

**UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF NEW YORK**

-----X

LUCIANO F. PAONE,

Plaintiff,

-against-

MICROSOFT CORPORATION,

Defendant.

-----X

**MEMORANDUM OF
DECISION AND ORDER**
07-cv-2973 (ADS)(ARL)

APPEARANCES:

Kirkland & Ellis LLP

Attorneys for the Plaintiff

153 East 53rd Street
New York, NY 10022

By: Andrew Gordon Heinz, Esq.,
Jeanne M. Heffernan, Esq.,
John Michael Desmarais, Esq.,
Jon Todd Hohenthanner, Esq.,
Ryan Charles Micallef, Esq., of Counsel

Woodcock Washburn LLP

Attorneys for the Defendant

2929 Arch Street
12th Floor
Philadelphia, PA 19104-2891

By: Dale M. Heist, Esq.,
Daniel J. Goettle, Esq.,
John E. McGlynn, Esq.,
Steven J. Rocci, Esq., of Counsel

Westerman, Ball, Ederer, Miller & Sharfstein, LLP

Attorneys for the Defendant

1201 RXR Plaza
Uniondale, NY 11556

By: Greg S. Zucker, Esq.,
Jeffrey A. Miller, Esq., of Counsel

SPATT, District Judge.

In this patent infringement case, the Plaintiff Luciano F. Paone (“Paone”) alleges that the Defendant Microsoft Corporation (“Microsoft”) has infringed United States Patent 6,259,789 (“the ‘789 Patent”), which Paone holds. Pursuant to the Supreme Court’s decision in Markman v. Westview Instruments, Inc., 517 U.S. 370, 116 S. Ct. 1384, 134 L. Ed. 2d 577 (1996), the Court previously construed the disputed claim terms of the ‘789 Patent. Presently before the Court is the Defendant’s motion for summary judgment. For the reasons set forth below, the motion is granted in part and denied in part.

I. BACKGROUND

A. Background of the Invention

For purposes of the present motion, the Court will restate the relevant facts as they appeared in the previous claim construction decision issued on February 9, 2011 (the “Markman Order”). See generally Paone v. Microsoft Corp., 771 F. Supp. 2d 224 (E.D.N.Y. 2011).

On July 10, 2001, the United States Patent and Trademark Office (“PTO”) issued the ‘789 Patent, entitled “Computer Implemented Secret Object Key Block Cipher Encryption and Digital Signature Device and Method”, to the Plaintiff Luciano F. Paone. The ‘789 Patent describes a method of translating (or “encrypting”) ordinary data (called “plaintext”) into encoded data (called “ciphertext”), so that the plaintext may not be viewed by an unintended reader. In cryptography, which is the science of encryption, such a method is called a “cipher”. Generally, encrypted ciphertext is later decoded, or “decrypted”, into plaintext, so that the data is again usable.

There are several ways to encrypt computer data, but the ‘789 Patent deals exclusively with “symmetric key” encryption of computer data, which requires that the encryptor and decryptor share knowledge of both a cipher and a key. Computer based symmetric key ciphers

divide roughly into two types: block ciphers and stream ciphers. As a general matter, a stream cipher employs its key to encrypt data one bit at a time, while a block cipher employs its key to encrypt bits in groups. The '789 Patent describes a block cipher.

Computer implemented block ciphers have been in wide use in the United States since at least 1976, when a block cipher called the "Data Encryption Standard" or "DES" was adopted by the United States government for general use. As of 1997, when Paone filed his patent application, dozens of additional block encryption algorithms had been published. However, most of those inventions used a single key to encrypt successive blocks of data. Paone's innovation—in the most general terms—was to change the encryption key for each data block, based on additional, randomly generated data.

B. Infringing Technology

There are two components of Microsoft's flagship computer operating system, Windows, that Paone asserts infringe the '789 Patent. Temporal Key Integrity Protocol ("TKIP") encryption is an industry-standard data encryption protocol used to encrypt and decrypt data that is transmitted over wireless local area networks. TKIP technology improves on prior wireless encryption standards by using a dynamic keying scheme in which the encryption key changes from one block of data to the next. The Defendant Microsoft implements or supports TKIP in many of its products, including Windows Vista, XP and 7, and Xbox 360. Paone claims that method claims 2 and 33 in the '789 Patent, as well as system claims 24 and 34 in the '789 Patent, are infringed by the TKIP technology. In order for TKIP to infringe any these claims, Paone necessarily must prove that TKIP incorporates every limitation of the claim, either literally or under the doctrine of equivalents.

The second component Paone asserts as being infringed is BitLocker. BitLocker is an encryption feature implemented in Microsoft software, which is used to encrypt and decrypt data on a customer's hard drive. BitLocker targets the "lost laptop" scenario, in which unprotected data could be vulnerable to theft or offline attacks. The BitLocker technology encrypts data on a hard drive using an encryption key that changes from one block of data to the next. As with TKIP, Microsoft has incorporated the BitLocker feature into a number of its products, including Windows Vista and 7 Ultimate. Paone claims that the method claims 2 and 33 in the '789 Patent are infringed by the BitLocker technology. In order for BitLocker to infringe either of these claims, Paone necessarily must prove that BitLocker incorporates every limitation of either of the claims, either literally or under the doctrine of equivalents.

In sum, Paone alleges that Microsoft's products implementing and supporting TKIP encryption infringe claims 2, 24, 33 and 34 of the '789 Patent, while products incorporating Microsoft's BitLocker encryption feature infringe claims 2 and 33 of the '789 Patent.

C. Procedural History

Paone filed his application for the '789 Patent on December 12, 1997. Following the initial application, the PTO rejected Paone's claims as unpatentable in three successive office actions, dated September 29, 1999, March 16, 2000, and October 30, 2000. In response to each of these rejections, Paone modified his claims and provided additional argument. Then, on July 10, 2001, the PTO deemed the described invention patentable, and issued the '789 Patent.

On July 23, 2007, Paone commenced the present action against Microsoft, alleging that Microsoft was infringing the '789 patent. Specifically, Paone asserts that two components of Microsoft's flagship computer operating system, Windows, infringe the '789 Patent, as set forth above. The parties proceeded with discovery, during which time Microsoft on May 16, 2008 requested the PTO to reexamine the '789 Patent. In early 2009, with the reexamination

proceeding still pending, the parties briefed claim construction motions. However, before holding a Markman hearing, the Court on April 5, 2009 stayed the case pending the resolution of the reexamination proceeding. Then, before the stay was lifted, Microsoft also filed two more reexamination requests, dated June 29, 2009, and July 27, 2009. The three reexamination proceedings as a whole resulted in the cancellation of claims 1, 3, 23, and 32 of the '789 Patent, but also a ruling that several claims, including claims 2, 24, 33, and 34, were patentable. After certain amendments, a number of other claims were also found to be patentable.

After the reexamination proceedings had been finalized, the Court on March 3, 2010 lifted the stay of the case. Partly as a result of the reexamination proceedings, Paone modified his position to assert that Microsoft was infringing claims 2, 24, 33, and 34 of the '789 Patent, all of which had been ruled patentable by the PTO. These claims read in full as follows (the language of claims 1, 23, and 32 are also included, because some of the asserted claims are dependent on those claims).

What is claimed is:

1. A computer implemented method for encrypting data comprising the steps of:
 - creating at least one object key in a block cipher, the at least one object key comprising data and methods that operate on said data;
 - creating a key schedule based upon the at least one object key;
 - encrypting a random session object key in a block cipher encryption process with the at least one object key;
 - encrypting a block of input plaintext data utilizing said key schedule;
 - modifying the at least one object key based on seeding from the random session object key;
 - modifying the key schedule based upon the at least one modified object key;
 - encrypting a next block of input plaintext data utilizing said modified key schedule; and

- repeating the steps of modifying the at least one object key, modifying the key schedule and encrypting utilizing the modified key schedule until the encrypting of blocks of plaintext data is completed.
2. A computer implemented method as defined in claim 1, wherein the modification of the key schedule is independent of the input plaintext data.
23. A cryptographic communications system comprising:
- at least two networked computer systems linked by a communication channel; and
 - each computer system including a central processing unit and a memory storage device for executing a block cipher encryption/decryption process;
 - wherein the encryption process transforms an input plaintext message to a ciphertext message and the decryption process transforms the ciphertext message to the input plaintext message, the encryption/decryption process using at least one dynamic object key which is modified using a non-linear function for each block of input data, each object key being associated with a different key schedule to encrypt/decrypt the input plaintext/output ciphertext message.
24. A cryptographic communications system as defined in claim 23, wherein the encryption/decryption process further includes the use of a random session object key having an initial state randomly generated by the computer system, and wherein the object key modifications are based on seeding from the random session object key.
32. A computer implemented method for encrypting data comprising the steps of:
- creating at least one object key in a block cipher, the at least one object key comprising data and methods that operate on said data;
 - creating a key schedule based upon the at least one object key;
 - encrypting a block of input plaintext data utilizing said key schedule;
 - modifying the at least one object key based on at least a non-linear function;
 - modifying the key schedule based upon the at least one modified object key;
 - encrypting a next block of input plaintext data utilizing said modified key schedule; and

repeating the steps of modifying the at least one object key, modifying the key schedule and encrypting utilizing the modified key schedule until the encrypting of blocks of plaintext data is completed.

33. A computer implemented method as defined in claim 32, where in the non-linear function is a hashing function.

34. A cryptographic communications systems [*sic*] as defined in claim 23, wherein the non-linear function is a hashing function.

It is crucial to note that claims 2 and 33 in the '789 Patent are method claims. See In re Kollar, 286 F.3d 1326, 1332 (Fed. Cir. 2002) (“the distinction between a claim to a product, device, or apparatus, all of which are tangible items, and a claim to a process, which consists of a series of acts or steps.”); see also NTP, Inc. v. Research In Motion, Ltd., 418 F.3d 1282, 1322 (Fed. Cir. 2005) (“The invention recited in a method claim is the performance of the recited steps.”). “The law of this circuit is axiomatic that a method claim is directly infringed only if each step of the claimed method is performed.” Muniauction, Inc. v. Thomson Corp., 532 F.3d 1318, 1328 (Fed. Cir. 2008). Paone claims that both of the methods claims are infringed by TKIP and BitLocker. On the other hand, 24 and 34 are systems claims, and not method claims. Paone claims that both of these claims are infringed, but only by TKIP and not by BitLocker.

Upon the lifting of the stay, the parties resumed their claim construction motions. Then, on the consent of the parties, the Court appointed Gale R. Peterson, Esq., an experienced and well-regarded patent attorney, as a special master in this case for the limited purpose of conducting a Markman hearing and issuing a report and recommendation to the Court on claim construction. On June 23, 2010 Special Master Peterson conducted a non-evidentiary Markman hearing, at which the parties' attorneys presented their arguments on claim construction. On August 11, 2010, Special Master Peterson issued an extensive and detailed Report and Recommendation on claim construction (the “Special Master’s Report”). In response to this

report, both Paone and Microsoft filed memoranda of law on September 17, 2010, each objecting in part to the Special Master’s Report.

Prior to the Markman hearing, the parties presented nine claim terms for construction. However, after meeting and conferring at the suggestion of the Special Master, the parties agreed on meanings for three of those terms. The remaining six terms that were in dispute were:

Disputed Terms/Phrases	Asserted Claims (or underlying independent claims) Containing the Disputed Terms/Phrases
“object key”	1, 23, 24, 32
“random session object key”	1, 24
“repeating the step[] of modifying the at least one object key”	1, 32
“block”	1, 23, 32
“key schedule”	1, 2, 23, 32
“block cipher”	1, 23, 26, 32

On February 9, 2011, this Court issued a decision construing these six claim terms (the “Markman Order”). As for “object key”, the Court concluded that the “object key” is an encryption key that is not available to the general public, and which is composed of both (1) key data and (2) methods that modify that key data. The Court also found that the object key need not be implemented in an object-oriented programming language. As for “random session object key”, the Court construed the term to mean an object key, as defined above, with data that is generated randomly for each instance of the use of the encryption application. As for “repeating the step[] of modifying the at least one object key”, the Court found the phrase to be understood as limited in the sense that the object key’s data, as it presently exists in the object key at each instance of modification, must be an input into the modification methods of that object key. As

for “block”, the Court adopted in its entirety the Special Master’s analysis and construction of the term, which is:

The term “block” as used in the claims and specification of the ‘789 patent means a sequence of bits wherein that sequence has a fixed length that does not vary from block-to-block. The length of a block may be determined through selection in an encryption algorithm. Plaintext data having a length longer than the length of a block is divided into blocks of fixed length.

As for “key schedule”, the Court construed this term as a string of bits that is used in encrypting a block of input plaintext data. Moreover, the Court explained that it is created by using an object key as an input to an expansion function, which increases the size of the object key to form the key schedule. In addition, sub-portions of the key schedule must be accessible to be used separately in the encryption algorithm if called for, but those sub-portions may be accessed merely by identifying those sub-portions’ positions in the string of bits that forms the key schedule. Finally, as for “block cipher”, the Court found that a cipher that encrypts data in blocks, but does so by performing a single, successive, comparison of each element of the data block to an element of key data, is not a block cipher as the term is used in the ‘789 Patent.

On December 21, 2011, the Defendant Microsoft filed the instant motion for summary judgment requesting a determination of non-infringement.

II. DISCUSSION

A. Legal Standard on a Motion for Summary Judgment

Summary judgment is appropriate “if the pleadings, the discovery and disclosure materials on file, and any affidavits show that there is no genuine issue as to any material fact and that the movant is entitled to judgment as a matter of law.” Fed. R. Civ. P. 56(c). A genuine issue of material fact exists if “the evidence is such that a reasonable jury could return a verdict for the nonmoving party.” Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 248, 106 S. Ct. 2505, 91 L. Ed. 2d 202 (1986). “Where the record taken as a whole could not lead a rational trier of

fact to find for the non-moving party, there is no genuine issue for trial.” Matsushita Elec. Indus. Co., Ltd. v. Zenith Radio Corp., 475 U.S. 574, 587, 106 S. Ct. 1348, 89 L. Ed. 2d 538 (1986) (citation and internal quotation marks omitted).

“The standard for summary judgment in a patent case is the same as in any other case.” CA, Inc. v. Simple.com, Inc., No. 02 Civ. 7248, 2009 WL 7445199, at *2 (E.D.N.Y. Mar. 5, 2009) (citing Desper Prods., Inc. v. QSound Labs, Inc., 157 F.3d 1325, 1332 (Fed. Cir. 1998) and Union Carbide Corp. v. Am. Can Co., 724 F.2d 1567, 1571 (Fed. Cir. 1984)). Summary judgment on the ground of noninfringement of a patent, the relevant inquiry in this case, may be granted “where the patentee’s proof is deficient in meeting an essential part of the legal standard for infringement liability.” Johnston v. IVAC Corp., 885 F.2d 1574, 1577 (Fed. Cir. 1989); see Travel Sentry, Inc. v. Tropp, 736 F. Supp. 2d 623, 631 (E.D.N.Y. 2010).

“When deciding issues in a patent case, a district court applies the law of the circuit in which it sits to nonpatent issues and the law of the Federal Circuit to issues of substantive patent law.” In re Omeprazole Patent Litig., 490 F. Supp. 2d 381, 399 (S.D.N.Y. 2007) (citing Invitrogen Corp. v. Biocrest Mfg., L.P., 424 F.3d 1374, 1378–79 (Fed. Cir. 2005)); see, e.g., Desenberg v. Google, Inc., No. 09 Civ. 10121, 2009 WL 2337122, at *5 (S.D.N.Y. July 30, 2009).

In this case, there is no dispute that the ‘789 Patent has been deemed valid upon its initial examination and subsequent reexamimation. Accordingly, the only issues remaining for determination with respect to the patent are infringement and damages. At this stage of the litigation, the Defendant contends that it is entitled to summary judgment on the remaining issue of patent infringement.

B. Patent Infringement

Pursuant to 35 U.S.C. § 271(a), “whoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent.” Patent infringement is “a strict liability offense”. In re Seagate Technology, LLC, 497 F.3d 1360, 1368 (Fed. Cir. 2007), cert. denied sub nom. Convolve, Inc. v. Seagate Technology, LLC, 552 U.S. 1230, 128 S. Ct. 1445, 170 L. Ed. 2d 275 (2008). Thus, while intent is not necessarily an element of a direct infringement claim, an alleged infringer cannot claim ignorance or a good faith belief of non-infringement in order to defend against a claim of infringement. See In re Omeprazole Patent Litig., 490 F. Supp. 2d 381, 413 (S.D.N.Y. 2007) (“Making, using, selling, or offering to sell matter covered by a patent without authority of the owner constitutes infringement regardless of knowledge or intent.”), aff’d, 281 Fed. App’x. 974 (Fed. Cir. 2008), cert. denied sub nom. Apotex Corp. v. Astrazeneca AB, ---U.S. ----, 129 S. Ct. 1593, 173 L. Ed. 2d 677 (2009); see also Metal Film Co. v. Metlon Corp., 316 F. Supp. 96, 111 n.15 (S.D.N.Y. 1970) (“neither lack of knowledge of the patent nor lack of intent to infringe is a defense on the issue of infringement.”).

To assess a patent infringement claim, there are two relevant inquiries. “First, the court must construe the patent’s claims as a matter of law to determine their proper scope.” Serby v. First Alert, Inc., No. 09 Civ. 4229, 2011 WL 4464494, at *4 (E.D.N.Y. Sept. 26, 2011). This task was previously completed by the Court in the Markman Order. “Second, a jury generally determines the factual issue of whether infringement has occurred, unless there is no genuine issue of material fact, in which case summary judgment is appropriate.” Id. Thus, as the Court previously resolved all disputes over the language of the patent claims, the issue of infringement

may be resolved as a matter of law on a motion for summary judgment. See Moore U.S.A. Inc. v. The Standard Register Co., 144 F. Supp. 2d 188, 191–92 (W.D.N.Y. 2001); see also Gart v. Logitech, Inc., 254 F.3d 1334, 1339 (Fed. Cir. 2001) (noting that infringement is properly decided upon summary judgment when no reasonable jury could find that every limitation recited in the properly construed claim either is or is not found in the accused device), cert. denied, 534 U.S. 1114, 122 S. Ct. 921, 151 L. Ed. 2d 886 (2002); Wolverine World Wide, Inc. v. Nike, Inc., 38 F.3d 1192, 1199 (Fed. Cir. 1994) (“In order for a court to find infringement, the plaintiff must show the presence of every . . . [limitation] or its substantial equivalent in the accused device.”).

Paone asserts that there is infringement by Microsoft as to four claims of the ‘789 patent. In particular, the Plaintiff asserts that four claims—2, 24, 33, and 34—are infringed in connection with the TKIP component, and two claims—2 and 33—are infringed in connection with the BitLocker component. As for the latter, all of the infringement claims in connection with BitLocker are asserted under the doctrine of equivalents (“DOE”), namely, not literal infringement. The Plaintiff’s claims are summarized as follows:

Claim	Accused Technology		Infringement Theory	
	TKIP	BitLocker		DOE
2 (method)	TKIP	BitLocker		DOE
33 (method)	TKIP	BitLocker		DOE
24 (system)	TKIP		Literal	DOE
34 (system)	TKIP		Literal	DOE

The core of the Defendant's present summary judgment motion on the basis of non-infringement is that neither of Microsoft's accused technologies—TKIP nor BitLocker—fall within the claim boundaries of the '789 patent set by the Court's previous Markman Order. “[I]nfringement is assessed by comparing the accused device to the claims[;] the accused device infringes if it incorporates *every limitation of a claim*, either literally or under the doctrine of equivalents.” Nazomi Commc'ns, Inc. v. Arm Holdings, PLC, 403 F.3d 1364, 1732 (Fed. Cir. 2005) (emphasis added). If, however, even one claim limitation is missing or not met, there is no literal infringement. Mas-Hamilton Group v. LaGard, Inc., 156 F.3d 1206, 1211 (Fed.Cir.1998). The Defendant's arguments center on three claim limitations; two of the limitations are found in all four claims, one of the limitations is found in only two claims.

The first limitation at issue is “block”, which is contained in all four claims. Microsoft contends that TKIP cannot meet this claim limitation either literally or under the doctrine of equivalents, and therefore TKIP cannot be found to infringe any of the four claims. See z4 Techs., Inc. v. Microsoft Corp., 507 F.3d 1340, 1348 (Fed. Cir. 2007) (“We presume, unless otherwise compelled, that the same claim term in the same patent or related patents carries the same construed meaning.” (internal quotation marks omitted)). As a practical matter, if Microsoft ultimately prevails on this argument, then claims 2, 33, 24, and 34 cannot be infringed by TKIP.

The next limitation at issue is the “object key” limitation, which also is contained in all four claims. Microsoft maintains that neither TKIP nor BitLocker meet this claim limitation, and therefore neither component can be found to infringe any of the four claims. Again, as a practical matter, if Microsoft ultimately prevails on this argument, then claims 2, 33, 24, and 34 cannot be infringed by TKIP or BitLocker.

The last limitation at issue is the “repeating step” limitation, which is only contained by incorporation in the method claims 2 and 33. Microsoft maintains that neither TKIP nor BitLocker meet this claim limitation, and therefore neither component can be found to infringe the two method claims. As a practical matter, if Microsoft ultimately prevails on this argument, then claims 2 and 33 cannot be infringed by TKIP or BitLocker.

Ultimately, the resolution of one of these contentions has the potential to dispose of the entire matter, or at least dispose of certain claims that are at issue. For instance, if there is no question of fact that BitLocker does not meet the “repeating step” limitation found in the method claims 2 and 33, then it is irrelevant whether it also meets the “object key” limitation in those two claims. However, for purposes of the present motion, the Court will fully address all issues raised by the parties and will assess each claim limitation independently. In the conclusion, the Court will summarize the practical effect of the Court’s findings on the claims that remain.

C. As to Whether Microsoft Encrypts “Blocks” of Plaintext with a “Block Cipher”

The first relevant inquiry is whether Microsoft’s TKIP component literally infringes the ‘789 patent or infringes under the doctrine of equivalents, in light of the method in which TKIP encrypts “blocks” of plaintext with a “block cipher”.

As mentioned above, TKIP is an industry-standard data encryption protocol used to encrypt and decrypt data that is transmitted over wireless local area networks. The Defendant Microsoft implements or supports TKIP in many of its products. The data units used in the TKIP encryption, called MAC Protocol Data Units (“MPDUs”), are the only components that could potentially qualify as “blocks” in a “block cipher” under the ‘789 patent.

When a data message such as a computer document is transmitted over a TKIP-enabled network, the system determines the size of the message and compares it to the maximum TKIP

block size, also known as the “fragmentation threshold.” If, and only if, the message has a length that is precisely an exact multiple of the applicable fragmentation threshold, will this result in an encryption where each and every MPDU has the same length. However, if the message length exceeds the fragmentation threshold, the message will be fragmented and encrypted (and subsequently decrypted) as a series of TKIP MPDUs, with the final MPDU having a different length. This last MPDU consists of what are the leftover bits after the message is divided. The capability to split a data message if the fragmentation threshold is exceeded is a requirement of the TKIP standard.

According to Microsoft, every asserted claim—2, 33, 24, and 34—requires encryption of “blocks” of plaintext in a “block cipher.” (See ‘789 patent claims 2, 33, 24 & 34.) Thus, in order to prove infringement, Paone would necessarily need to demonstrate at trial that Microsoft’s TKIP component satisfies these limitations in the claims. Honeywell Int’l, Inc. v. United States, 70 Fed. Cl. 424, 446 (Fed. Cl. 2006) (noting that a *prima facie* case of literal infringement requires the accused matter to fall clearly within the patent claim, meaning that “every limitation of a claim [must] be met to establish literal infringement.”) (quoting Intellicall, Inc. v. Phonometrics, Inc., 952 F.3d 1384, 1389 (Fed. Cir. 1992) (overturned on other grounds); see also Research Plastics, Inc. v. Federal Packaging Corp., 421 F.3d 1290, 1297 (Fed. Cir. 2005) (“Literal infringement requires that the accused device embody each of the limitations of the asserted claim.”). In other words, for Microsoft to succeed on its motion for summary judgment, it must show that there is no genuine issue of material fact as to whether TKIP encrypts plaintext “blocks” and does so in a “block cipher”. Microsoft contends that the record clearly confirms that TKIP contains neither of these elements either literally or equivalently, and accordingly, cannot be found to infringe the ‘789 patent.

1. “Block”

As previously and thoroughly analyzed in the Markman Order, the claim term “block” is used to describe groups of data to be encrypted or decrypted. As relevant to the allegedly infringed claims, the term appears in claim 1 (incorporated by reference in claim 2), claim 23 (incorporated by reference in claims 24 and 34), and claim 32 (incorporated by reference in claim 33). For example, claim 1 describes the methods of the invention as including the steps of:

encrypting a *block* of input plaintext data utilizing said key schedule;

...

encrypting a next *block* of input plaintext data utilizing said modified key schedule; and

repeating the step[] of ... encrypting utilizing the modified key schedule until the encrypting of *blocks* of plaintext data is completed.

(‘789 Patent, 11:26–38 (emphasis added).) The systems claims—24 and 34—state that a dynamic object key is modified “for each *block* of input data.” (*Id.* at 11:28–30 (emphasis added).)

The primary dispute between the parties at the time of claim construction regarding the claim term “block” was whether it required that all blocks of data in a given encryption be of the same length, or whether the block length could vary among blocks. Paone’s patent contained a preferred embodiment where the blocks have a “fixed length”; namely, each block contains 512 bits of information. The patent application explains that data blocks of 512 bits (64 bytes) each are encrypted, and that the input file will be padded with random bytes in order to produce a file with a length having a multiple of 512 bits. Nevertheless, Paone proposed at the claim construction stage that blocks are simply groups of data that are input into the block cipher. The Plaintiff further suggested that each block must have a fixed length, so that the length is “known to the cipher algorithm”, but that there is no limitation in the patent that each and every block

must be set to the same size. On the other hand, Microsoft contended that “block” meant a string of fixed length that applies to *all* strings of data, as defined by a block cipher.

Ultimately, the Court concurred with the Special Master and found that the term “block” as used in the claims and specification of the ‘789 patent means “a sequence of bits wherein that sequence has a fixed length that *does not vary from block-to-block*. The length of a block may be determined through selection in an encryption algorithm. Plaintext data having a length longer than the length of a block is divided into blocks of fixed length.” (Markman Order, at 43.) The Court explicitly accepted the Defendant’s interpretation that every block must be the same size and thus be “not subject to change or variation”, thereby rejecting Paone’s interpretation that “fixed” meant only “determined, established, or set”. (Special Masters Report, at 197.)

Therefore, in order for Microsoft to literally infringe the relevant four claims of the ‘789 patent, the blocks in TKIP’s encryption would necessarily need to have a fixed length, and that length cannot vary from block-to-block. The Defendant asserts that it is entitled to summary judgment on the ground that the blocks in TKIP do not necessarily have the same length from block-to-block, and consequently cannot literally infringe. In addition, the Defendant contends that it cannot be found to have infringed under the doctrine of equivalents, as posited by the Plaintiff. The Court will address each of these contentions in turn.

a. Literal Infringement

Literal infringement of a claim occurs when every limitation recited in the claim appears in the accused device, *i.e.*, when “the properly construed claim reads on the accused device exactly.” Amhil Enters., Ltd. v. Wawa, Inc., 81 F.3d 1554, 1562 (Fed. Cir. 1996).

As an initial matter, the Court notes that based upon the evidence in the record, there is no question of fact that TKIP allows in theory for two different scenarios: one in which MPDU blocks are all the same length, and one in which the MPDU blocks are not all the same length. The Plaintiff does not appear to dispute this fact in its opposition papers or in its Local 56.1 Statement. (See Pl. Resp. 56.1, ¶ 34.) Dr. Blaze, the Plaintiff's technical expert, testified to this explicitly in his deposition:

A. The MPDUs are blocks in TKIP, yes.

Q. And the length of an MPDU can vary from MPDU to MPDU, right?

A. Well, that depends. That depends on what's presented to TKIP. Whether an MPDU is the same size as a previous MPDU, whether they're all the same size or not, depends on a number of factors.

Q. But they can be different, then; the length of one MPDU can be different than the length of a subsequent MPDU?

A. There may be scenarios under which that's true, and, again, it depends on a variety of factors whether that will be true.

(Blaze 9/13/2011 Tr. at 124–25.)

For the moment, the Court sets aside the possible circumstance where the MPDUs are uniformly the same length. Instead, the Court will focus on the other potential (and presumably more likely) outcome: where the MPDUs are not uniformly the same size. Typically, this scenario is illustrated by an extremely high number of MPDUs that do have the same length, followed one single MPDU of a different length that represents the leftover bits that were not an exact multiple of the fragmentation threshold.

Paone argues that even in this situation—where one MPDU is not the same length as every other MPDU—this nevertheless meets the “blocks” limitation in the ‘789 patent. He contends that the MPDUs still comprise a “sequence of bits wherein that sequence has a fixed length that does not vary from block-to-block” and therefore literally meets the Court’s

construction of “block.” In short, Paone asserts that simply because TKIP could encrypt a single MPDU having a different length, i.e., representing the remaining bytes after the message is encrypted into identical length MPDUs, does not mean that TKIP does not contain the “blocks” element of the claim. Paone also attempts to argue that this potential additional MPDU that is non-conforming in length is merely an additional step that does not remove the accused technology from the literal scope of the claims. See CollegeNet, Inc. v. ApplyYourself, Inc., 418 F.3d 1225, 1235 (Fed. Cir. 2005) (“The transitional term ‘comprising’ . . . is inclusive or open-ended and does not exclude additional, unrecited elements or method steps. . . . A drafter uses the term ‘comprising’ to mean ‘I claim at least what follows and potentially more.’”) (internal quotations and citations omitted).

However, despite Paone’s arguments to the contrary, the Court finds that in this particular scenario, TKIP would not be infringing. This is because when the MPDUs are all of uniform length except for a single non-conforming MPDU, they are not fixed in a way that every block has precisely the same length, as the claims of the ‘789 patent have been construed. The Court’s claim construction requires that TKIP encrypted blocks *do not vary* from block to block. To the extent that several or merely one does, this simply does not literally infringe the patent. It may be a minor distinction, but it is one that patent law requires this Court to make. See, e.g., Atlas Powder Co. v. E.I. Du Pont de Nemours and Co., 588 F. Supp. 1455, 1471 (D.C. Tex. 1983) (“The differences between the two emulsifiers are slight, but they preclude a finding of literal infringement because ‘[m]inor modifications are . . . sufficient to avoid literal infringement.’”) (quoting Weidman Metal Masters v. Glass Master Corp., 623 F.2d 1024, 1026 (5th Cir. 1980)).

To the extent that Paone attempts again to assert that merely because the block length is set by the algorithm so that it is “fixed”, whether or not they are all of the same length, this

argument is once again rejected. Fixed in this case means constant. While it is unnecessary for the Court to restate its reasoning as to this notion, as it has been thoroughly explored in the Markman Order, the Court notes that it previously relied upon claim 15 when construing the claim term:

an additional piece of intrinsic evidence apparently not addressed by the parties or the Special Master supports the Special Master's conclusion. The patent provides in claim 15 for "[a] computer implemented method as defined in claim 2, wherein input plaintext is compressed . . . and padded . . . to produce a file with a length that is evenly divisible by the block length" (789 Patent, 13:36–41.) Claim 15's description of a file that is "evenly divisible by the block length" is a strong indication that, at least for that claim, block length does not vary from block to block. Otherwise, the term "evenly divisible" would have little meaning. The Court is cognizant of the bar on reading the additional limitations of claim 15 into claims 1 and 2. However, claim 15 does not define "block" separately from claims 1 and 2, but rather inherits this term from them as a dependent claim. Moreover, claim 15 says nothing about the "block" or "block length" until it is referenced in a way that implies that a block has a fixed length. While this evidence is not conclusive, the Court finds that it does support the Special Master's finding that the term "block" as used in claim 1 has a fixed length. See Phillips, 415 F.3d at 1314 ("Because claim terms are normally used consistently throughout the patent, the usage of a term in one claim can often illuminate the meaning of the same term in other claims.")

(Markman Order, at 42.) This same reasoning supports the conclusion here. The description of a file that is "evenly divisible by the block length" led the Court, in part, to conclude that the blocks must have a fixed length. If infringement could be found where there was uneven division, so that there was one block of unfixed length, this would contradict the rationale as set forth in this Court's previous decision.

The Court now addresses the more complicated issue, which is that there is a potential permutation of the TKIP encryption protocol in which the data file has a length that is precisely a multiple of the fragmentation constraint, so that MPDUs of exactly equal length are created. There does not appear to be a question of fact that this is conceivable and hence theoretically possible. Certainly there is some possibility, however remote, that the data file has a length that

is precisely a multiple of the fragmentation constraint, as the Plaintiff's expert posits there is. This then leads to two further lines of inquiry. First, is it enough for TKIP to merely be capable of infringement? While certainly this is not like other patent cases in which the technology is set to a non-infringing default, it is somewhat similar to those cases in which there is both a non-infringing and an infringing mode. However, unlike the precedent cited by the parties, whether the TKIP will result in blocks of equal length is not a result of a user implemented choice to operate one particular feature of the technology. Rather, in what appears to be a matter of first impression, whether TKIP results in infringement is entirely driven by randomness.

As for the question of whether capability alone is sufficient, it is highly relevant as to what types of claims are at issue. With regard to method claims 2 and 33, which speak to the steps of encryption, capability is determinative. "It is well established that a patent for a method or process is not infringed unless *all steps* or stages of the claimed process are utilized." Roberts Dairy Co. v. United States, 208 Ct. Cl. 830, 530 F.2d 1342, 1354 (1976) (citing Engelhard Industries, Inc. v. Research Instrumental Corp., 324 F.2d 347 (9th Cir. 1963), cert. denied, 377 U.S. 923, 84 S. Ct. 1220, 12 L. Ed. 2d 215 (1964) (emphasis added)). "A patented method is a series of steps, each of which must be performed for infringement to occur. It is not enough that a claimed step be 'capable' of being performed." Cybersettle, Inc. v. Nat'l Arbitration Forum, Inc., 243 Fed App'x 604, 606–07 (Fed. Cir. 2007) (citing Ormco Corp. v. Align Tech., Inc., 463 F.3d 1299, 1311 (Fed. Cir. 2006) (rejecting an argument that a claim requiring the replacement of appliances can be performed if the appliances are merely "capable of" being replaced)); see also NTP, 418 F.3d at 1318 ("[T]he use of a [claimed] process necessarily involves doing or performing each of the steps cited.").

Therefore, even if the Plaintiff could produce evidence to demonstrate that TKIP, in certain random implementations, would result in blocks of uniform equal length, that capability alone would not be sufficient for a finding of infringement, at least as to the methods claims. See Lucent Technologies, Inc. v. Gateway, Inc., 543 F.3d 710 (Fed. Cir. 2008) (finding that patent owner did not demonstrate infringement of audio coding methods claim by offering circumstantial evidence that proved at most that accused encoder possibly was capable of running; owner had to show, circumstantially or otherwise, that accused encoder actually had run and performed claimed method); E-Pass Technologies, Inc. v. 3Com Corp., 473 F.3d 1213, 1216 (Fed. Cir. 2007) (affirming district court’s grant of summary judgment of non-infringement where the patents were directed to a method of substituting an electronic multi-function card for a plurality of credit cards and the evidence provided by the patentee at best showed that customers were taught each step of the claimed method in isolation, yet failed to establish that all of the steps of the method had actually been performed in the prescribed order, so that it would be too speculative to conclude that any customer actually performed the claimed method).

On the other hand, as for the systems claims, capability may be sufficient. It is clear that “an accused product that sometimes, but not always, embodies a claimed method nonetheless infringes.” Bell Commc’ns Research, Inc. v. Vitalink Commc’ns Corp., 55 F.3d 615, 622–23 (Fed. Cir. 1995). Compare Netscape Commn’s Corp. v. ValueClick, Inc., 684 F. Supp. 2d 699, 772 (E.D. Va. 2010) (“these patent claims do not require a user to execute the claimed method; rather, the claimed computer systems are simply described as capable of performing the method, not as actually performing the method.”).

Nevertheless, even if capability to form equal length “blocks” is sufficient to constitute infringement, the Court need not reach the issue of whether capability is sufficient for systems

claims. This is because there is no genuine issue of material fact as to whether TKIP is even capable of meeting this limitation. The second relevant inquiry is, assuming that it is enough for TKIP to be merely capable of infringing in one particular randomly generated instance, is it sufficient that no real world implementations of this result have been demonstrated?

The Defendant contends that summary judgment is appropriate because MPDUs may only *theoretically* be of uniform length. Put another way, the question is whether Microsoft can be held liable because TKIP may hypothetically result in identical length blocks, although the Plaintiff has been unable to demonstrate at this stage of the litigation that real-world implementations of the protocol can infringe. Microsoft asserts that “[m]erely because, hypothetically, consecutive MPDUs may sometimes have the same length does not literally mean that MPDU length is ‘fixed.’” (Pl. Mem. at 8.) The Defendant’s logic is that the possibility that MPDUs may sometimes have the same length runs contradictory to this Court’s ruling that block lengths not be sometimes the same for two consecutive blocks, but rather be fixed for all blocks. In this regard, the Court agrees with Microsoft that merely presenting a theoretical possibility of capability is insufficient.

The ultimate burden of proving infringement rests with the patentee so that “an accused infringer seeking summary judgment of noninfringement may meet its initial responsibility either by providing evidence that would preclude a finding of infringement, or by showing that the evidence fails to establish a material issue of fact essential to the patentee’s case.” Novartis Corp. v. Ben Venue Labs., Inc., 271 F.3d 1043, 1046 (Fed. Cir. 2001). Here, Microsoft has not provided evidence that TKIP is incapable of resulting in equal block length, only that it is “theoretical” and presumably, highly unlikely. Cf. Harris Corp. v. Fed. Exp. Corp., 07 Civ. 1819, 2010 WL 129794, at *8 (M.D. Fla. 2010) (finding that because FedEx provided credible

evidence of noninfringement, the burden shifted to the Plaintiff to provide evidence raising actual doubt as to potential infringement). Nevertheless, the Defendant has shown that the evidence fails to establish a material issue of fact essential to the patentee's case.

At this point, the Plaintiff can only demonstrate that it is theoretically possible for the TKIP component to infringe his patent. “[H]e has shown no more than a theoretical possibility or ‘metaphysical doubt,’ which is insufficient to create a genuine issue of material fact. Jansen v. Rexall Sundown, Inc., 342 F.3d 1329, 1334 (Fed. Cir. 2003) (quoting Anderson, 477 U.S. at 261, 106 S. Ct. 2505); see Fire King Intern. LLC v. Tidel Eng’g, L.P., 613 F. Supp. 2d 836, 842 (N.D. Tex. 2009) (“none of this hypothetical evidence creates a genuine issue of material fact as to whether any of the Sentinel products installed at Speedy Stop stores are actually configured in such a way as to infringe Claim 1 and Claim 11 of the ‘252 Patent.”).

Essentially, TKIP is capable of acting in both a non-infringing mode and an infringing mode. When the data exceeds the fragmentation threshold in a way that is not precisely divisible by the applicable number, then the result is blocks that are not all of equal length. On the other hand, if the data exceeds the fragmentation threshold in a way that is precisely divisible by the applicable number so that equal MPDUs are created, then this implementation of TKIP may be infringing, as it would literally satisfy the claims limitation in the ‘789 patent. However, the Plaintiff has failed to demonstrate a single instance of the latter actually occurring in the real world, and thus has failed to prove that TKIP is actually capable of infringement. The Defendant asserts that Dr. Blaze was required to compare the ‘789 patent with the encryption actually utilized by Microsoft technology, look for real “blocks”, and see whether there are actual instances in the real world where the blocks are uniformly of equal length. The Court agrees.

Instead, Dr. Blaze supposedly reached his conclusions based solely upon the text of the TKIP protocol and thus merely pondered whether such a result would be theoretically possible. This is insufficient to show that TKIP is actually capable of meeting the “block” limitation, which is necessary to prove infringement. See Cybersettle, 243 Fed App’x at 609 (“In the course of their operation, the accused ANS 3 and ANS 1x systems would infringe (assuming all the other claim limitations were satisfied) only when they received multiple demands and multiple offers; proof that those systems were capable of receiving multiple demands and multiple offers is not proof that they ever performed the claimed methods”); WebZero, LLC v. ClicVU, Inc., No. 08 Civ. 0504, 2009 WL 8173102, at *4–5 (C.D. Cal. May 1, 2009) aff’d, 392 F. App’x 863 (Fed. Cir. 2010) (noting that “while it is true that an accused product that sometimes, but not always, embodies a claimed method nonetheless infringes, the accused infringing product must be capable of accomplishing the entire method of the claim”) (internal quotation marks omitted); see also Fire King Intern. LLC v. Tidel Eng’g, L.P., 613 F. Supp. 2d 836, 842 (N.D. Tex. 2009) (finding that even where marketing materials and expert testimony demonstrated that a system was capable of infringing the patent, “none of this hypothetical evidence creates a genuine issue of material fact as to whether any of the Sentinel products installed at Speedy Stop stores are actually configured in such a way as to infringe Claim 1 and Claim 11 of the ‘252 Patent.”). This is in stark contrast to a case like Vita-Mix Corp. v. Basic Holding, Inc., 581 F.3d 1317, 1326 (Fed. Cir. 2009), for example, where there was expert testimony that certain testing and demonstrations conducted by the defendant constituted infringement.

This is not to say that in every instance of patent infringement, the patentee must necessarily demonstrate specific instances of infringement. See ACCO Brands, Inc. v. ABA Locks Mfr. Co., 501 F. 3d 1307, 1313 (Fed. Cir. 2007) (holding that the patent owner must show

actual infringement, rather than just the capability to infringe). What the Court finds here is that Paone has failed to even satisfy the minimal burden that TKIP is in reality capable of meeting the “blocks” limitation. Cf. Fujitsu, 620 F.3d 1321 (“The claim language at issue in the ‘789 patent does not merely require the *capacity* to contain blocks; it explicitly requires “blocks” in the claim language. Unless the claim language only requires the capacity to perform a particular claim element, we have held that it is not enough to simply show that a product is capable of infringement; the patent owner must show evidence of specific instances of . . . infringement.”); Intel Corp. v. U.S. Int’l Trade Comm’n, 946 F.2d 821, 832 (Fed. Cir. 1991) (holding that the claim term “programmable selection means” only required that the infringing product be capable of infringing).

Thus, Paone has failed to establish a genuine issue of material fact regarding literal infringement with regard to TKIP.

As a final matter on this subject, the Court will address the recent case relied on by the Defendant entitled Fujitsu Ltd. v. Netgear Inc., 620 F.3d 1321, 1326–29 (Fed. Cir. 2010). According to Microsoft, comparing a standard like TKIP with the claims in a patent can only warrant recovery if the Plaintiff can demonstrate that real-world implementations of the standard *always* infringe. See Fujitsu Ltd., 620 F.3d at 1326–29. In Fujitsu, the issue was whether the Defendant’s implementation of a certain wireless networking protocol for sending and receiving messages between a base station, such as a wireless router, and a mobile station, such as a laptop, infringed the plaintiffs’ patent. In the summary judgment motion filed by the plaintiffs, they argued that by simply complying with the standard, Netgear necessarily infringed the asserted claims. In other words, the plaintiffs claimed that because the standard itself allowed for infringement, that was sufficient without any proof of real world implementations of that

standard. In its First Noninfringement Order, the district court held that any product that complied with certain sections of the standard infringed the asserted claims. But in its Second Noninfringement Opinion, the district court noted that the fragmentation option is disabled by default in the accused products and required Philips to show evidence of direct infringement by users turning on the fragmentation function.

On appeal, Netgear argued that it was legally incorrect to compare claims to a standard rather than directly to accused products. The Court held that:

a district court may rely on an industry standard in analyzing infringement. If a district court construes the claims and finds that the reach of the claims includes any device that practices a standard, then this can be sufficient for a finding of infringement. We agree that claims should be compared to the accused product to determine infringement. However, if an accused product operates in accordance with a standard, then comparing the claims to that standard is the same as comparing the claims to the accused product. . . . An accused infringer is free to either prove that the claims do not cover all implementations of the standard or to prove that it does not practice the standard.

Id. at 1327.

The Fujitsu court went on to state that public policy weighed in favor of this approach, because if a court determined that *all* implementations of a standard infringed the claims of a patent, then it would be a waste of judicial resources to separately analyze every accused product that undisputedly practices the standard. However, the court acknowledged “that in many instances, an industry standard does not provide the level of specificity required to establish that practicing that standard would always result in infringement.” Id. For instance, in the Fujitsu case, the standard was merely optional, so that standards compliance alone did not establish that the accused infringer chose to implement the infringing optional section. In those instances, the Federal Circuit stated that:

it is not sufficient for the patent owner to establish infringement by arguing that the product admittedly practices the standard, therefore it infringes. In these cases, the patent owner must compare the claims to the accused products or, if

appropriate, prove that the accused products implement any relevant optional sections of the standard. This should alleviate any concern about the use of standard compliance in assessing patent infringement. Only in the situation where a patent covers every possible implementation of a standard will it be enough to prove infringement by showing standard compliance.

Id. at 1328.

While the Fujitsu case is certainly relevant to the current analysis, it is not determinative and has several key distinctions. One key distinguishing factor is that the standard in Fujitsu had a default mode that was automatically non-infringing. Only if a user chose to implement an optional feature of the standard would infringement occur. Here, while there is arguably an infringing and non-infringing “mode” of TKIP, there is no default mode. More importantly, whether the standard can be implemented in an infringing way is not based upon a user actively utilizing an option in the protocol, but rather upon randomness and chance. Therefore, the Court agrees with Paone that the Fujitsu precedent is not applicable to the instant case for several reasons. Regardless, putting aside this authority, the Court still finds that the Plaintiff has failed to demonstrate a genuine issue of material fact in regard to whether the blocks limitation may be literally met. Even if the Court is to disregard Fujitsu and reject the notion that the Plaintiff must demonstrate that real-world implementations of the standard always infringe, he must at least produce evidence that the standard is actually capable of infringing. The Plaintiff has only presented Dr. Blaze’s conjecture that TKIP may theoretically result in MPDUs that are uniform length. This is plainly insufficient.

Moreover, the Plaintiff also argues that Fujitsu is not applicable to claims 24 and 34 because they are directed communications systems, rather than methods. See NTP, Inc. v. Research in Motion, Ltd., 418 F.3d 1282, 1318 (Fed. Cir. 2005) (“[T]he use of a process necessarily involves doing or performing each of the steps recited. This is unlike use of a system

as a whole. . . .”). In this vein, the Plaintiff urges the Court to disregard Fujitsu and instead follow the dictates of Finjan, Inc. v. Secure Computing Corp., 626 F.3d 1197, 1204 (Fed. Cir. 2010). In this latter case, the Federal Circuit held that because the claims at issue were “system” claims that did not require the performance of any method steps, “to infringe a claim that recites capability and not actual operation, an accused device need only be capable of operating in this described mode.” However, in the Finjan case, the claim language only required the *capacity* to perform a particular claim element. In particular, “Finjan’s apparatus claims [did] not require that the proactive scanning software be configured in a particular way to infringe—only that it be *programmed* for performing the claimed steps.” Id. at 1204 (emphasis added). That is plainly not the system claims at issue here. Rather, 24 and 34 both contain language that requires “blocks” of data. It does say that the data need only be capable of being divided into equal block units. Thus, the Plaintiff’s reliance on Finjan is misplaced.

In sum, the Court grants the Defendant’s motion for summary judgment as to the literal infringement of the four claims, in light of the Plaintiff’s failure to produce evidence that TKIP even has the capability to meet the blocks limitation.

b. Doctrine of Equivalents

In light of the Court’s finding that there are no genuine issues of material fact as to whether TKIP can literally infringe any of the relevant four patent claims as a matter of law based upon the “block” limitation, the Court must proceed to assess whether TKIP can nevertheless be found to infringe the four patent claims because it meets the “block” limitation under the doctrine of equivalents.

Infringement under the doctrine of equivalents requires that the accused method or system contain each limitation of the claim *or its equivalent*. See Warner-Jenkinson Co. v.

Hilton Davis Chem. Co., 520 U.S. 17, 40, 117 S. Ct. 1040, 137 L. Ed. 2d 146 (1997) (noting that because each limitation contained in a patent claim is material to defining the scope of the patented invention, a doctrine of equivalents analysis must be applied to individual claim limitations, not to the invention as a whole). An element in the accused method or system is equivalent to a claim limitation if the differences between the two are “insubstantial” to one of ordinary skill in the art. See id.

Paone argues that TKIP may infringe the ‘789 patent claims because it meets the “block” limitation under the doctrine of equivalents. In particular, the Plaintiff contends that “one of ordinary skill in the art would consider the differences between a system or method that encrypts blocks that vary in length in some circumstances, and one that does not, to be insubstantial—it is merely a design choice.” (Pl. Mem. at 7.) Moreover, the Plaintiff maintains that although the patent discloses an embodiment that encrypts blocks that do not vary in length, that aspect is not the inventive focus of the asserted claims, which do not address block length at all. Finally, Paone asserts that in scenarios when the TKIP’s “blocks” vary in length from one to the next, those MPDUs nonetheless perform substantially the same function—dividing a large amount of data into manageable portions to facilitate block cipher encryption; in substantially the same way—by grouping data for encryption by a block cipher; to achieve substantially the same result—the conversion by a block cipher of unencrypted plaintext blocks of data into encrypted ciphertext blocks of data—as the “blocks” of the asserted claims.

Microsoft makes several arguments as to why the application of the doctrine of equivalents should be precluded. First, the Defendant relies upon an exception to the doctrine of equivalents; namely, the claim-vitiation doctrine. This doctrine provides that “an element of an accused product or process is not, as a matter of law, equivalent to a limitation of the claimed

invention if such a finding would entirely vitiate the limitation.” Freedman Seating Co. v. Am. Seating Co., 420 F.3d 1350, 1358 (Fed. Cir. 2005) (citing Warner–Jenkinson Co. v. Hilton Davis Chem. Co., 520 U.S. 17, 29, 117 S. Ct. 1040, 137 L. Ed. 2d 146 (1997)). “There is no set formula for determining whether a finding of equivalence would vitiate a claim limitation. . . . Rather, courts must consider the totality of the circumstances of each case and determine whether the alleged equivalent can be fairly characterized as an insubstantial change from the claimed subject matter without rendering the pertinent limitation meaningless.” Id. at 1359. “Claim vitiation applies when there is a ‘clear, substantial difference or a difference in kind’ between the claim limitation and the accused product.” Trading Techs. Int’l, Inc. v. eSpeed, Inc., 595 F.3d 1340, 1355 (Fed. Cir. 2010) (quoting Freedman, 420 F.3d at 1360). “It does not apply when there is a ‘subtle difference in degree.’” Id.

The purpose of the doctrine is to not allow the recapture of subject matter excluded by a deliberate claim-drafting decision. Planet Bingo, LLC v. GameTech Int’l, Inc., 472 F.3d 1338, 1344 (Fed. Cir. 2006). The “doctrine of equivalence cannot be used to erase meaningful structural and functional limitations of the claim on which the public is entitled to rely in avoiding infringement.” Conopco, Inc. v. May Dep’t Stores Co., 46 F.3d 1556, 1562 (Fed. Cir. 1994) (internal citation omitted); see, e.g., Tronzo v. Biomet, Inc., 156 F.3d 1154, 1160 (Fed. Cir. 1998) (noting that the finding of all shapes to be equivalent structures would entirely vitiate the limitation requiring a “generally conical shape”). “If [the doctrine] were otherwise, then claims would be reduced to functional abstracts, devoid of meaningful structural limitations on which the public could rely.” Safe Prods., Inc. v. Devon Indus., Inc., 126 F.3d 1420, 1424-25 (Fed. Cir. 1997).

Ultimately, the Court finds that there are questions of fact that preclude the granting of the Defendant's motion for summary judgment as to whether the MPDUs may meet the "blocks" limitation under the doctrine of equivalents.

Many of the arguments Paone raises in this regard have more to do with a rehashing of its arguments as to whether the claim term "blocks" should be construed to require the MPDUs to have equal length. However, as set forth in the previous Markman Order, this issue has already been resolved. The Court explicitly rejected Paone's contentions and found that the term "block" meant a sequence of bits wherein that sequence has a fixed length that *does not vary from block-to-block*. Thus, the Court finds that Paone's arguments miss the mark and agrees that Paone failed to argue that the hypothetical "remainder" situations infringe under the doctrine of equivalents. Nevertheless, there is enough in the record for the Court to find, even without an argument to this effect, that the hypothetical remainder situation may infringe under the doctrine of equivalents.

At this stage of the litigation, the Court cannot say that the differences between the "block" limitation and the MPDUs are insubstantial as a matter of law. Whether the difference is a "subtle difference in degree" rather than a "difference in kind" is a factual question, and one that is more properly left to the jury. Trading Techs., 595 F.3d at 1355. First, Microsoft admits that there are often times when the TKIP encryption processes results in blocks that are of fixed length that do not vary from block-to-block, except for one single MPDU having a different length, i.e., representing the remaining bytes after the message is encrypted into identical-length MPDUs (See Blaze Decl. at ¶ 33.) Thus, it is conceivable that a jury could find that where millions of blocks are precisely the same size with only one deviation, this is a subtle difference in degree rather than a difference in kind. This finding would depend on a large number of

factual disputes, such as, how often such a result occurs, what percentage of the encryption is not uniform, and the absolute size of the nonconforming block, relative to the size of the computer document or image as a whole. Contrary to Microsoft's assertions, this could result in a finding that the difference between the two is insubstantial.

Second, even in the situation where the data message does not reach the applicable fragmentation threshold so that no fragmentation will occur (which results in MPDUs that vary in length from one to the next), there is a question of fact as to the doctrine of equivalents. In other words, the differences between a system or method that encrypts in uniform length blocks versus one that encrypts in varying length blocks, may be a design choice that the trier of fact ultimately finds to be insubstantial. Certainly, Paone contends that the TKIP blocks perform substantially the same function, in substantially the same way, to achieve substantially the same result, as the "blocks" in the claimed subject matter. (Blaze Suppl. Expert Report at 51.) The Plaintiff has presented sufficient evidence in this regard to at least present this dispute to a jury.

For example, the Plaintiff's expert Dr. Blaze has testified that "a person of ordinary skill in the art would consider the differences between TKIP blocks and the claimed subject matter to be insubstantial." (Blaze Suppl. Expert Report at 49-51.) Blaze contends that in both the TKIP encryption process and in the '789 patent, a group of data is split up into smaller sequences so that it may be processed together, in part for security reasons. To encrypt multiple smaller blocks rather than one large block likely results in a more effective encryption process. Thus, whether the multiple smaller blocks are all of equal length, according to the Blaze, is inconsequential. To support this contention, the Plaintiff also points to prior art that was well known in the field since before Paone filed his patent, in which the block ciphers encrypt blocks

of variable size. See, e.g., Article: Two Practical and Provably Secure Block Ciphers: BEAR and LION, R. Anderson et al., at 1 (“[O]ur constructions allow arbitrary sized blocks to be enciphered . . .”).

Therefore, the Court will not impose vitiating to prevent the jury from reaching the factual question of Paone’s equivalence theory. Paone has presented evidence that that one block having a different length than every other single block does not change the efficacy of the method or system. A jury may disagree with Paone’s theory. The Court, however, is unwilling to take that decision out of the jury’s hands. See, e.g., Depuy Spine, Inc. v. Medtronic Sofamor Danek, Inc., 469 F.3d 1005, 1020 (Fed. Cir. 2006) (rejecting defendant’s argument of claim vitiating because of the finding “that a question of fact exists as to whether the difference between the ‘spherical-shaped’ limitation and the alleged equivalent is substantial.”). The Court does not find that Paone’s arguments would necessarily render the “blocks” claim limitation meaningless, as construed by this Court. The relevant inquiry is, in light of the fact that the Court has found that the blocks must all be of equal length, would it eviscerate the limitation to find that an encryption in which every single block but one is of equal length, infringes under the doctrine of equivalents. The Plaintiff has presented the Court with sufficient evidence to demonstrate that there is at least a question of fact as to this issue.

For this reason, the Court declines to apply the doctrine of vitiating and finds that the question of whether the doctrine of equivalents is applicable to the MPDUs in the TKIP component is properly left to the jury. Thus, the Defendant’s motion for summary judgment on this ground it is denied.

2. “Block Cipher”

All four asserted claims also require that the encryption be carried out in a “block cipher”, which the Court has construed as “a cipher that encrypts data in blocks.” (Markman Order at 61.) Thus, the Defendant argues that because TKIP does not encrypt data in “blocks”, as the term has been defined by the Court, then TKIP cannot utilize a “block cipher.” For all of the reasons stated above in connection with “block”, the Defendant’s motion with regard to literal infringement is granted but with regard to the doctrine of equivalents it is denied.

D. As to Whether TKIP and Bitlocker Employ an “Object Key” Having Data and Methods that Modify That Data

The next issue of contention concerns “object key”, an element that all four claims of the ‘789 patent require.

1. The Previous Claim Construction

The parties have always agreed, at a minimum, that the term “object key” refers to a key that is used to encrypt data, and they also agreed that this term did not have a standard meaning within the fields of computer science or cryptography prior to the filing of the ‘789 patent. The use of object key throughout the patent is typified by its use in claim 1, which states that one of the methods of the invention is “creating at least one object key in a block cipher, the at least one object key comprising data and methods that operate on said data.” (‘789 Patent, 11:19–21; see also, e.g., id. at 18:12–14.) The Special Master construed “object key” as follows:

In the field of cryptography, encryption and decryption are controlled by keys. The inventor coined the term “object key” to refer to a “first” encryption key which is distinct from a “second” encryption key which the inventor called a “random session object key,” also a coined term. Both are “dynamic,” i.e., both are modified from an initial static state, however it is unnecessary to expressly add that to the construction of “object key” because that modification is required by other claim language. The term “object key” per se as used in the claims and specification, simply means a “first encryption key.”

The Special Master further ruled that the “object key” need neither be “secret” nor defined in object-oriented programming and self-contained.

However, the Court did not adopt this exact finding. Rather, the Court read the claims in view of the specification and concluded that the “object key” needed to be secret, not only because the object key is referred to as “secret” in several places in the specification and the patent title, but the “Background of the Invention” portion of the specification focuses almost exclusively on the importance in cryptography of maintaining the secrecy of an encrypted message. Importantly, it appeared to the Court that a non-secret object key would undermine the usefulness of the entire invention.

With regard to the more central argument over the meaning of “object key”—namely, whether the object key must be defined in what computer scientists call “object-oriented programming”—this presented a more thorny issue at the claim construction stage. It is worthwhile here to repeat some relevant background information.

Object-oriented programming (“OOP”) is not easy to define, and a recently published book for beginning computer programmers notes that “OOP is difficult to summarize because it doesn’t represent a single concept . . . and even experts are unable to agree on a common definition.” Richard Mansfield, Programming: A Beginner’s Guide, p. 264 (McGraw Hill 2009). Even the programming language C++, one of the most widely used object-oriented programming languages, is viewed by some as “not a pure OOP language.” Namir C. Shammas, Foundations of C++ and Object-oriented Programming, p. 10 (IDG Books Worldwide, Inc. 1995).

However, on a very basic level, object-oriented programming is usually defined in contrast to a method of computer programming that preceded OOP, which is called “structured” (or “functional”) programming. Structured programming is a method of giving computers

instructions in “reusable subroutines and functions . . . [which] enjoy[] a certain level of autonomy”. Shamma at 7. Thus, structured programs can define a complicated task for a computer to do—say, calculate the square root of a number—and define that task in a “subroutine”. That subroutine contains all of the steps necessary to do the task, so the task can be repeated over and over by using the subroutine, without forcing the programmer to re-write the individual steps each time.

However, structured programming generally stores data in ways that permit data to be manipulated broadly, which “[i]n large-scale software projects . . . can lead to chaos.” See Shamma at 7, 9. In other words, this means that in structured programming, the data used by a given subroutine is shared with and can be modified by other subroutines. One result of this is that when data becomes damaged or lost, it can be difficult to determine the source of the problem because so many different subroutines have access to the data.

Object-oriented programming seeks to solve this problem. While an object-oriented program still defines complex tasks into subroutines, it allows only the piece of the program that uses each piece of data most directly to have immediate access to that data. Thus, in object-oriented programming, instead of using subroutines as the building blocks of a program, programmers use “objects”. “Objects” consist of both the instructions for various subroutines and the data needed to carry out those instructions. Thus, whereas the data in a structured program might be very loosely thought of as being in a shared pool, the data in an object-oriented program is encapsulated into pockets, accessible only by the “object” that will use that data directly. Any other part of the program that wishes to change an object’s data must go through the object to do so. Objects are thus described variously as having “attributes and [] methods that alter these attributes,” Shamma at 15; “both data . . . as well as code that processes

that data,” Mansfield at 264; or “data and the operations that manipulate the data,” Matt Weisfeld, The Object-Oriented Thought Process, p. 9 (Sams Publishing 2004). (The Court notes that while some of these sources post-date the filing of the ‘789 patent, there is no indication that the understanding of this basic theory in object-oriented programming has changed since the ‘789 patent was filed in 1997.) Generally, a programming language is viewed as either a “structured programming language” or an “object-oriented programming language”—although there are some exceptions to this rule.

In the Markman Order, the Court ultimately agreed with the Special Master that an “object key” need not be executed in an object-oriented program and hence need not be self-contained. Paone’s implicit reference to object-oriented programming did not limit the technical execution of the invention to an object-oriented programming language. (See Markman Order at 25–26 (“The claim language and specification say nothing that indicates that the invention must be practiced using a computer programming language that prevents other parts of the program from modifying the object key’s data, such as an object-oriented programming language would do. Indeed, the patent does not even suggest that this would benefit the invention.”).)

However, the Court respectfully disagreed with the Special Master’s determination that the word “object” in the claim term connotes no special meaning in the context of the patent. Rather, the Court construed an “object key” as an encryption key that is not available to the general public, and which is composed of both (1) key data and (2) methods that modify that key data. Specifically, the Court held that “object key” refers to an element of the invention that functions analogously to an object in an object-oriented program. Just like a bona fide object in an object-oriented program, the “object key” contains both data—the key data that will inform the operation of the cipher—and the methods that operate on that data. More importantly, just as

in an object-oriented program, the methods contained in the object key are the only methods that operate on the key data.

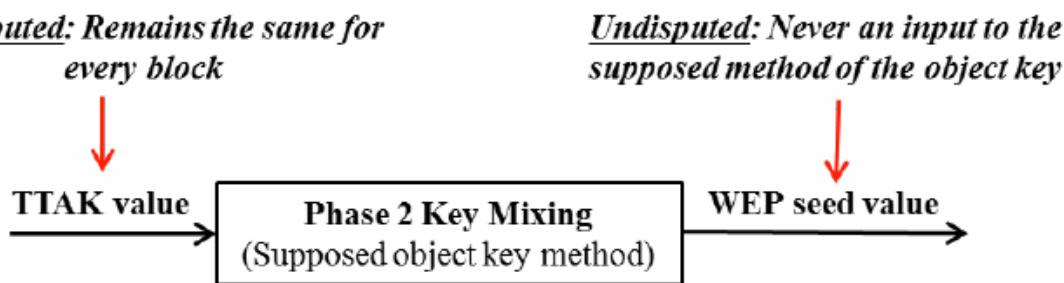
2. As to the TKIP Technology

In short, the Defendant argues that TKIP does not encrypt “object keys”, as that phrase has been defined by the Court. According to Microsoft, the Plaintiff’s expert Dr. Blaze admitted that (1) certain data he identifies as object key data is never modified; and (2) other data he identifies as object key data is modified, but not by the alleged methods of those object keys. Thus, according to the Defendant, the requirement that the object key data be modified and that it be modified by the methods contained in that object key are not satisfied and thus TKIP cannot be found to infringe the ‘789 patent.

In order to fully assess the Defendant’s claims in this regard, the Court must endeavor to explore the technology in TKIP that is alleged to be equivalent to the “object keys” in Paone’s patent. Dr. Blaze identified two values in TKIP as key data. First, he alleged that the initial state of the data of the “object key” used in the TKIP encryption—the TKIP-mixed transmit address and key (“TTAK”)—is key data. Second, he identified the per-frame keys in TKIP encryption, or “WEP seeds”, which are produced as a result of the methods of the “object key”, as key data. The latter are, in essence, subsequent modified versions of the initial “object key”. In addition, the Plaintiff’s expert identified one method, referred to as “phase 2 key mixing,” that he alleges is the method of TKIP’s “object key”. The Defendant provided a simplified diagram showing the relationship of these three elements:



However, according to the Defendant, Dr. Blaze admitted that the first data value, TTAK, is not modified by any method because it remains unchanged for many MPDUs or “blocks”. More specifically, under the TKIP encryption process, the TTAK value remains unchanged for 65,000 MPDUs, at which point a new TTAK is calculated. (See IEEE 802.11 standard.) The Defendant further emphasizes that while the WEP seed value is different for each encrypted MPDU, it is not modified by the “phase 2 key mixing” method. Rather, the WEP value changes due to another value that is an input to the phase 2 key mixing function. Put simply, Microsoft alleges that this specific object key data is modified, but not pursuant to the object key’s method. Thus, according to the Defendant, the following diagram is an accurate description of the undisputed facts in this case:



To summarize, Microsoft contends that it is undisputed that the TTAK value—one form of object key data—remains unchanged for many blocks, and the WEP seed data value—another form of object key data—changes from block to block, but is not changed by the “method” of the TKIP object key.

On the other hand, these contentions are based solely upon interpretations of Dr. Blaze’s expert testimony and reports. Moreover, the above pictorial representations of TKIP are highly simplified. In Dr. Blaze’s declaration, he has provided a more comprehensive diagram of how TKIP actually operates:

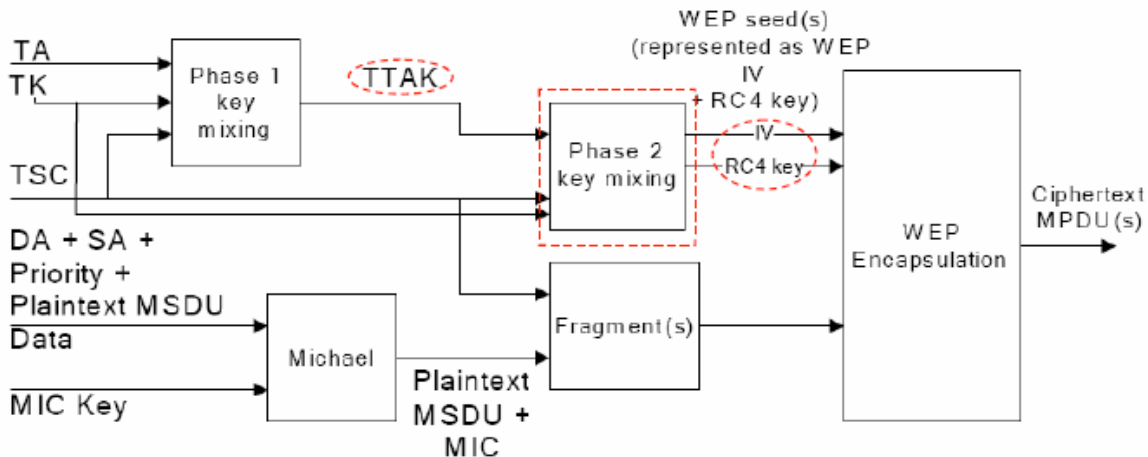


Figure 8-4—TKIP encapsulation block diagram

According to the Plaintiff, TKIP literally meets the “object key” limitation of the asserted claims, as construed by the Court. First, the TTKAK is determined by the phase-1 mixing function, which utilizes as inputs (1) a transmitted address value (TA), (2) a temporal key value (TK) (the secret key), and (3) a TKIP sequence counter value (TSC). The TTKAK is “not available to the general public” as required by the Court, which the Defendant does not dispute. More importantly, the Plaintiff also contends that TKIP’s “object key” comprises methods that modify the TKIP. For instance, the phase 2 mixing function is a method that modifies the TTKAK, and thereby produces modified versions of the “object key” data—otherwise known as “WEP seeds” or per-frame seeds—for each block of input plaintext data. (See Blaze Decl. at ¶¶ 52–54.) Thus, with regard to the fact that the same TTKAK value is used for multiple MPDUs, the Plaintiff argues that TTKAK is nevertheless modified because the phase 2 mixing function modifies it in every block to form the WEP seed. With regard to the fact that the “WEP seeds” are not input into the phase-2 mixing function, the Plaintiff essentially contends that this is irrelevant because there is “key data”—the TTKAK—and the phase 2 key mixing is modifying that key data.

With regard to the first alleged element of key data—the TTAK—the Defendant maintains that it is merely an “input” to be used in the phase 2 mixing process and consequently TTAK is not modified, but rather acts only as a building block to create the WEP seed. On the other hand, the Plaintiff argues that TTAK, by being used in a mixing function that creates the WEP seed, is in fact being modified for each block or MPDU. An analogy in this regard may be helpful. The Defendant views TTAK as one brick, which is utilized in the building process to create a home. Seen in this light, the brick is not “modified”—it remains a brick, yet it is used as a building block to create something new. The Plaintiff views TTAK as basic white paint, which is then mixed with other colors, such as red, green, and blue, to create a new brown color. If looked at from this perspective, the white paint has clearly been “modified” to create a new color of paint.

As a preliminary matter, the Court declines to adopt a narrower definition of “modify” than was set out in the Markman Order. Any notion by the Plaintiff that modify must mean that a mutation occurs, is disregarded. Rather, the Court has previously construed the modification of object key data as follows:

In the opinion of the Court, this language implies that the object key’s data—whether in original or already modified form—is an input into the object key’s new state. (See, e.g., “Modify”, Merriam Webster Online Dictionary, available online at <<http://www.merriam-webster.com/dictionary/modify>> (“3a: to make minor changes in; 3b: to make basic or fundamental changes in often to give a new orientation to or to serve a new end <the wing of a bird is an arm modified for flying>”) (accessed February 2, 2011)). . . . There is no limitation on including other inputs into the modification algorithm—and indeed, claim 1 and claim 32 require additional inputs—but the object key’s own data is a required input to “modify” the object key.

(Markman Order at 37.) Thus, the Court’s previous interpretation of the term “modify” relied upon the plain meaning of the term, and specifically stated that the language implied that the object key’s data is an *input* into the object key’s new state.

Here, the evidence put forth by the parties clearly indicates that TTAK is, at a minimum, an input into the phase 2 mixing function, and thus is utilized in the process in order to create the resulting data—the WEP seed. The relevant protocol states that phase 2 comprises three steps: (1) Step 1 makes a copy of TTAK and brings in the TSC; (2) Step 2 is a 96-bit bijective mixing, employing an S-box; and (3) Step 3 brings in the last of the temporal key TK bits and assigns the 24-bit WEP IV value. (IEEE Standard 802.11, at 178.) A figure that lays out each step for the phase 2 mixing function also appears in the protocol:

```

Input: intermediate key TTAK0... TTAK4, TK, and TKIP sequence counter TSC
Output: WEP Seed WEPSeed0... WEPSeed15
PHASE2-KEY-MIXING(TTAK0... TTAK4, TK0... TK15, TSC0... TSC5)
  PHASE2_STEP1:
    PPK0 ← TTAK0
    PPK1 ← TTAK1
    PPK2 ← TTAK2
    PPK3 ← TTAK3
    PPK4 ← TTAK4
    PPK5 ← TTAK4 + Mk16(TSC1, TSC0)
  PHASE2_STEP2:
    PPK0 ← PPK0 + S[PPK5 ⊕ Mk16(TK1, TK0)]
    PPK1 ← PPK1 + S[PPK0 ⊕ Mk16(TK3, TK2)]
    PPK2 ← PPK2 + S[PPK1 ⊕ Mk16(TK5, TK4)]
    PPK3 ← PPK3 + S[PPK2 ⊕ Mk16(TK7, TK6)]
    PPK4 ← PPK4 + S[PPK3 ⊕ Mk16(TK9, TK8)]
    PPK5 ← PPK5 + S[PPK4 ⊕ Mk16(TK11, TK10)]
    PPK0 ← PPK0 + RotR1(PPK5 ⊕ Mk16(TK13, TK12))
    PPK1 ← PPK1 + RotR1(PPK0 ⊕ Mk16(TK15, TK14))
    PPK2 ← PPK2 + RotR1(PPK1)
    PPK3 ← PPK3 + RotR1(PPK2)
    PPK4 ← PPK4 + RotR1(PPK3)
    PPK5 ← PPK5 + RotR1(PPK4)
  PHASE2_STEP3:
    WEPSeed0 ← TSC1
    WEPSeed1 ← (TSC1 | 0x20) & 0x7F
    WEPSeed2 ← TSC0
    WEPSeed3 ← Lo8((PPK5 ⊕ Mk16(TK1, TK0)) >> 1)
    for i = 0 to 5
      WEPSeed4+(2·i) ← Lo8(PPKi)
      WEPSeed5+(2·i) ← Hi8(PPKi)
    end
  return WEPSeed0... WEPSeed15

```

Figure 8-14—Phase 2 key mixing

Thus, it appears to the Court that TTAK could inform the operation of the cipher, and the method of the phase 2 mixing function may operate on the TTAK. For instance, the process plainly specifies that TTAK is subject to bijective mixing, employing an S-box, and the figure demonstrates that several algorithms function on the TTAK data. Moreover, Dr. Blaze testified that “I’m identifying—to the extent that the TTAK is the data of the object key, I believe that the data of the object key can be considered to be TTAK initially, and then as we operate that data, the data of the object key is the WEP seed.” (Blaze 9/3/2011 Tr. at 83.)

Moreover, TKIP clearly requires that TTAK is copied for each block before the data is used or manipulated. The Defendant seems to argue that because TTAK is copied for each block and consequently utilized in the same format for every block, that it cannot be deemed “modified”. However, the Court understands this fact as actually undercutting the Defendant’s position. It is precisely because TTAK is copied for each block for use in the phase 2 mixing function, that there appears to be evidence to support the finding that TTAK is modified. If it were not, then there would be no need to copy the data; instead, the same master copy could be utilized for each block. In addition, the Defendant argues that “Dr. Blaze admitted that the TTAK value is not modified by any method because it remains unchanged for many, many MPDUs.” (Pl. Mem. at 14.) However, that is a mischaracterization of Dr. Blaze’s testimony. When asked, “So if the implementation of the TKIP recalculates the TTAK for every MPDU in a session, there will be many, many, many MPDUs with the same TTAK”, Dr. Blaze testified “With the same resulting calculation, that’s right”. (Blaze 9/3/2011 Tr. at 84:17–22.) However, that only means that the TTAK input might be the same from block to block. It does not mean, as Microsoft attempts to argue, that the input cannot be modified *within* a block. In fact, Dr. Blaze further responded to that line of questioning by stating “because the method is the Phase 2

key mixing, which will modify it, producing a different WEP seed for each packet.” (Id. at 85:3–5.)

Finally, to the extent that the phase 2 key mixing combines TTAK with another input—TSC—for use in the mixing function, does not preclude a possible finding that TTKIP infringes the “object key” claim limitation. The Court specifically noted in the Markman Order that “[t]here is no limitation on including other inputs into the modification algorithm—and indeed, claim 1 and claim 32 require additional inputs—but the object key’s own data is a required input to ‘modify’ the object key.” (Markman Order at 37.)

Other than using the terminology of “input” rather than “change” or “modify”, the Defendant has not presented the Court with any evidence or otherwise pointed to sufficient deficiencies in the Plaintiff’s evidence to demonstrate that there is an absence of a genuine issue of material fact in this regard. Rather, the Court finds that there is a dispute as to what functionally happens to the TTAK data when it is processed in the phase 2 mixing function. The Court is merely provided with what is essentially an algorithm, and the parties’ competing explanations over what that means in plain terms. The Plaintiff’s expert claims that the data is, under the Court’s definition, “modified”, in that it is put through the mixer function and utilized in a way to create something new—the WEP seed. The Defendant does not provide its own expert report or testimony, but rather attempts to poke holes in the Plaintiff’s case by pointing to gaps in Blaze’s opinion and grasping at whatever arguable admissions he makes. What the Court does not have before it at this stage of the proceedings is enough evidence to make this determination. All the Court has is Blaze’s testimony and the TKIP protocol language itself. As indicated above, the party seeking summary judgment has the initial burden of showing that no

issue of material fact exists. Thus, Microsoft has failed at this stage to meet its burden to demonstrate an absence of genuine issue of material fact.

Paone's expert testified to his understanding that TTAK is modified, and the Court sees no reason at this point to disregard that opinion. Based on the highly complicated and sophisticated nature of this case, it is simply not a situation where the Court can rely on Microsoft's legal arguments and positions in their memorandum of law to resolve the issue of how this encryption process functions. As stated long ago by the Fifth Circuit:

Where by the nature of things, the moving papers themselves demonstrate that there is inherent in the problem a factual controversy then, while it is certainly the part of prudence for the advocate to file one, a categorical counter-affidavit is not essential. Bruce Construction Corp. v. United States, 5 Cir., 242 F.2d 873; Whitaker v. Coleman, 5 Cir., 115 F.2d 305.

How thoroughly the function of . . . purporting to state facts in support of motions for summary judgment was confused with the laudable function of the advocate and the hope that counsel's observations could take the place of testimony normally offered is further evident. [The Plaintiff] undertakes to compare, claim by claim, step by step, each of the two operations. From this he then concludes as a fact that for which expert witnesses are normally used. Analyzing it step by step substantially . . . he asserts that the patent claims 'read on' each of these ingredients. We may assume that an expert patent counsel or engineer might be permitted in proper circumstances to give such evidence, but in an appliance so remote from the common experience of Judges, it will be an unusual case in which this 'fact' is compulsorily established as a matter to law.

Inglett & Co. v. Everglades Fertilizer Co., 255 F.2d 342, 349 (5th Cir. 1958).

The same reasoning equally applies here. Microsoft's counsel has set out to explain facts, fueled by certain statements made by the Plaintiff's expert, and then compares claim by claim, step by step, each of the relevant technologies. From this, Microsoft concludes that the TTAK is not modified. However, this computer encryption technology is simply too remote a common experience of judges, so that the "fact" of non-modification cannot be established as a matter of law. "[W]ithout impugning any improper professional motive to this obviously able counsel, [the Court] doubt[s] that the disposition of patent cases is furthered by counsel being the

personal vehicle by which the ‘undisputed’ facts are put before the Court. [The Court] consider[s] it a tribute to the high calling of advocacy to say that we think it an unnatural, if not virtually impossible, task for counsel, in his own case, to drop his garments of advocacy and take on the somber garb of an objective fact-stater.” Id. “Unsubstantiated attorney argument regarding the meaning of technical evidence is no substitute for competent, substantiated expert testimony.” Invitrogen Corp. v. Clontech Labs., Inc., 429 F.3d 1052, 1068 (Fed. Cir. 2005); see Howmedica Osteonics Corp. v. Tranquil Prospects, Ltd., 482 F. Supp. 2d 1045, 1065 (N.D. Ind. 2007) (“Such wholly conclusory assertions on a legal issue cannot carry Tranquil’s burden on summary judgment) (citing Biotec Biologische Naturverpackungen GmbH & Co. KG v. Biocorp, Inc., 249 F.3d 1341, 1353 (Fed. Cir. 2001)).

In sum, the Defendant has not met its burden in demonstrating that there are no genuine issues of material fact pertaining to whether TKIP contains an “object key”. The technical questions concerning the concept of encryption processes and algorithms, specifically what happens to the TTAK data during the phase-2 mixing function, cannot be resolved as a matter of law on the record presently before the Court. By reason of the extremely technical nature of the fact question involved, the Court feels as did the Court of Appeals in the Second Circuit: ‘Were we skilled in the art it might be simple to determine whether there was any ‘genuine issue’ as to any material fact * * * but we lack that special knowledge which would permit us to read the patents so understandingly and this record is barren of proof to enable us to do so.’ Bridgeport Brass Co. v. Bostwick Labs., Inc., 181 F.2d 315–19 (2d Cir. 1950); see, e.g., Aconstiflex Corp. v. Owens–Corning Fiberglas Corp., 572 F. Supp. 936, 937 (N.D. Ill. 1983) (“[A] patent case is not ripe for summary judgment on the issues of validity or enforceability where the technical aspects are not readily comprehensible by one unskilled in the art, where the record is inadequate

to decide the issue, where there is a need for expert testimony, where the expert testimony submitted is conflicting, or where there is some other genuine issue of credibility on a motion for summary judgment.”); see also May v. Carriage, Inc., 688 F. Supp. 408, 413 (N.D. Ind. 1988) (“Many patent cases are virtually impossible to decide on summary judgment due to the need for expert testimony relating to technical or scientific matters beyond the court’s expertise.”). Cf. Shemitz v. Deere & Co., Inc., 623 F.2d 1180, 1184 (7th Cir. 1980) (“Rule 56 applies to patent cases and is used where, as here, the structure and mode of operation of the invention described and claimed in the patent may be readily comprehended by the court without need for technical explanation by expert witnesses and in such circumstances; if said invention is found invalid because of the prior art, then summary judgment is proper.”); G.B. Lewis Company v. Gould Products, Inc., 436 F.2d 1176 (2d Cir. 1971) (finding summary judgment appropriate in design patent cases because expert testimony would not aid the Court in evaluating non-technical matters of visual impression).

With regard to the WEP seed, Microsoft also contends that the existence of an element that the Plaintiff labels as “key data”, but that is not modified by the method contained in the object key, similarly warrants a finding of summary judgment as to the TKIP technology. The Court agrees with the Defendant that the evidence clearly demonstrates that there is no question of fact that WEP is not modified by the only process identified as being part of the “object key”—the phase 2 mixing. In fact, the Plaintiff admits that the WEP seed is an output of the process, and logically, cannot be modified by the process that created it. Thus, the Court finds, as a matter of law, that the WEP seed cannot be object key data under the terms of the patent.

However, this finding does not alter the Court’s finding as to the denial of summary judgment on this issue. The patent only requires the “at least one [dynamic] object key”. Thus,

even if the Court were to disregard the existence of the WEP seed, the TKIP technology still possibly meets the limitations of the patent claims because the TTAK and phase 2 mixing function might nevertheless qualify as an object key, of which there must be “at least one.” As long as TTAK may possibly meet the claims limitations, the WEP seed is irrelevant. For this reason, the Defendant’s arguments in this regard are denied.

3. As to the BitLocker Technology

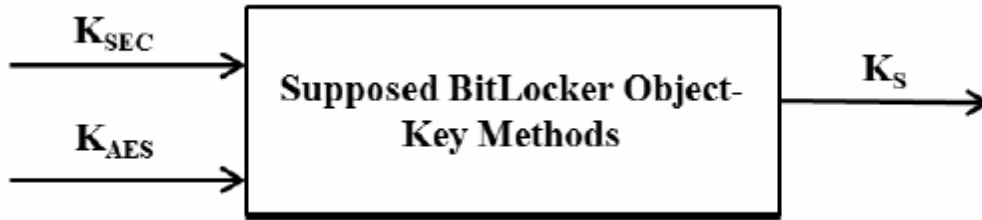
BitLocker, the other accused technology of Microsoft, uses an algorithm, implemented in software, to encrypt and decrypt data on a system volume in blocks corresponding to disk sectors. In the BitLocker encryption process, for each fixed-size disk sector to be encrypted, a 512-bit key, comprising a sector key component (“ K_{SEC} ”) and an AES-CBC component (“ K_{AES} ”), is allegedly “modified” by the methods described by the following equation:

$$K_s := E(K_{SEC}, e(s)) \parallel E(K_{SEC}, e'(s))$$

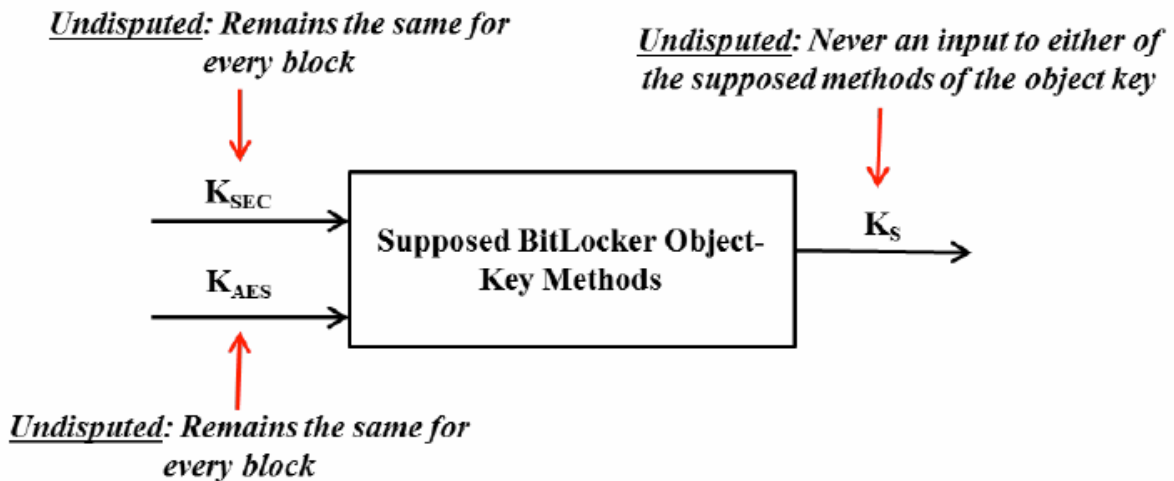
$E()$ is the AES encryption function, and $e()$ and $e'()$ are encoding functions. This results in a different modified key for each disk sector. (Blaze Decl., at ¶ 57–58.)

With regard to BitLocker, the Defendant claims that Dr. Blaze repeated the same faulty analysis in connection with consideration of the alleged “object key” in this technology. Dr. Blaze identified certain data values that he claimed were the “key data” of BitLocker’s “object key”: “ K_{SEC} ”, “ K_{AES} ”, and K_S ”. He also identified two “methods” in BitLocker that modify the key data.

[remainder of page intentionally left blank]



However, according to Microsoft, Dr. Blaze admitted that two of the three key data values—“ K_{SEC} ” and “ K_{AES} ”—are never modified at all, and remain the same for every block encrypted. Moreover, although K_S does vary from block to block, it does not do so pursuant to the “key method” identified by Dr. Blaze. (See Blaze 9/13/2011 Tr. at 117–18 (“No, the K_S value of a previous sector is not one of the inputs to this equation [used to derive K_S]”).) Therefore, Microsoft contends that the undisputed record can be accurately depicted by the following simplified diagram:



Microsoft’s arguments in this regard are nearly identical to those in the TKIP context. Microsoft contends that K_{SEC} and K_{AES} are merely “inputs” into the identified key methods and that these inputs are identical from block to block, so that it cannot be said that they are ever “modified” under the Court’s claim construction. However, once again the Defendant

misconstrues the Plaintiff's expert's supposed "admissions", and fails to offer any evidence of its own to demonstrate that there are no issues of fact as to this highly complicated encryption technology.

Instead, it appears to the Court that there may be an issue of fact as to whether the identified key data—namely K_{SEC} and K_{AES} —are modified by the supposed BitLocker object-key methods. For instance, the BitLocker Whitepaper (Pl. Ex. G) describes how K_{AES} is used in a function— $E(K_{AES}, e(s))$ —and then the plaintext is encoded using the output of that function and one of the relevant methods—AES-CBC. The Plaintiff's expert provides support for this evidence. He maintains that the initial-state key data, specifically a 512-bit key, comprising K_{SEC} and K_{AES} , is repeatedly modified by the methods defined in the relevant equation. He states that the 512-bit key for BitLocker "is modified by the identified methods because those methods take the initial state data as input and change it to produce a different output—the modified key, comprising K_S and K_{AES} ."

The Defendant's contention that "because . . . K_{SEC} and K_{AES} (the supposed object key data) are inputs into the equation that Dr. Blaze says is the object key method, the result of running that equation is not that K_{SEC} and K_{AES} change, but rather that a new value of a different item, K_S , is created." Once again, that does not preclude a finding that those two data values are not "modified". The Defendant has provided nothing except for its own interpretation of the technology, that running the equation does not modify the data. This is insufficient for the Court to find that there is no question of fact as to whether this data is in fact "modified" under the plain meaning this Court has given this term. This is not to say that simply because something is an "input", then it automatically means it is "modified". However, it is a question that should

rightfully go to jury and cannot be determined by this Court at this juncture based on the record before it.

With regard to K_S , as with the WEP seed above, the Court agrees that it cannot be “key data” under the object key claims limitation as a matter of law. However, to the extent K_{SEC} and K_{AES} may possibly be found to be modified for each block pursuant to the process contained in the object key, then the claims limitation may nevertheless be satisfied.

Accordingly, the Defendant’s motion for summary judgment in connection with the “object key” claims limitation is denied.

On a final note, in its opposition brief to the instant summary judgment motion, the Plaintiff raises the argument that both TKIP and BitLocker satisfy the “object key” limitation under the doctrine of equivalents. The Defendant characterizes this as a “surprise infringement theory” because Dr. Blaze never offered a doctrine of equivalents theory for “object key” in any of his expert reports and accordingly, Microsoft contends that it should be discarded as in violation of Fed. R. Civ. P. 26 and Local Rule 56.1. The Court agrees with the Defendant that the Plaintiff may not be permitted at this late stage of the litigation to assert a new theory of liability. Munoz v. City of New York, No. 04 Civ. 1105, 2008 WL 464236, at *7 (S.D.N.Y. Feb. 20, 2008) (“In general, plaintiffs are not permitted to raise new theories of their case in opposition to a motion for summary judgment.”) (citing Rendely v. Town of Huntington, No. 03 Civ. 03805, 2006 WL 5217083, at *6 (E.D.N.Y. Aug. 30, 2006)). Thus, to the extent the Plaintiff raises any doctrine of equivalents arguments in connection with the “object key”, these contentions will be disregarded.

E. As to Whether TKIP and BitLocker Satisfy the “Repeating Step”

Finally, the Defendant claims that the Court should hold that Microsoft has not infringed either of claims 2 and 33 with regard to either TKIP or BitLocker. Claims 2 and 33 are each directed at a method for encrypting data, as opposed to the system claims. Each of these claims requires, in part, the steps of:

- “creating at least one object key in a block cipher,”
- “modifying the at least one object key . . .,” and
- “repeating the steps of modifying the at least one object key, modifying the key schedule and encrypting utilizing the modified key schedule until the encrypting of blocks of plaintext data is completed.”

In the Markman Order, the Court found the claim term “repeating the step[] of modifying the at least one object key” to be understood as limited, because the Court construed this to mean that the object key’s data, as it presently exists in the object key at each instance of modification, must be an input into the modification methods of that object key. In light of this construction, Paone no longer contends that these claims are literally infringed by either technology. (Def. Ex. C, at ¶ 5.) This is because the key modification methods performed by TKIP and BitLocker differ from the “repeating step” of the asserted claims in that the initial state of the “object key” data is repeatedly input into the “object key” methods for each input block of plaintext data, rather than the output of the “object key” methods being input into those methods for the next block of input plaintext data. (See Blaze Decl., at ¶¶ 68–70.) Thus, the Court now holds, as a matter of law, that there has been no literal infringement of claims 2 or 33.

In addition, the Defendant also urges the Court to find that Paone is barred by prosecution history estoppel from asserting that the use of TKIP or BitLocker satisfies the recited “repeated” step under the doctrine of equivalents. For its part, the Plaintiff claims that even though

prosecution history estoppel is presented as a question of law, it raises a number of genuine issues of underlying material fact that are vigorously disputed by the parties, and thus is not a matter for summary judgment.

1. Prosecution History Estoppel

Prosecution history estoppel may, as a matter of law, prevent a patentee from subsequently relying on the doctrine of equivalents for a particular claim limitation. Bai v. L & L Wings, Inc., 160 F.3d 1350, 1354 (Fed. Cir. 1998). Amendment-based prosecution history estoppel “arises when an amendment is made to secure the patent and the amendment narrows the patent’s scope.” Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co., Ltd., 535 U.S. 722, 736, 122 S. Ct. 1831, 152 L. Ed. 2d 944 (2002). Thus, prosecution history estoppel limits the range of equivalents covered by a patent claim when the claim has been distinguished from relevant prior art during prosecution of the patent. In other words, it stops a patentee from attempting to recapture through equivalence certain claims coverage given up during prosecution. Whatever subject matter was distinguished from the patent claim is deemed surrendered by the patentee, and cannot be recaptured by claims of infringement under the doctrine of equivalents. See Southwall Techs., Inc. v. Cardinal IG Co., 54 F.3d 1570, 1578–79 (Fed. Cir. 1995). When a court applies the doctrine of prosecution history estoppel, “a close examination must be made as to, not only what was surrendered, but also the reason for such a surrender.” Id. at 1580 (quotations and citations omitted).

“A patentee’s decision to narrow his claims through amendment may be presumed to be a general disclaimer of the territory between the original claim and the amended claim.” Festo, 535 U.S. at 740, 122 S. Ct. 1831. But the patentee may rebut this presumption by “demonstrat[ing] that the alleged equivalent would have been unforeseeable at the time of the

narrowing amendment, that the rationale underlying the narrowing amendment bore no more than a tangential relation to the equivalent in question, or that there was ‘some other reason’ suggesting that the patentee could not reasonably have been expected to have described the alleged equivalent.” Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co., Ltd. (“Festo II”), 344 F.3d 1359, 1368 (Fed. Cir. 2003) (en banc) (quoting Festo, 535 U.S. at 741, 122 S. Ct. 1831). The Court will address in turn the presumption of surrender and Paone’s effort to rebut that presumption.

2. Presumption of Surrender

During prosecution, Paone narrowed the scope of the ‘789 patent by filing amendments to his application claims that specifically recited the repeated step language. As explained above, claim 1 is incorporated by reference into claim 2. The original application for claim 1 was filed as follows:

1. A computer implemented method for encrypting data comprising the steps of:

creating an object key comprising data and methods that operate on said data;
and

encrypting input plaintext data utilizing said object key in conjunction with an encryption process.

(Paone’s 56.1, at ¶ 8.) Application claim 10, as originally filed, depended on claim 1. (‘789 File History at MSLP0854062–64, Def. Ex. H.) Application claim 10 further required that the “object key” be “dynamic” and that “a modification method of said object key includes a hashing function.” (Id. at 64.)

The Patent Office initially rejected these claims, in part because of prior art disclosures in the “Wood patent”, U.S. Patent 5003596. (‘789 File History, Def. Ex. H.) In this regard, on March 16, 2000, the PTO issued a communication, known as an “office action”, during prosecution of the ‘789 patent, which stated:

As per claim 1, the claimed invention teaches encrypting input plaintext using an object key where the object key comprises data and methods. Wood (US 5003596) teaches a block encryption method to convert a block of input plaintext into a unique block of ciphertext (see column 3, lines 38–40 and Fig. 1). In the method of Wood (US 5003596), encryption keys are selected from a key table for use in the encryption process (column 3, lines 66–68). The block encryption method in Wood (US 5003596) differs from the block encryption method in claim 1 since the encryption keys in Wood (US 5003596) are not object keys as defined in claim 1.

[However], one of ordinary skill in the art of programming would know to code the encryption keys in Wood (US 5003596) . . . to create an encryption key object

(See '789 patent file history, 3/16/2000 Office Action at 3 (MSLP0854434).) Paone understood the Wood reference as teaching a static key schedule. In other words, while each plaintext input block in Wood would be encrypted uniquely, this was based solely on the combination of an unchanging key schedule (table), the current state of the cypher text, and what were referred to as “mask values”.

The major difference between the two innovations, according to Paone, was that his invention used a dynamic object key to create a dynamic key schedule for each block. Thus, the object key and the key schedule were changing for each block of plaintext data. Accordingly, to distinguish his invention over the prior art, Paone undisputedly modified his application claim 1 by amending it to include five additional steps, one of which was the “repeating” step. The following demonstrates the relevant revisions to the claim language:

A computer implemented method for encrypting data comprising the steps of:
creating [an] at least one object key in a block cipher, the at least one object key
comprising data and methods that operate on said data;

creating a key schedule based upon the at least one object key;

encrypting a block of input plaintext data utilizing said [object] key schedule;
[in conjunction with an encryption process]

modifying the at least one object key;

modifying the key schedule based upon the at least one modified object key;

encrypting a next block of input plaintext data utilizing said modified key schedule; and

repeating the steps of modifying the at least one object key, modifying the key schedule and encrypting utilizing the modified key schedule until the encrypting of blocks of plaintext data is completed.

(Id. at ¶ 15 (emphasis added), ‘789 File History at MSLP0854544–45, Def. Ex. H.) As for the dependent claim 10, he rewrote the claim as a new independent claim—application claim 41 (now patent claim 32), and added the “repeating step” limitation to this as well. (‘789 File History at MSLP0854640, Ex. H.) The following demonstrates the revisions to the claim language:

Application claim 41. (Newly added) A computer implemented method for encrypting data comprising the steps of:

creating [an] at least one object key in a block cipher, the at least one object key comprising data and methods that operate on said data;

creating a key schedule based upon the at least one object key; encrypting a block of input plaintext data utilizing said [object] key schedule; [in conjunction with an encryption process]

modifying the at least one object key using at least a non-linear function;

modifying the key schedule based upon the at least one modified object key;

encrypting a next block of input plaintext data utilizing said modified key schedule; and

repeating the steps of modifying the at least one object key, modifying the key schedule and encrypting utilizing the modified key schedule until the encrypting of blocks of plaintext data is completed.

(emphasis added).

Paone notified the Patent Office that the new claim 41 included the “allowable subject matter of claims 1 and 10.” (Id. at 641.) To be clear, claim 41 was not amended to add the repeating step. Rather, this was a new independent claim that included this language. However,

according to Microsoft, “the narrowing effect is the same because the claim language is the same in relevant part and patentees cannot make an end-run around prosecution history estoppel by rewriting dependent claims in independent form.” (Def. Mem., at 22 (citing Honeywell Intern. Inc. v. Hamilton Sundstrand Corp., 370 F.3d 1131, 1140 (Fed. Cir. 2004); Deering Precision Instruments, L.L.C. v. Vector Distribution Sys., Inc., 347 F.3d 1314, 1326 (Fed. Cir. 2003).) In sum, claim 1—which forms the basis of what is now claim 2—and claim 41—which forms the basis of what is now claim 33—were either amended or redrafted to include the “repeating step” limitation language, which is arguably narrower than the two-step method originally recited in the as-filed claims.

The Court agrees that based upon the particular circumstances of this case, namely, the relevant prosecution history, “[t]he mere fact that his claims were narrowed by rewriting them to include the repeating step creates a presumption that Mr. Paone added the step for the purpose of securing his patent” and thus there is a presumption against the availability of the doctrine of equivalents. (Def. Mem. at 23); see Warner-Jenkinson Co., 520 U.S. at 33; Festo II, 344 F.3d at 1366–67 (“When the prosecution history record reveals no reason for the narrowing amendment, Warner-Jenkinson presumes that the patentee had a substantial reason relating to patentability”). There is no indication in the prosecution history as to why Paone would have added the “repeating step” language were it not to secure the patent. In fact, the prosecution history lends support to the idea that this was Paone’s sole purpose for adding the language. Once Paone’s patent application was rejected and his claims were rewritten and amended, he made several arguments to the Patent Office in order to distinguish his patent from the prior art of the Wood patent.

First, he maintained that the Wood patent disclosed encrypting with a “static” key schedule, whereas his claimed method, as amended, required a dynamic key schedule. The term “dynamic” was used to identify that the object key and the key schedule are being modified, i.e., changing, *with each input block of plaintext data*. Second, Paone contended that his claimed method, as amended, was patentable over the Wood patent because, unlike Wood, his key schedule was modified in a way that “is not dependent upon the input plaintext”

Third, and most relevant to the present inquiry, Paone argued that his claimed method, as amended, was patentable over the Wood patent not only because his key schedule changed with each block of input plaintext data, but also because the process of changing the key schedule based upon the object key data “continued,” i.e., was repeated, “until all the plaintext had been encrypted”. The August 16, 2000 Amendment submitted during prosecution of the ’789 patent states:

Accordingly, the present invention describes a novel encryption technique in which at least one object key is created, a key schedule is created based upon the at least one object key, a block of input plaintext data is encrypted using the key schedule, the object key is then modified, *the modified object key is used to create a modified key schedule, and this modified key schedule is used to encrypt the next block of input plaintext data. This method is continued until all plaintext data has been encrypted.*

(’789 patent file history, 8/16/2000 Amendment at 15) (emphasis added). Based upon these arguments to the Patent Office, it appears obvious to the Court that Paone added the repeating step limitation for the purpose of securing his patent. See Festo, 535 U.S. at 727, 122 S. Ct. 1831 (“When the patentee responds to the rejection by narrowing his claims, this prosecution history estops him from later arguing that the subject matter covered by the original, broader claim was nothing more than an equivalent.” (emphasis added)); id. at 734, 122 S. Ct. 1831 (“[A patentee’s] decision to forgo an appeal and submit an amended claim is taken as a concession that the invention as patented does not reach as far as the original claim.” (emphasis added)); id. at 740,

122 S. Ct. 1831 (“A patentee’s decision to narrow his claims through amendment may be presumed to be a general disclaimer of the territory between the original claim and the amended claim.” (emphasis added)).

Certainly, one crucial difference between the Wood prior art and Paone’s application was that the key schedule in Wood was static, while the key schedule in Paone was dynamic. Moreover, another key distinction was that Paone’s key schedule was created based upon a dynamic object key, rather than the plaintext or the ciphertext as in the Wood patent. However, it is the addition of language describing the extrapolation from these two facts—the repeating step—that was a crucial for Paone to secure his patent. The creation of a key schedule for each block based upon a modified object key was only a part of what distinguished Paone’s invention from the prior art. What accomplished the full distinction was that a modified object key would create a modified key schedule for each and every new block of plaintext data, until all of the data had been encrypted. Put another way, “repeating the steps of modifying the at least one object key, modifying the key schedule and encrypting utilizing the modified key schedule until the encrypting of blocks of plaintext data is completed”, was a necessary final step of the process in Paone’s application to help distinguish his invention over prior art. Without the “repeating step”, an object key would be created; a key schedule would be created; a block would be encrypted; both the object key and key schedule would be modified; and then the next block would be encrypted utilizing the modified key schedule. However, it appears from the prosecution record that the recurrence and dependence of these steps was not entirely clear without the addition of the repeating step language. It is the repetitive nature of modifying the object key for each block, and in turn modifying the key schedule for each block, that truly made Paone’s patent application unique. Thus, in part to overcome the Wood patent, it was necessary

to include a statement that every single block is subject to a modified key schedule based upon a modified object key.

Although claim 41 (which now forms the basis of claim 33) was not an amended claim *per se*, but rather a rewritten new independent claim, this is irrelevant. Deering Precision Instruments, L.L.C. v. Vector Distribution Sys., Inc., 347 F.3d 1314, 1325 (Fed. Cir. 2003) (“Deering’s addition of [a rewritten independent claim], coupled with the clear surrender of the broader subject matter of the deleted original independent claim presumptively bars Deering from arguing infringement under the doctrine of equivalents.”); see also Honeywell, 370 F.3d at 1142 (“[T]he fact that the scope of the rewritten claim has remained unchanged will not preclude the application of prosecution history estoppel if, by canceling the original independent claim and rewriting the dependent claims into independent form, the scope of subject matter claimed in the independent claim has been narrowed to secure the patent.”).

Furthermore, it is also immaterial that that the amendments were to application claims 1 and 41, rather than to application claims 2 and 33, which are the present basis for Paone’s alleged infringed claims. The presumption of surrender “applies to all claims containing the [added] [l]imitation, regardless of whether the claim was, or was not, amended during prosecution.” Deering, 347 F.3d at 132; see also Builders Concrete, Inc. v. Bremerton Concrete Prods. Co., 757 F.2d 255, 260 (Fed. Cir. 1985) (“[T]he prosecution history of all claims is not insulated from review in connection with determining the fair scope of [the asserted claim]. To hold otherwise would be to exalt form over substance and distort the logic of this jurisprudence, which serves as an effective and useful guide to the understanding of patent claims. The fact that the [the limitation in question] was not itself amended during prosecution does not mean that it

can be extended by the doctrine of equivalents to cover the precise subject matter that was relinquished in order to obtain allowance of [another claim].”).

The Plaintiff argues that the addition of the “repeating step” did not narrow the scope of the asserted claims, so that the presumption of prosecution history estoppel need not apply. In particular, Paone contends that Dr. Blaze has explained that one of ordinary skill in the art would have understood the addition of the “repeating step” to have merely explicitly clarified a requirement that was already implicit in the claims as previously written—namely, that the “object key” data was to be repeatedly modified by “object key” methods. (Blaze Decl., at ¶¶ 83–84.) However, this argument is without merit. Dr. Blaze’s interpretation of what was “implicit” in the original patent application is not determinative. Rather, it is the prosecution history record that is significant.

The initial application included the creation of an object key to play a role in the encryption of input plaintext data, in which the object key would have both data and methods that operate on said data. While it may have been implicit that the object key would change — application 10 did state that the “object key” would be “dynamic” — it is not plain from the initial application that the process of changing the object key, and consequently the key schedule, continued for every single block of plaintext data until the entire document or image was encrypted. While this may have been implicit to Paone, there is nothing in the prosecution history to support that notion. Moreover, the Court rejects the idea that repeating the modification of the object key was so obvious, that it need not have been included because it did not act to narrow the claims. “[C]laims are interpreted with an eye toward giving effect to all terms in the claim.” Bicon, Inc. v. Straumann Co., 441 F.3d 945, 950 (Fed. Cir. 2006) (“Allowing a patentee to argue that physical structures and characteristics specifically described

in a claim are merely superfluous would render the scope of the patent ambiguous, leaving examiners and the public to guess about which claim language the drafter deems necessary to his claimed invention and which language is merely superfluous, nonlimiting elaboration.”); see, e.g., Elekta Instrument S.A. v. O.U.R. Scientific Int’l, Inc., 214 F.3d 1302, 1305, 1307 (Fed. Cir. 2000) (finding claim language “only within a zone extending between latitudes 30°–45°” did not read on a device with radiation sources extending between 14° and 43° because “[a]ny other conclusion renders the reference to 30° superfluous”); Unique Concepts, Inc. v. Brown, 939 F.2d 1558, 1563 (Fed. Cir. 1991) (“When the language of a claim is clear, as here, and a different interpretation would render meaningless express claim limitations, we do not resort to speculative interpretation based on claims not granted.”); In re Danly, 46 C.C.P.A. 792, 263 F.2d 844, 847 (1959) (limiting claims to require that the claimed device actually be connected to an alternating current source because, although the claims “do not positively recite a source of alternating current as an element of the claims,” any other interpretation would render certain language in the claims meaningless).

Therefore, the Court agrees that there is a presumption that the “rendering step” limitation was added to the claims to narrow their scope to permit allowance.

3. Tangentiality

Nonetheless, Paone may rebut this presumption by showing, inter alia, that “the rationale underlying the narrowing amendment bore no more than a tangential relation to the equivalent in question.” Festo II, 344 F.3d at 1368. “[T]he patentee bears the burden of showing that a narrowing amendment did not surrender a particular equivalent” Festo, 344 F.3d at 1368. “Whether a patent-holder has successfully rebutted the Festo presumption of the surrender of equivalents is a question of law, which [is] review[ed] de novo.” Amgen Inc. v. Hoechst Marion

Roussel, Inc., 457 F.3d 1293, 1312 (Fed. Cir. 2006). It is a rare case in which the presumption is rebutted. As noted by Federal Circuit Judge Randall R. Rader in a concurring opinion five years ago, “Festo itself recognized that rebuttals under the tangential principle will be rare. . . . Cases in the interim have confirmed Festo’s insight; only two cases have successfully invoked the tangential rebuttal principle in this court.” Cross Med. Prods., Inc. v. Medtronic Sofamor Danek, Inc., 480 F.3d 1335, 1346 (Fed. Cir. 2007) (concurring).

“[A]n amendment made to avoid prior art that contains the equivalent in question is not tangential; it is central to allowance of the claim. . . . [T]he inquiry into whether a patentee can rebut the Festo presumption under the ‘tangential’ criterion focuses on the patentee’s objectively apparent reason for the narrowing amendment[, which must be] discernible from the prosecution history record. . . .” Id. at 1369; see Intervet Inc. v. Merial Ltd., 617 F.3d 1282, 1291 (Fed. Cir. 2010) (“Although there is no hard-and-fast test for what is and what is not a tangential relation, it is clear that an amendment made to avoid prior art that contains the equivalent in question is not tangential.”).

“The scope of the estoppel must fit the nature of the narrowing amendment. A district court must look to the specifics of the amendment and the rejection that provoked the amendment to determine whether estoppel precludes the particular doctrine of equivalents argument being made.” Id.; See Festo, 535 U.S. at 737–38, 122 S. Ct. 1831 (“There is no reason why a narrowing amendment should be deemed to relinquish equivalents . . . beyond a fair interpretation of what was surrendered.”).

Paone requests the Court to find that he has successfully rebutted the presumption of amendment-based prosecution history estoppel. Specifically, Paone argues that the accused equivalents differ from the claimed invention only in that in Microsoft’s technologies, the

repeating step of modifying the object key data is enacted solely through acts on the initial state of the data, rather than the output of the previous block's object key methods (i.e., modified object key data). This difference, Paone submits, is unrelated to the rationale for the amendment, which was to distinguish the claimed invention from the prior art based only on the static nature of Wood's key schedule.

The inquiry is “whether the reason for the narrowing amendment is peripheral, or not directly relevant, to the alleged equivalent.” Festo II, 344 F.3d at 1369. The tangential rebuttal principle is quite narrow. Cross Med. Prods., Inc. v. Medtronic Sofamor Danek, Inc., 480 F.3d 1335, 1348 (Fed. Cir. 2007); see Biagro W. Sales, Inc. v. Grow More, Inc., 423 F.3d 1296, 1306 (Fed. Cir. 2005) (distinguishing a case finding the tangentiality presumption rebutted as limited to situations in which the prosecution history clearly demonstrates that “the amendment and alleged equivalent involve different aspects of the invention”).

Here, the applicant amended the claims to overcome prior art that encrypted data utilizing keys based on a static key schedule. In amending the claims, Paone added the additional steps, including the “repeating step”, to demonstrate that object keys were modified for each block, which in turn modified the key schedule so that it was dynamic, and that these two modifications occurred for each and every block until the encryption was complete. Paone himself argued that the thrust of the novelty of his invention was not only that an object key was created, and that a key schedule was created based upon this object key, but that a modified object key would then go on to create a modified key schedule, in a somewhat domino effect, until all plaintext data had been encrypted. There is clearly more than a tangential relationship between the reason for the amendment and the accused equivalents.

Moreover, Paone’s asserted reason for why this amendment is merely “peripheral” is that it was only meant to distinguish the claimed invention from the Wood prior art based upon the dynamic key schedule. However, it is not “objectively apparent” from Paone’s arguments that he made the addition of the “repeating step” language only to address the dynamic key schedule. This is because from the initial filing to the PTO, Paone made clear throughout the application that the key schedule would be modified for each block. (See ‘789 Patent Application, at MSLP0854043 (“The key modification is performed for each input plaintext data block so that each data block is encrypted with a different key.”).) Thus, Paone has identified no explanation in the prosecution history for the addition of the repeating step limitation. Therefore, Paone cannot meet his burden to show that the rationale for adding the repeating step limitation was tangential to the equivalents in question—using the initial state of the object key data (as opposed to the present state) as the input to the modification methods. The point of the repeating step was to indicate that Paone’s advancement was the consistent loop: the object key would create the key schedule, a modified object key (based upon a random session key) would then create a modified key schedule, and so forth. This is precisely connected to—and not merely tangential to—the question of whether an object key’s methods act upon key data that is modified or in its original state. The addition of the “repeating step” language is to ensure modification of the *already modified* object key.

Accordingly, the Court holds that Paone is barred by prosecution history estoppel from asserting infringement under the doctrine of equivalents as to the “repeating step” limitation.

For the foregoing reasons, the Court grants summary judgment of noninfringement of claims 2 and 33 with respect to both BitLocker and TKIP. Paone alleges infringement of these claims only under the doctrine of equivalents, and because he is barred from doing so, as a

matter of law, by virtue of the doctrine of prosecution history estoppel, infringement of these claims is denied. Accordingly, the Defendant's summary judgment motion in this regard is granted.

III. CONCLUSION

Based upon the above findings, any infringement claims by Paone against Microsoft as to the method claims in the '789 patent—2 and 33—are dismissed. Consequently, any infringement claims as to the BitLocker technology are also dismissed. As to the remaining system claims asserted against Microsoft as to TKIP, these claims are still viable. Paone will need to demonstrate that TKIP meets each of the limitations in those two claims, either literally or under the doctrine of equivalents. In particular, there are questions of fact that remain as to whether TKIP meets the “block” limitation under the doctrine of equivalents and whether TKIP meets the “object key” limitation literally.

For the foregoing reasons, it is hereby

ORDERED that the Defendant's motion for summary judgment for a finding of no literal infringement as to the “block” and “block cipher” limitation in all claims is GRANTED; and it is further

ORDERED that the Defendant's motion for summary judgment for a finding of no infringement as to the “block” and “block cipher” limitation in all claims under the doctrine of equivalents is DENIED; and it is further

ORDERED that the Defendant's motion for summary judgment for a finding of no infringement as to the “object key” limitation in all claims is DENIED; and it is further

ORDERED that the Defendant's motion for summary judgment as to claims 2 and 33 is GRANTED; and it is further

ORDERED that the parties are directed to appear on Monday, August 6, 2012 at 9:30am for a conference to set a trial date.

SO ORDERED.

Dated: Central Islip, New York
July 30, 2012

/s/ Arthur D. Spatt
ARTHUR D. SPATT
United States District Judge