that because the IVK can do both, the addition of “with other software” is unnecessarily limiting and does nothing to aid the jury's understanding of the term. (Dkt. No. 86 at 20.)

Defendants further contend that PET's construction then confounds the IVK's role with digital signature. Defendants contend that the passage above shows that the IVK verifies that a

program image in memory corresponds to a digital signature by computing a digital signature for a program image and comparing it against a supplied value. (Dkt. No. 86 at 20 (citing '399 Patent, 5:18–20, 5:34–39; '550 Patent, 3:14–19).) Defendants contend that PET's vague "use" of a digital signature not only fails to define what such "use" is but eliminates the essential comparison function of the IVK.

Finally, Defendants assert that the specifications teach that, like the tamper resistant key module, the IVK is tamper resistant as that term is used in the context of these patents: "[a]n IVK is . . . 'armored' using methods to ensure that it is not easily tampered with," which simply means that it is tamper resistant. '399 Patent 5:20–22; *see also id.* at 6:2–3 ("[T]he tamper resistant compiler is applied to the IVKs . . .").

In reply, PET contends that its construction is supported by the specification of both patents. PET further states that any description contained in one patent but not the other would strongly support the notion that such description is merely of a preferred embodiment, and not definitional. PET contends that the '399 Patent states simply that "[a]n integrity verification kernel (IVK) is software that verifies that a program image corresponds to the supplied digital signature" and the '550 Patent states "[a]n IVK is software that verifies that a program image corresponds to a supplied digital signature." '399 Patent 5:18–26; '550 Patent 3:4–13.

PET asserts that both patents also describe a multitude of uses for the IVKs and describe them working in connection with other software. (Dkt. No. 87 at 6.) PET points to Defendants' construction as showing that an IVK can be "a small code segment," indicating that it would often be used in connection with other software. PET contends, however, neither the claim language nor the specification requires that a failure to verify the code necessarily results in the software becoming unable to execute. PET further states that the specification specifically

describes, as noted above, embodiments where the code continues to operate even while under attack. (Dkt. No. 87 at 6.)

As to the '550 Patent, PET contends the specification of the '550 Patent describes an IVK as “software that verifies that a program image corresponds to the supplied digital signature.” '550 Patent 3:4–6. PET further points to the file history for the '550 Patent. (Dkt. No. 87 at 10 (citing Dkt. No. 78 Ex. D at FH0197 (“The present invention also determines the integrity of the remote process, to detect if it has been tampered with, through the use of an [IVK].”))).)

### **Analysis**

PET points to extrinsic glossary and dictionary evidence, with regard to a constituent element of the term, to contend that the term has an ordinary meaning known in the art. For example, PET identifies 2016 websites, a 2006 source, and makes mere attorney argument that some of the evidence dates to 2001. The Court finds that PET’s extrinsic evidence provides little probative value as to the meaning of the term as a whole to one skilled in the art in 1997. *See Brookhill-Wilk*, 334 F.3d 1299–1300; *See Phillips*, 415 F.3d at 1312–13 (“We have made clear, moreover, that the ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.”).

In contrast, the specifications provide clear statements of meaning. *See 3M Innovative Props. Co. v. Tredegar Corp.*, 725 F.3d 1315, 1321 (Fed. Cir. 2013) (noting idiosyncratic technical terms may be best understood by the reference to the specification). The '399 Patent states simply that “[a]n integrity verification kernel (IVK) is software that verifies that a program image corresponds to the supplied digital signature” and the '550 Patent states “[a]n IVK is software that verifies that a program image corresponds to a supplied digital signature.” '399

Patent 5:18–26; ’550 Patent 3:4–13. In fact, both parties identify these passages as definitional. At the hearing, PET agreed to the meaning as stated in ’399 Patent 5:18–20 and stated that PET merely wants to make clear that the software can be used with other software. (Dkt. No. 98 at 66–67.) What PET seeks to clarify is explicitly not disputed by Defendants and only serves to make the term more confusing. (Dkt. No. 86 at 20.) At the hearing, Defendants acknowledged the passages at ’399 Patent 5:18–26 and ’550 Patent 3:4–13, but stated that the rest of the passages further inform the full meaning of IVK. (Dkt. No. 98 at 67–68.)

The full passages in question read:

An integrity verification kernel (IVK) is software that verifies that a program image corresponds to the supplied digital signature. An IVK is a small code segment that has been “armored” using methods to ensure that it is not easily tampered with. An IVK can be used alone, to ensure that its tasks are executed correctly, or it can be used in conjunction with other software to provide the assurance that the other software has executed correctly (that is, they can be used as verification engines).

’399 Patent 5:18–26.

An IVK is software that verifies that a program image corresponds to a supplied digital signature. This provides a robust mechanism for detecting changes made to executing software, where those changes might be caused by transmission errors or malicious attacks to the software. Any change to the software results in a failure in the verification process. IVKs for tamper resistant software are constructed to perform self-checks of object code, bilateral authentication of partner modules, and checks on local and remote data to verify the integrity of a software module.

’550 Patent, 3:4–13. Defendants include in their construction most of the limitations Defendants sought to add with regard to “tamper resistant” module for many of the same reasons. For the same reasons as recited above, the Court declines to include those limitations. The passages described above provide a clear and concise understanding of the terms (“[a]n integrity verification kernel (IVK) is software that verifies that a program image corresponds to the supplied digital signature” and “[a]n IVK is software that verifies that a program image

corresponds to a supplied digital signature”). In addition, both passages make clear the IVK software is tamper resistant. The discussion, with regard to the meaning of “tamper resistant,” is thus relevant to the use of IVK in both patents. In context of the specifications, an integrity verification kernel is software that verifies that a program image corresponds to a supplied digital signature and software that is tamper resistant.

**The Court construes “integrity verification kernel” to mean “software that verifies that a program image corresponds to a supplied digital signature and that is resistant to observation and modification.”**

**7. “integrity verification kernel code” (’399 Patent Claim 10)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
source code that can be used in conjunction with other software to determine that code has not been altered through the use of a digital signature	source code for calculating digital signatures that is generated using the asymmetric public key for the manifest of the selected program

The principal issue is whether or not the addition of “code” to “integrity verification kernel” requires additional construction beyond “integrity verification kernel.”

**Positions of the Parties**

PET contends that all parties agree that “integrity verification kernel code” is source code for the IVK. PET contends that incorporating “code” at the end of “integrity verification kernel,” does not alter the meaning of the IVK. PET contends that once the meaning of “integrity verification kernel” is determined by the Court, addition of the word “code,” a common word in the art of computer programming, should not require resorting to the specification to resolve any ambiguity. PET also asserts that the mere addition of “code” should not cause the term to be

rewritten to read on Defendants' interpretation of one of the preferred embodiments. (Dkt. No. 87 at 7.)

PET asserts that the specification describes creating digital signatures using private keys. '399 Patent 1:59–62 (“The second use is digital signatures where the public key is used to verify the digital signature while the private key is used to create the signature.”). PET objects to Defendants' construction as requiring that the digital signature is generated using “an asymmetric public key for the manifest of the selected program” and not a private key. PET contends that this, thus, excludes this embodiment of the disclosed invention. (Dkt. No. 78 at 19.)

Defendants contend their construction is the meaning given to this coined term by the patentees (acting as their own lexicographer), as expressed in the '399 Patent specification. Defendants assert that claim terms, like the term “integrity verification kernel code,” that do not have a customary meaning within the art are to be construed by reference to the specification. Defendants point to the specification: “key compiler 208 computes the Montgomery components of the asymmetric public key 200 for the manifest and generates IVK source code for key module 210 for calculating digital signatures using those components.” '399 Patent 9:33–36.

Defendants contend that, thus, the IVK source code is code for calculating digital signatures and is generated by using the asymmetric public key for the manifest of the selected program, which is Defendants' proposed construction. Defendants assert that PET's proposal nullifies the element of claim 10 requiring “combining manifest parser generator code and the integrity verification kernel code to produce the integrity verification kernel.” '399 Patent, 11:30–32. Defendants also point to the specification for such requirements: “[t]he generated ‘C’ IVK source code for the key module 210 and the manifest parser generator source code 212 are

combined into the single IVK source code module 206.” *Id.* at 9:51–54. Defendants assert that PET’s proposal also conflates the purpose of the IVK code with that of the IVK itself, of which the IVK code is merely a part. At the hearing, Defendants pointed to Figure 5 as demonstrating that the IVK code is limited to just a portion of the IVK because the IVK generation function 204 includes multiple types of code. (Dkt. No. 98 at 73.)

Defendants also state that PET’s allegation that Defendants’ construction excludes creating digital signatures using private keys is incorrect, because digital signatures and IVK code are two separate and distinct items.

### **Analysis**

The Court has construed the term “integrity verification kernel” above. The context of the usage of “integrity verification kernel code” within the asserted claim is indicative that merely adding “code” does not require changing the construction for “integrity verification kernel.” PET acknowledged to the Court at the hearing that, having construed “integrity verification kernel,” the addition of “code” does not mandate further construction. (Dkt. No. 98 at 70–71.)

Claim 10 begins by stating that the “generating an integrity verification kernel” step of claim 9 comprises a collection of steps. The first recited claim 10 step references “accessing an asymmetric public key of a predetermined asymmetric key pair associated with a manifest.” The second recited claim 10 step includes “producing integrity verification kernel code with the asymmetric public key for verifying the signed manifest.” It is clear from the claim language that the code is merely code of the integrity verification kernel. Further, the claim itself provides the details as to the production of the code: “with the asymmetric public key for verifying the signed manifest.” Defendants’ proposed construction adds needless confusion, and the claim language itself best describes the claimed requirements for how the code is produced. *See Phillips*, 415

F.3d at 1314 (the claims themselves can provide substantial guidance in determining the meaning of particular claim terms). Having construed “integrity verification kernel,” no further construction is needed for “integrity verification kernel code.” It is clear that the term recites code of the “integrity verification kernel” and, elsewhere, the claim provides the limitations regarding producing the code. Finally, as to Defendants’ argument that Figure 5 limits the IVK code to just a portion of the source code of the IVK module, the Court does not find that the specification creates such a fine distinction. Defendant is correct that block 210 of Figure 5 is referenced as “IVK source code for key module.” However, block 206 entitled “IVK source code module” encompasses all of the output of “integrity verification generation function 204.” This reference to “IVK source code module” is broader than the distinction Defendants attempt to make. ’399 Patent 9:17–31, Fig. 5.

**The Court finds that “integrity verification kernel code” needs no further construction.**

**8. “manifest” (’399 Patent Claim 10; ’550 Patent Claim 16)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
code and/or data that contains metadata describing other code	a data file containing a statement regarding the integrity and authenticity of a specific installation of the selected program comprising a unique identifier for that specific installation and its corresponding digital signature

The parties disagree as to what the term’s ordinary meaning is, whether that meaning should apply, and if not, whether the specification is limited as asserted by Defendants.

**Positions of the Parties**

PET contends that “manifest” has a well-known meaning in the software art, citing 2016 online dictionaries. (Dkt. No. 78 at 19 (citing <http://www.yourdictionary.com/manifest> (“A file

containing metadata describing other files.”) and [https://en.wikipedia.org/wiki/Manifest\\_file](https://en.wikipedia.org/wiki/Manifest_file) (“A manifest file in computing is a file containing metadata for a group of accompanying files that are part of a set or coherent unit.”)).) PET contends that in software, a “manifest” is like a ship’s manifest—it describes what is inside the code, the way a ship’s manifest describes what is inside the ship. PET contends that although the specification describes various attributes of embodiments of the manifest in the invention, there is no reason to read these limitations from the specification into the claims.

Defendants contend that the ’399 Patent specification states what a manifest is:

The manifest is a statement of the integrity and authenticity (i.e., a signature) of the trusted player software. The manifest is generated by the manufacturer of the trusted player or other provider of the trusted player software. Generally, the manifest is a credential about the trusted player including a digital signature of the trusted player software. Signed manifests describe the integrity of a list of digital objects of any type and associate arbitrary attributes with those objects in a manner that is tightly binding and offers non-repudiation.

...

The trusted player and its signature are freely distributable. However, there is no secret (such as a decryption key) embedded in the trusted player. In contrast, the manifest 46 is unique for each trusted player 42. It contains a unique identifier relating to the trusted player. For example, the unique identifier could be a number randomly generated by the manufacturer or other provider, a serial number, a credit card number, etc.

’399 Patent 6:46–7:15. Defendants also point to the ’550 Patent specification:

...the tamper resistant module will only reveal the secret if Process B’s credentials and actual image in memory agree. Process B’s credentials may be sent along with the tamper resistant module if Process A learns of Process B’s credentials beforehand, or they may be examined at Process B’s system by the tamper resistant module (i.e., the location of the credentials on process B’s system could be passed as an argument to the tamper resistant module before it is executed). The credentials typically will be a predetermined signed manifest of Process B’s code image.

'550 Patent 2:39–48. Defendants contend that PET's proposed construction ignores what the patentees said about the manifest and, instead, is based entirely on non-contemporaneous extrinsic evidence.

PET contends that even if one assumes, *arguendo*, that “manifest” has no ordinary meaning, there is still simply no reason to incorporate a plethora of limitations from the specification. (Dkt. No. 87 at 7.) PET contends that the claims explain in detail the various requirements of what needs to be included in the manifest. PET points to claim 10 of the '399 Patent:

. . . accessing an asymmetric public key of a predetermined asymmetric key pair associated with a manifest of the program signed by an asymmetric private key of the predetermined asymmetric key pair, producing integrity verification kernel code with the asymmetric public key for verifying the signed manifest of the program . . . .

'399 Patent Claim 10. PET contends that this requires that the manifest for the program be digitally signed with a private key, which would be an unnecessary limitation if the construct of “manifest” alone required everything proposed by Defendants. Further, PET contends there is nothing in the claim about the “integrity and authenticity of a specific installation” of the program, much less “a unique identifier for that specific installation.” (Dkt. No. 87 at 7.) PET contends that Defendants are “mixing and matching” intrinsic evidence between the two separate patents asserted in order to read limitations into one patent that do not even appear in the specification for that patent. PET contends that claim 16 of the '550 Patent only requires that “the first process is verified when a digital signature of the first process determined by the module corresponds to a predetermined signed manifest of the first process.” PET contends that there is no reason to read limitations from the '399 Patent into the '550 Patent, and the broad description in the '550 Patent only supports a broader reading of the term. (*Id.* at 7–8 (citing '550

Patent at 2:47–48 (“The credentials typically will be a predetermined signed manifest of Process B’s code image.”).)

### Analysis

PET points to 2016 extrinsic online dictionary evidence to contend that the term has a well-known meaning in the art and that such meaning relates to metadata. The Court finds that PET’s extrinsic evidence provides little probative value as to the meaning of the term as a whole to one skilled in the art in 1997. *See Brookhill-Wilk*, 334 F.3d 1299–1300; *See Phillips*, 415 F.3d at 1312–13 (“We have made clear, moreover, that the ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.”). At the hearing, Defendants noted that the term does carry an ordinary meaning. Defendants then stated that, in context of the patent, the term is described as “a statement of the authenticity and the integrity of a program.” (Dkt. No. 98 at 75–76.)

The ’399 Patent specification provides meaning to the term in conformance with Defendants’ acknowledgement:

The manifest is a statement of the integrity and authenticity (i.e., a signature) of the trusted player software. The manifest is generated by the manufacturer of the trusted player or other provider of the trusted player software. Generally, the manifest is a credential about the trusted player including a digital signature of the trusted player software. Signed manifests describe the integrity of a list of digital objects of any type and associate arbitrary attributes with those objects in a manner that is tightly binding and offers non-repudiation.

...

The trusted player and its signature are freely distributable. However, there is no secret (such as a decryption key) embedded in the trusted player. In contrast, the manifest 46 is unique for each trusted player 42. It contains a unique identifier relating to the trusted player. For example, the unique identifier could be a number randomly generated by the manufacturer or other provider, a serial number, a credit card number, etc.

'399 Patent 6:46–7:15. Figure 3 provides a diagram of an example manifest containing data such as version number, cryptographic algorithm, signature version, and a digital signature. '399 Patent 7:4–7. As noted, the manifest is a “statement of the integrity and authenticity (i.e., a signature)” of a trusted program. '399 Patent 6:46–47. “Generally, the manifest is a credential about the trusted player including a digital signature of the trusted player software.” *Id.* at 6:50–52. Thus, as described in the '399 Patent, a manifest is a credential of the integrity and authenticity (i.e., a digital signature) of software. Though Defendants seek to include specific further embodiments from the '399 Patent passage in question, it is clear that the passage provides a broader disclosure not limited to all the elements of Defendants' construction.

Similarly, the '550 Patent references a signed credential for software:

... the tamper resistant module will only reveal the secret if Process B's credentials and actual image in memory agree. Process B's credentials may be sent along with the tamper resistant module if Process A learns of Process B's credentials beforehand, or they may be examined at Process B's system by the tamper resistant module (i.e., the location of the credentials on process B's system could be passed as an argument to the tamper resistant module before it is executed). The credentials typically will be a predetermined signed manifest of Process B's code image.

'550 Patent 2:39–48. In the context of the specification in each patent, the term relates to a credential of the integrity and authenticity of software. This also conforms to the language surrounding the term in the claim of each patent.

**The Court construes “manifest” to mean “a credential of integrity and authenticity.”**

**9. “manifest parser generator code” (’399 Patent Claim 10)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
manifest source code which can be compiled so the manifest can be used	static source code that includes the integrity verification kernel’s entry code, generator code, accumulator code, and other code for tamper detection

PET contends the term is self-explanatory. Defendants contend the term is coined in the specification.

**Positions of the Parties**

PET contends the term is self-explanatory. (Dkt. No. 78 at 20.) PET contends that the purpose of the manifest is to describe code and while there must be code which can be compiled so that the manifest can perform its function, there is no claim requirement limiting that code to any particular type. (Dkt. No. 78 at 20 (citing ’399 Patent 9:51–54 (“The generated ‘C’ IVK source code for the key module 210 and the manifest parser generator source code 212 are combined into the single IVK source code module 206.”))).) PET contends that in this example, the code is described as being in the “C” programming language in the preferred embodiments, but there is no reason to read this limitation from the specification into the claims. PET contends that the rest of Defendants’ limitations are also merely adding embodiments from the specification.

Defendants contend that the term is a term “coined by the patentee [that is] best understood by reference to the specification,” as the patentees again acted as their own lexicographers. *3M*, 725 F.3d at 1321. Defendants contend their construction is consistent with the specification definition that “[t]he manifest parser generator source code 212 *is* static source code that includes the IVK’s entry code, generator code, accumulator code, and other code for tamper detection.” ’399 Patent 9:43–46 (emphasis added). Defendants contend that as “manifest

parser generator code” is a coined term, it is neither self-explanatory, nor does it have an established meaning in the art.

### **Analysis**

Though PET contends the term is self-explanatory, the Court disagrees. PET has not pointed to any evidence that the Court finds sufficient to establish that this term has a self-explanatory meaning to a person of ordinary skill in the art. PET cites to a specification passage: “[t]he generated ‘C’ IVK source code for the key module 210 and the manifest parser generator source code 212 are combined into the single IVK source code module 206.” ’399 Patent 9:51–54. One passage somewhat corresponds to the end of claim 10: “combining manifest parser generator code and the integrity verification kernel code to produce the integrity verification kernel.” ’399 Patent 11:29–31. Though the specification passage and the claim both describe that the manifest parser generator code and the integrity verification kernel code are combined, neither provides guidance as to what the “manifest parser generator code” is. Elsewhere, the specification provides clear guidance: “[t]he manifest parser generator source code 212 *is* static source code that includes the IVK’s entry code, generator code, accumulator code, and other code for tamper detection.” ’399 Patent 9:43–46 (emphasis added). The specification description controls.

**The Court construes “manifest parser generator code” to mean “static source code that includes the integrity verification kernel’s entry code, generator code, accumulator code, and other code for tamper detection.”**

**10. “address space” (’550 Patent Claim 14)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
memory used by one or more processor(s) during operation	all memory locations available to a process

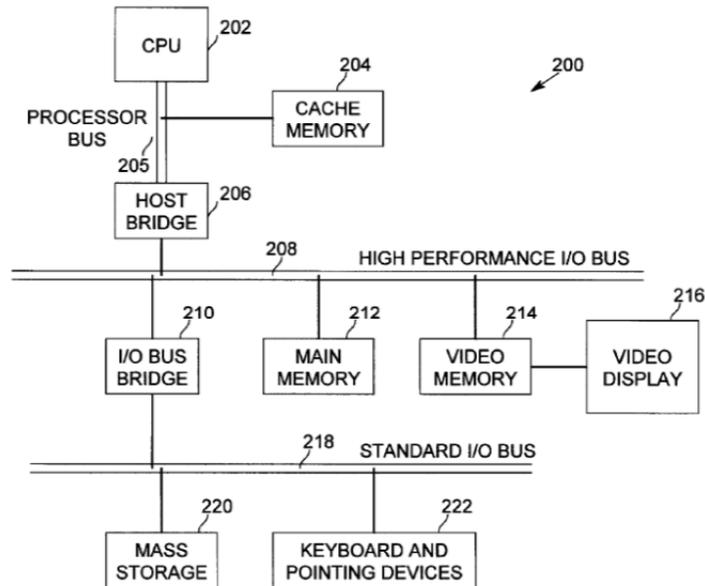
PET contends that the claim merely requires that the two processes operate in different address spaces. Defendants contend that the claim requires that all the memory locations available to the first process must be different than all the memory locations available to the second process.

**Positions of the Parties**

PET contends that the ordinary meaning of the term “address space” is “the range of memory [a processor] uses while running.” (Dkt. No. 78 at 21 (citing 2016 online technical dictionaries).) PET contends that the invention of the ’550 Patent involves a first and second process that are running on different memory spaces during operation, in accordance with the ordinary meaning of the term, as confirmed by the specification. (*Id.*) PET points to the specification:

It would be better if one party could authenticate the other party to ensure that the other party has not be [sic] tampered with or “hacked”, as opposed to just validating that the other party shares the secret. This can be done when the parties share the same address space by checking the contents of memory of the other party, computing its digital signature, and verifying its integrity. However, this cannot be accomplished across different process address spaces unless the memory is shared.

’550 Patent 1:25–34. PET contends that an address space is not the entire range of memory potentially “available” to a process as Defendants contend. Instead, PET contends it is the memory actually used by a process. (Dkt. No. 78 at 21 (citing ’550 Patent 2:18–20 (“These processes do not share an address space and may or may not exist on the same processor or computer system.”))).) PET points to the hardware of Figure 3 of the ’550 Patent for a computer system:



**FIG. 3**

'550 Patent Fig. 3. PET states that the '550 Patent explains:

FIG. 3 illustrates a sample computer system suitable to be programmed with the authentication method in accordance with embodiments of the present invention. Sample computer system 200 may be used to execute processing steps described above for Process A, Process B, or both. When Process A and Process B are on systems remote from each other, Process A is executing on a first sample computer system and Process B is executing on a second sample computer system connected to the first sample computer system via a network.

*Id.* at 4:61–5:3.

PET states that the patent, thus, discloses that the same computer system sharing the same potentially available range of memory may be used to execute processing steps for both Process A and Process B. PET states that with respect to Figure 3 of the '550 Patent, the memory attached to the processor would clearly be “accessible” to both the first and second processes. PET contends that what is required is that the first and second processes be actually running in different memory spaces during operation, not that the first and second process could not (in theory) access the memory of the other. PET contends that otherwise, it would not be possible to describe different processes running on the same computer as being in different address spaces,

since all the memory for the computer would be “available” to the processes. (Dkt. No. 78 at 22–23.)

Defendants contend that if the memory locations of the first process are available to the second process, there is no need for the claimed invention—the second process can verify the first process by accessing the first process’s memory locations directly. In particular, Defendants contend that, in the Background of the Invention section of the ’550 Patent, the patentee indicates the ability to verify that another party has not been modified is simple, if the two parties share an address space. (Dkt. No. 86 at 25 (citing ’550 Patent at 1:25–32 (“[Verification] can be done when the parties share the same address space by checking the contents of memory of the other party, computing its digital signature, and verifying its integrity.”))).) Defendants assert that from this passage alone, it is clear that “share the same address space” simply means that the first party can access the memory locations where the second party is loaded, in order to “check the contents” and “comput[e] its digital signature.” (*Id.*)

Defendants also contend that it is unclear whether the “processor(s)” introduced in PET’s construction is the “processing unit” recited later in the claim, or some other processor(s). (Dkt. No. 86 at 25.)

### **Analysis**

The term in question is merely “address space.” Both parties add additional limitations to this relatively clear term beyond the meaning of the term. The context of usage in the claim provides guidance: “a first process operating in an address space different than that of a second process.” ’550 Patent Claim 14. Defendants would have the term “address space” require that all possible memory locations available to the first process be different from all possible memory locations available to the second process. However, that is not what is claimed. The claim itself

focuses on the address space that the process is “operating in.” This is not the entire possible address space that a process could operate in. *See Phillips*, 415 F.3d at 1314 (the claims themselves provide substantial guidance in determining the meaning of particular claim terms).

Moreover, the specification provides further guidance contradicting Defendants’ construction. *See Accent Packaging, Inc. v. Leggett & Platt, Inc.*, 707 F.3d 1318, 1326 (Fed. Cir. 2013) (holding that a construction that excludes the preferred embodiment “is rarely, if ever, correct.”). First, the patent describes that the processes are not required to be within separate systems but could in fact be in the same processor: “these processes do not share an address space and may or may not exist on the same processor or computer system.” ’550 Patent 2:18–20. Again, the focus is on whether the processes are sharing space during operation not what the processes could be capable of using. Two processes running on the same processor counsels against Defendants’ construction. Further, with regard to Figure 3, the specification also makes clear that the two processes could be performed on the same computer system: “Sample computer system 200 may be used to execute processing steps described above for Process A, Process B, or both.” ’550 Patent 4:63–65. As to the memory where the processes operate the specification explicitly states:

These elements perform their convention functions well known in the art. In particular, mass storage 220 is used to provide permanent storage for the executable instructions of the authentication and tamper resistant programs/applications, whereas main memory 212 is used to temporarily store the executable instructions of authentication and tamper resistant programs/applications during execution by CPU 202.

’550 Patent 5:14–21. Thus, it is clear that for the embodiment in which “computer system 200” executes “both” processes, the memory available for storage of the processes may be mass storage 220 (permanent storage) and main memory 212 (temporary storage during execution).

PET’s construction, on the other hand, has potential to interject confusion and ambiguity into the term as PET adds “one or more processor(s).” Elsewhere, claim 14 calls out “a processing unit” and the “first process to be executed by the processing unit.” ’550 Patent 6:21–25. PET’s construction would create confusion as to whether the newly injected “processor(s)” include the claimed “processor unit.” Further, PET adds the concept of the memory that is “used” “during operation.” This is not an inherent limitation of “address space” but rather a result of the surrounding claim language which describes “a first process operating in an address space.” The surrounding claim language itself is understandable. The parties agree that “address space,” at its base, references memory. This also conforms to the usages in the specification cited by both parties. At the hearing, PET further agreed that the term references memory “locations.” (Dkt. No. 98 at 79.) Having rejected the extraneous limitations sought by both parties, the proper construction need only reference the memory concept.

**The Court construes “address space” to mean “memory locations.”**

**11. “first process” (’550 Patent Claims 14–17) / “second process” (’550 Patent Claim 14)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
<p>“first process:”<sup>7</sup>  a series of actions or steps running in different address spaces from the second process</p>	<p>“first process:”  an instance of a computer program that is executing within a specific address space that is different from the address space of the second process</p>

PET objects to Defendants limiting each “process” to a single computer program. Defendants object to PET’s construction as being divorced from computer software.

---

<sup>7</sup> Both parties utilize the same construction for “second process” except they replace the word “second” with “first.”

## **Positions of the Parties**

PET contends that the claim term “process” is intended to be conceptually different from a “program.” PET contends that if the inventors of the ’550 Patent had wanted to limit the claims to an instance of a computer program, this language would have been readily available and known to them. Instead, the inventors used the term “process,” and the ordinary meaning of that term should apply. PET states that while the steps of the claimed first and second processes must run on processors, they need not be part of a single “program.” (Dkt. No. 78 at 23.)

PET states that the ’550 Patent specification rejects the notion that a process must be an instance of a single computer program:

In particular, mass storage 220 is used to provide permanent storage for the executable instructions of the authentication and tamper resistant programs/applications, whereas main memory 212 is used to temporarily store the executable instructions of authentication and tamper resistant programs/applications during execution by CPU 202.

’550 Patent 5:15–21. PET contends that, thus, the specification explicitly discloses that the series of steps that make up the first process and the second process need not be limited to a single computer program, much less an instance of only one, but can come from multiple “programs/applications” so long as the steps of the process are met.

Defendants contend that the parties’ dispute on these terms reflects a fundamental disagreement of whether the word “process,” in the meaning of the ’550 Patent, is a software process or a broader process (i.e., any sequence of steps). Defendants point to the claim language of “a storage medium having therein a plurality of programming instructions of the first process to be executed by the processing unit” and “a digital signature of the first process.” ’550 Patent Claims 14, 16. Defendants contend that both of these limitations reference computer software and neither of these limitations have meaning if the “process” is simply an abstract set

of steps. Defendants contend that “process” clearly has its ordinary meaning within the computer arts—an executing computer program. (Dkt. No. 86 at 26 (citing 1992 technical publication).)

Defendants contend that the specification language identified by PET does not mention “process” at all but simply describes the general purpose hardware used to convert method claims into apparatus claims. (*Id.* at 27 (citing ’550 Patent 5:14–15 (“These elements perform their conventional functions well known in the art.”)).)

### **Analysis**

At the hearing, PET acknowledged that the meaning of the claim term references programming instructions. (Dkt. No. 98 at 82–83.) In context of the specification and claims, it is clear that “process” references computer software, and it is clear that neither party disputes this. The remaining dispute focuses on whether the process must be a single instance of a computer program. The ’550 Patent specification does not limit a “process” to a single computer program. Rather, a broader context is described: “the parties or processes performing the [challenge-response] protocols” ’550 Patent 1:12–14. Figure 1 is described with relation to two processes interacting, Process A and Process B, without any limitation that each must be only a single computer program. ’550 Patent 2:15–48. Similarly, the interaction of Process A and Process B, as described with relation to Figure 2, makes no limitation on either process being a single computer program. *Id.* at 4:1–37. Moreover, the specification provides indication that the “process” may be more than a single computer program. For example, Process B is referenced as having an “operating system,” indicating that the process may additionally include an operating system. *Id.* at 4:45. Additionally, as noted by PET, in describing the hardware environment for the authentication techniques, “programs/applications” (plural) are referenced for the system of each process:

In particular, mass storage 220 is used to provide permanent storage for the executable instructions of the authentication and tamper resistant programs/applications, whereas main memory 212 is used to temporarily store the executable instructions of authentication and tamper resistant programs/applications during execution by CPU 202.

'550 Patent 5:15–21.

The parties use somewhat different language with regard to the operation of the process: PET referencing “running” and Defendants referencing “executing.” Neither party has justified deviation from the claim language itself which is “a first process operating in an address space different than that of a second process.” Defendants contend that the address space must be limited to one specific address space. However, the specification does not make such an emphasis. Rather, the specification states “different process address spaces.” '550 Patent 1:29–33. What is emphasized is that the space of Process A and Process B are different and, the processes do not operate in the same space, not that the space for either process is limited to only one space. *See* '550 Patent 1:44–45, 2:17–18, 2:54–55. In this context, “space” is not limited to a singular space but encompasses the plural.

**The Court construes “first process” to mean “software program(s) or application(s) operating in different address space than the second process.”**

**The Court construes “second process” to mean “software program(s) or application(s) operating in different address space than the first process.”**

## 12. “embedded” (’550 Patent Claim 14)

PET’s Proposed Construction	Defendants’ Proposed Construction
made an integral part of	compiled within

The parties dispute whether “embedded” requires “compiled.”

### Positions of the Parties

PET cites to 2016 non-technical dictionaries for the ordinary meaning of “embedded.” (Dkt. No. 78 at 26.) PET contends that the ordinary meaning was used during prosecution of the ’550 Patent: “Neither Berry nor Penzias teach or suggest that a key used to encode a response to a challenge may be embedded within a tamper resistant module, and that the key is accessible only after integrity verification of the remote process is performed.” (Dkt. No. 78 Ex. D at FH0199.) PET contends that, thus, “embedded,” consistent with its ordinary meaning, means that a secret (whether a key or otherwise) is an integral part of the tamper resistant module. PET contends that “compiled within” is not the normal meaning ascribed to the word “embedded.” Further, PET contends that Defendants’ construction is unclear as it inserts a more technical term “compiled within,” without explanation as to what that means.

Defendants point to the language of claim 14: “a secret embedded in the tamper resistant module.” Defendants assert that the “secret” in Claim 14 of the ’550 Patent is described in the intrinsic record as contained in the tamper resistant module: “The tamper resistant module 14 also contains a secret 16 which the tamper resistant module will not divulge until the integrity verification for Process B 12 is a success” (’550 Patent 3:30–32); “Process A creates a tamper resistant module containing a secret” (*Id.* at 4:3–4); and “a secret (key) obtained from a tamper resistant module” (Dkt. No. 78 Ex. D at FH0199.) Defendants contend that as the tamper resistant module in the ’550 Patent is executable software, Defendants’ proposed construction properly defines what it means to be contained within software. Defendants object to PET’s

arguments as relying solely on extrinsic evidence. Defendants contend that PET's proposed construction merely substitutes "embedded" with an unclear phrase "integral part of."

### **Analysis**

The disputed term is found in the claim in the context of "recover a secret embedded in the tamper resistant module." '550 Patent 6:29–30. Defendants seek to require "embedded" to be limited to "compiled within." The '550 Patent provides only one limited discussion regarding compiling:

The software is generated by using a tamper resistant compiler (not shown). The tamper resistant compiler is a compiler that, when applied to a well prepared software module, replaces the plain-text source code compiler generated image with a new image that is obfuscated. This self-decrypting software will only execute properly if no part of the image has been altered from the time it was compiled by the tamper resistant compiler. The tamper resistant compiler is a software approach towards providing kernels of software with the ability to run in a "hidden" execution mode.

'550 Patent 2:56–66. The passage does not directly address the method of embedding the secret in the module, much less provide a disavowal limiting the term to compiling. The term in question, though, is much broader, merely requiring "embedding." Defendants have not pointed to any disclaimer or disavowal limiting the meaning of "embedded." Rather, Defendants merely, at most, point to an embodiment of the specification. However, even a single embodiment is not necessarily enough to read a limitation into the claim from the specification. *Arlington*, 632 F.3d at 1254 ("[E]ven where a patent describes only a single embodiment claims will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using words of expressions of manifest exclusion or restriction.") (citation omitted). In addition, the specification repeatedly merely references the module "containing a secret." '550 Patent 1:47–48 ("creating . . . a module containing a secret"), 3:30 ("tamper resistant module 14 also contains a secret"), 3:34–35 ("tamper resistant module 14 containing the secret"), 4:3–4 ("tamper resistant

module containing a secret”). In context of the specification as a whole, “embedded” is not limited to compiling. Rather, it is clear that the secret is “contained” in a module. *Id.*

**The Court construes “embedded” to mean “contained.”**

**13. “challenge” (’550 Patent Claims 14, 17)**

<b>PET’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
prompt for information to authenticate a user	arbitrary data known to the second process and unknown to the first process until received from the second process

The dispute presented to the Court is whether the specification and file history limit the meaning of “challenge” to the particular embodiment described in the specification.

**Positions of the Parties**

PET cites to several 2016 online sources (Webopedia and Techopedia) to contend that its construction of “challenge” conforms to the ordinary meaning in the software art. (Dkt. No. 78 at 27.) PET contends that the inventors adopted this “well-known” meaning during the prosecution of the ’550 Patent: “Penzias discloses a simple variation on a well-known challenge-response protocol whereby a human requester must supply a selected one of several previously provided and stored pieces of information in order to participate in a credit card transaction.” (Dkt. No. 78 Ex. D at FH198.)

Defendants point to the embodiment specification: “Process A transmits the tamper resistant module containing the secret, a challenge, and optionally, a request for information to Process B.” ’550 Patent 4:7–9. If Process B is able to recover the secret from the tamper resistant module, “Process B encodes the challenge received from Process A to produce the response” and then sends the response back to Process A. ’550 Patent at 4:18–25; Fig.1. Then, “[i]f the decoded

response contains the challenge Process A sent to Process B, then Process B has been authenticated.” ’550 Patent 4:28–30.

Further, Defendants state that the patentee explained that “various challenge-response protocols and their deficiencies are described in ‘Applied Cryptography’ by Bruce Schneier.” ’550 Patent 1:34–36. Defendants state that the Schneier reference describes the challenge in such a challenge-response protocol as being “[a] random number, sometimes called a nonce, chosen by Alice and Bob respectively.” (Dkt. No. 87 Ex. D, APPLIED CRYPTOGRAPHY, 2d ed. 1996, at 57.)

Defendants contend that PET relies solely on extrinsic evidence and cannot point to anything in the specification that supports its proposed construction. Defendants contend that at no point does the ’550 Patent describe or even reference “user authentication.” (Dkt. No. 87 at 29.) Defendants state that if the claimed “challenge” referred to a “prompt for information to authenticate a user,” the claims would be limited to only those processes that include some form of user interaction. However, Defendants assert the ’550 Patent does not disclose a single embodiment that contemplates user interaction in the authentication process.

As to the file history, rather than adopting PET’s meaning, Defendants contend the patentees rejected PET’s interpretation and distinguished the prior art which disclosed “a simple variation on a well known challenge/response protocol whereby a human requester must supply a selected one of several previously provided and stored pieces of information.” (Dkt. No. 86 at 30 (quoting Dkt. No. 78 Ex. D at FH0198).) In particular, Defendants points to the statement:

Penzias discloses a simple variation on a well known challenge/response protocol whereby a human requester must supply a selected one of several previously provided and stored pieces of information in order to participate in a credit card transaction. This method provides some minimal level of protection by purportedly authenticating a human user to prevent theft of telephone service. However, the pre-stored information may be acquired by a thief just as a “PIN”

number can be. Penzias does not teach or disclose a remote process, executing in an address space different than a local process, being authenticated and having its integrity verified as presently claimed. *Penzias does not teach or suggest using the challenge in the challenge/response protocol as the response and encoding it with a secret (key) obtained from a tamper resistant module* only after integrity verification has been performed on the remote process, as is required by the limitations of the present claims.

(Dkt. No. 78 Ex. D FH0198–99 (emphasis added).)

As to the Schneier reference, PET states that Schneier never defines a “challenge” as a “nonce” in the manner described by Defendants. PET states that the word “challenge” does not even appear on the pages cited by the Defendants. (Dkt. No. 87 at 10.) Instead, PET states the cited portions are part of a larger discussion on authentication, which describes typical challenge-response mechanisms, such as passwords. (*Id.* (citing Dkt. No. 87 Ex. F at 52, APPLIED CRYPTOGRAPHY, 2d ed. 1996).) PET states that while the secret used to encode the challenge can be a nonce (e.g., dependent claim 13) the challenge is not the nonce, but a prompt for information. PET contends that, thus, there is no reason to depart from the ordinary meaning of this term.

### Analysis

The specification and file history make clear that challenge-response protocols were known in the art prior to the filing of the '550 Patent. '550 Patent 1:12–16, 1:19–25, 1:34–36; (Dkt. No. 78 Ex. D at FH0198–99.) Defendants do not contest that the term had an ordinary meaning. As described in the Background of Invention, a challenge-response is discussed more generally without the limitations sought by Defendants. For example, it is clear that, as known, a challenge-response may merely be a secret shared between the two processes. '550 Patent 1:12–29. This is described as being for the purpose to “authenticate the other party” or as part of an “authentication process.” *Id.* at 1:25–41. Similarly, the file history describes “well known

challenge/response” protocols to be based on supplying “one of several previously provided and stored pieces of information” that is used for “authenticating.” (Dkt. No. 78 Ex. D at FH198.)

Defendants rely primarily, first, on the Schneier reference and, second, on a portion of the file history. The Court determines that neither piece of evidence supports Defendants’ limiting construction. PET and Defendants take conflicting views of the Schneier reference. On balance, the Court finds that PET’s interpretation of the Schneier evidence is more proper. *See Teva*, 135 S. Ct. at 837–38 (noting the Court’s role in making underlying factual determinations in claim construction). It is noted that the portion of Schneier that Defendants reference is in relation to only one particular protocol (the Wide-Mouth Frog protocol). (Dkt. No. 86 Ex. D at 57.) However, even in this example, it is clear that this process relates to authentication through sharing a secret. The random number referenced is clearly just the embodiment of the particular example, not a redefinition of the term “challenge.” (*Id.*)

As to the file history, it is clear from the passage that a “challenge” is known in the context of “one of several previously provided and stored pieces of information” that is used for “authenticating.” (Dkt. No. 78 Ex. D at FH198.) The end of the passage does reference:

Penzias does not teach or suggest using the challenge in the challenge/response protocol as the response and encoding it with a secret (key) obtained from a tamper resistant module only after integrity verification has been performed on the remote process, as is required by the limitations of the present claims.

(Dkt. No. 78 Ex. D at FH0198–99.) This does not, however, redefine “challenge.” Rather, it merely states that Penzias does not use a challenge-response protocol as used in the claim. More particularly, application claim 14, at the time, recited: “recover a secret embedded in the tamper resistant module when the integrity of the local process is verified” and “receive a challenge from the second process, encode the challenge using the secret to produce a response, and send the response to the second process.” (*Id.* at FH0193.) It is clear that the file history statement did

not redefine “challenge” but distinguished the use of the challenge as described in the other explicit claim limitations.

Based upon the usage in the specification, file history, and Schneier reference, the meaning of “challenge” is best described in the context of a prompt for information for use in authentication.

**The Court construes “challenge” to mean “prompt for information for use in authentication.”**

**So ORDERED and SIGNED this 21st day of July, 2016.**

  
\_\_\_\_\_  
RODNEY GILSTRAP  
UNITED STATES DISTRICT JUDGE