United States District Court,
N.D. California, San Jose Division.

CRYPTOGRAPHY RESEARCH, INC,
Plaintiff.
v.
VISA INTERNATIONAL SERVICE ASSOC., et al,
Defendant.

No. C 04-04143 JW

May 22, 2007.

Darren E. Donnelly, J. David Hadden, Lynn H. Pasahow, Laurie Michelle Charrington, Ryan Aftel Tyz, Fenwick & West LLP, Stephen Roger Dartt, Mountain View, CA, David Douglas Schumann, Jedediah Wakefield, Fenwick & West LLP, San Francisco, CA, for Plaintiff.

Ann S. Auiler, Mary Elaine Johnston, Joshua David Dick, Michael J. Gallagher, White & Case LLP, New York, NY, Brandon D. Baum, Dennis S. Corgill, Eric Butler Evans, Joshua Michael Masur, Michael A. Molano, William Healey, Ian N. Feinberg, Mayer Brown Rowe & Maw LLP, Palo Alto, CA, Elizabeth S. Campbell, Philadelphia, PA, Martin Frank Majestic, Alexandra V. Percy, Michael A. Duncheon, Hanson Bridgett Marcus Vlahos & Rudy, LLP, San Francisco, CA, Joseph Helmsen, W. Joseph Melnik, Pepper Hamilton LLP, Pittsburgh, PA, for Defendants.

THIRD CLAIM CONSTRUCTION ORDER

JAMES WARE**, United States District Judge.**

I. INTRODUCTION

This is the Third Claim Construction Order in this patent infringement action filed by Plaintiff Cryptography Research, Inc. (CRI) against Defendant Visa International Service Association (Visa). This Order sets forth the Court's construction of disputed words and phrases in claims of the '442 Patent according to the previously stated legal standards.

II. DISCUSSION

A. *The '442 Patent-Claim 1*

Claim 1 provides in part FN1:

FN1. Unless otherwise indicated, all bold typeface is added by the Court to emphasize words and phrases under consideration.

A method of **cryptographically processing** a message using an asymmetric cryptographic protocol involving a private key including a secret exponent, in a manner **resistant to** external detection of the secret exponent, comprising the steps of:
(a) obtaining a digital quantity representative of at least a portion of said message, said digital quantity to be cryptographically processed using an asymmetric cryptographic protocol involving a private key including a secret exponent and an associated modulus;

(b) transforming said exponent to an expanded representation thereof, said expanded representation including a sequence of symbols, each said symbol representing a modular multiplication;

(c) loading an **accumulator** with a positive integer power of said digital quantity;

(d) for each said symbol in at least a portion of said expanded representation:

(i) **multiplying, modulo said modulus, said value in said accumulator by a positive integer power of said digital quantity, said integer power being indicated by said symbol,** and

(ii) updating said accumulator with the result of said step (i),

(iii) by **executing a plurality of microprocessor instructions whose form is independent of the value of said symbol;** thereby crytographically processing said digital quantity in a manner resistant to external detection of said secret exponent.

1. "cryptographically processing ... in a manner resistant to external detection of the secret exponent"

The Court has previously construed the key phrases "cryptographically processing" and "resistant to." (*See* First and Second Claim Construction Orders, Docket Item Nos. 269, 330, respectively.) The Court declines to construe further the Preamble of Claim 1 of the '442 Patent. The parties are invited to notify the Court of any other particular words or phrases in the preamble for which construction is requested.

2. "loading an accumulator with a positive integer power of said digital quantity"

A step in the method is "loading an accumulator with a positive integer power of said digital quantity." At issue is the proper construction of the word "accumulator." The parties do not dispute the construction of the word "accumulator." They have stipulated that "accumulator" should be construed to mean "a memory location or variable." The Court declines to adopt this stipulation as its construction.

The '442 Patent discloses methods and apparatuses for securing cryptographic systems from external monitoring attacks. Although Claim 1 is a method claim, a person of ordinary skill in the art reading the patent documents would understand that the inventors use the word "accumulator" to mean both an apparatus and a value. Therefore, the Court must determine how the word is being used in the "loading step" of Claim 1.

A person of ordinary skill in the relevant art would understand that the word "accumulator" is commonly used to refer to an apparatus. An "accumulator" is a register used to hold values to be used for logic or arithmetic operations, usually to count items or accumulate a sum. *See* Microsoft Computer Dictionary, 15 (5th ed.2002); *see also* Webster's New World Computer Dictionary, 9 (10th ed.2003).

The word "accumulator" is also commonly used to describe a particular value which is being affected by a particular arithmetic operation. For example, an "accumulator value" would be understood by those skilled in the art as referring to a value in a register which may be modified depending upon arithmetic operations.

In the '442 Patent, the inventors use the word "accumulator" to refer both to a register and a value in a register:

(2c) step (d)(i) is thereafter performed as follows:

(i) if said symbol is of said first type, said step of multiplying said value in said **accumulator** by said positive integer power of said digital quantity includes multiplying said **accumulator value** by itself, where said **accumulator** contains a positive integer power of said digital quantity ...

('442 Patent, Col. 11:51-57.) In the above excerpt from Claim 2, the word "accumulator" refers to an apparatus, i.e., a register, and the phrase "accumulator value" refers to a value in the apparatus.

In the written description of the '442 Patent, the inventors use the phrases "accumulator R" and "the accumulator steps:"

Here, y.sub.13 i denotes bit i of y such that y.sub.-0 is the least significant bit and y__(k-1) is the most significant bit. In standard exponent encoding conventions, a "0" bit specifies a squaring (i.e., multiplication of **accumulator R by itself**) while a "1" bit specifies squaring followed by a multiplication (of **accumulator R by base x**).

* * *

In this case, **the accumulator steps** updating R in the modular exponentiation loop would perform two squaring (mod n) operations, followed by a multiplication (mod n) with one of the table values: x 0 for a "00" bit pair (or "0" in base 4); x 1 for a "01" bit pair (or "1" in base 4); x 2 for a "10" bit pair (or "2" in base 4); and x 3 for a "11" bit pair (or "3" in base 4).

* * *

It will also be understood by one skilled in the art that various combinations of the variations discussed here can be used in connection with the invention. For example and without limitation, the operation-based encoding schemes may be combined with the bit windowing techniques using k-ary modular exponentiation where each nonzero exponent digit could represent a power of x, while zero digits could represent squaring operations. For example, as stated above, a "1" digit denotes simple multiplication of the result accumulator by the value x (i.e., by exponentiation of x to the power 1). Similarly, a "2" digit (if used) denotes multiplication x 2 mod n, and so forth. In one embodiment of the invention, a table of pointers may be employed to indicate the value of the bit. For example, the first entry (offset zero) could be a pointer to the result **accumulator R** (for squaring operations), the entry at offset 1 could point to x (i.e., x 1), the entry at offset 2 (if used) could point to the precomputed value x 2 mod n, and the entry at offset 3 (if used) could point to the precomputed value x 3 mod n. The powers of x may be precomputed at the beginning of the modular exponentiation operation; even so, the performance benefit obtained by reducing the number of multiplication operations during the modular exponentiation generally more than compensates for the

precomputation time. Note that x 0 (equivalent to multiplication by 1) is not used; all steps involve multiplication with a number larger than 1 because "0" digits in the encoding represent multiplication by R.

('442 Patent, Col. 4:26-32; 51-56; Col. 7:64-8:23.) One of ordinary skill would understand the phrase "accumulator R" to refer to a value. The phrase "the accumulator steps" refers to operations on a value.

The Court declines to adopt the parties' stipulation defining "accumulator" as a "memory location." Although a register may be a memory location; a memory location is not a register.FN2 In addition, the Court declines to adopt the parties' stipulated definition, "variable." A "variable" is commonly understood to be a value which can change as a program executes. *See* Microsoft Computer Dictionary, 547 (5th ed.2002). Thus, while an "accumulator value" might be a variable, an "accumulator" is not a variable.

FN2. To define "accumulator" as a "memory location" might cause confusion with other devices such as "random access memory," or "memory cell," or "memory address," all of which might be called a "memory location."

The Court construes the word **"accumulator,"** as it is used in the phrase, "loading an accumulator with a positive integer power of said digital quantity," as follows: **a register used in arithmetic operations.**

3. "multiplying, modulo said modulus, said value in said accumulator by a positive integer power of said digital quantity, said integer power being indicated by said symbol"

The Preamble to Claim 1 discloses a method for cryptographically processing a "message" using a "private key including a secret exponent and an associated modulus." In step (a), a "digital quantity" representative of at least a portion of the "message" is obtained.FN3 In step (c) FN4, a positive integer power of the "digital quantity" is loaded into an accumulator. In step (b), the "secret exponent" of the private key is transformed into an "expanded representation" of itself, which includes a sequence of "symbols, each symbol representing a modular multiplication." Step (d) operates on each "symbol in at least a portion of the expanded representation." The step (d) operation on each "symbol" is divided into two FN5 sub-steps: (d)(i), a "multiplying" sub-step; and (d)(ii), an "updating" sub-step. The "multiplying" and "updating" sub-steps are accomplished by executing microprocessor instructions as disclosed in (d)(iii).

FN3. The provision in step (a) that "said digital quantity to be cryptographically processed using an asymmetric cryptographic protocol involving a private key including a secret exponent and an associated modulus" is not a limitation on the step. This language simply expresses the intended result of the process. *See* Texas Instruments Inc. v. U.S. Int'l Trade Commission, 988 F.2d 1165, 1172 (Fed.Cir.1993).

FN4. In tracing the method, the Court goes from step (a) to step (c) because step (b) operated on the private key disclosed in the Preamble and does not operate on the "digital quantity" disclosed in step (a).

FN5. Step (d) is enumerated with three subparagraphs. However, there are only two sub-steps, namely d(i) and d(ii). Subparagraph d(iii) limits how d(i) and d(ii) are performed.

In the "multiplying" sub-step (d)(i), a modular multiplication is performed. This sub-step contains the following limitation: "said integer power being **indicated by** said symbol." The parties dispute whether **"indicated by"** should be defined to mean "determined by" or whether it should be defined to mean "directly determined by."

The inclusion of the adverb "directly" (meaning immediately) in the construction would connote that the literal operation of the exponent must be followed. The method is not limited to literal operation of the exponent. Claim 1 discloses a cryptographic algorithm in which the exponent itself represents an algorithm. As explained in the following description of an embodiment, the exponent could be a pointer to another value:

It will also be understood by one skilled in the art that various combinations of the variations discussed here can be used in connection with the invention. For example and without limitation, the operation-based encoding schemes may be combined with the bit windowing techniques using k-ary modular exponentiation where each nonzero exponent digit could represent a power of x, while zero digits could represent squaring operations. For example, as stated above, a "1" digit denotes simple multiplication of the result accumulator by the value x (i.e., by exponentiation of x to the power 1). Similarly, a "2" digit (if used) denotes multiplication x 2 mod n, and so forth. **In one embodiment of the invention, a table of pointers may be employed to indicate the value of the bit.** For example, the first entry (offset zero) could be a pointer to the result accumulator R (for squaring operations), the entry at offset 1 could point to x (i.e., x 1), the entry at offset 2 (if used) could point to the precomputed value x 2 mod n, and the entry at offset 3 (if used) could point to the precomputed value x 3 mod n. The powers of x may be precomputed at the beginning of the modular exponentiation operation; even so, the performance benefit obtained by reducing the number of multiplication operations during the modular exponentiation generally more than compensates for the precomputation time. Note that x 0 (equivalent to multiplication by 1) is not used; all steps involve multiplication with a number larger than 1 because "0" digits in the encoding represent multiplication by R.

('442 Patent, Col. 7:66-8:23.)

Furthermore, an independent claim should not be interpreted in a way that is inconsistent with a claim which depends from it. Southwall Techs., Inc. v. Cardinal IG Co., 54 F.3d 1570, 1579 (Fed.Cir.1995). Claim 2, which depends from Claim 1 discloses two types of "symbols," one of which is not directly used, but uses "a previously stored positive integer power." Thus, a construction of "indicated by" which requires "direct determination" of the integer power by the symbol arguably would invalidate Claim 2.

Therefore, in the claim element, "multiplying, modulo said modulus, said value in said accumulator by a positive integer power of said digital quantity, said integer power being indicated by said symbol," the Court construes the phrase "indicated by" to mean: **directly or indirectly determined by.**

4. "executing a plurality of microprocessor instructions whose form is independent of the value of said symbol"

In sub-step (d)(iii) of the method disclosed in Claim 1, the parties dispute the construction of the phrase, "executing a plurality of microprocessor instructions whose form is independent of the value of said symbol." In particular, the Court is asked to decide whether the inventors intended the phrase to mean that the method is practiced by executing microprocessor instructions having a "fixed execution path." FN6 Other claims of the ' 442 Patent disclose a similar requirement of "independence" between the execution of

a sequence of microprocessor instructions and a value.FN7

FN6. Visa proffers the following construction: "Executing multiple microprocessor instructions having a fixed execution path."

FN7. Claim 13 provides:
A method of computing a modular exponentiation involving a digital message, as part of a private key operation in an asymetric cryptosystems involving a sequence of processor instructions, comprising the steps of:

* * *

(d) using a portion of said secret exponent, deriving an index referencing one of said previously stored values, by executing **a sequence of processor instructions having a form independent of said exponent**

Claim 18 provides:

A method of computing a modular exponentiation involving a digital message, as part of a private key operation in an asymmetric cryptosystems involving a sequence of processor instructions, comprising the steps of:

* * *

(e) multiplying, modulo said modulus, (i) a value in said accumulator and (ii) said specified multiplicand, by executing **a sequence of processor instructions having a form independent of a value of said index;**

Claim 23 provides:

A method of securing a computer processor against external monitoring of cryptographic operations performed on a digital message thereby, comprising:

* * *

(b) transforming said secret portion of said private key to produce a fixed path representation of said asymmetric cryptographic operation, by defining **a sequence of logic instructions and parameter accesses such that said sequence of logic instructions and parameter accesses is independent of the value of said secret portion;**

It is the Court's understanding that microprocessor instructions may be created in a "conditional" form, i.e., the execution path changes or "branches" based on dynamic changes in a particular value. Another form of microprocessor instructions is one in which the execution path is "fixed," i.e., it does not change based on dynamic changes in a particular value.

The summary of the invention describes one of the techniques of the invention to be crytographic processes with "branchless" or "fixed execution path" microprocessor routines:

One technique of the invention reduces leakage from cryptosystems by implementing critical operations using "branchless" or **fixed execution path** routines whereby the execution path does not vary in any manner that can reveal new information about the secret key during subsequent operations. In one embodiment, this is achieved by implementing modular exponentiation without key-dependent conditional jumps.

('442 Patent, Col. 3:42-49.)

The issue is whether the requirement of a "fixed execution path" should be incorporated into the Court's construction of the meaning of sub-step (d)(iii). The plain language of sub-step (d) (iii) requires that sub-steps (d)(i) and (d)(ii) be accomplished by executing microprocessor instructions whose "form" is such that their execution path is "independent" of the value of the "symbol" disclosed in step (b).

As previously construed by the Court,FN8 the word "independent" has a plain and ordinary meaning, i.e., "does not depend on." A person of ordinary skill would understand the word "form" of microprocessor instructions to mean the sequence of the instructions. Thus, the requirement of sub-step (d)(iii) of a "form ... independent of the value" means that the sequence of instructions followed during execution of the instructions does not depend on the "value." In other words, there are no "conditional branches" in the execution of the instructions which are conditioned on the "value." There may be conditional branches in the execution path, but those branches would have to be conditioned on factors other than the "value."

FN8. *See* Claim 28 of the '783 Patent in Second Claim Construction Order.

The Court construes the phrase "executing a plurality of microprocessor instructions whose form is

independent of the value of said symbol" to mean: **executing a plurality of microprocessor instructions whose sequence during execution does not depend on the value of the symbol.**FN9

FN9. This construction applies to each Claim of the '442 Patent which discloses a method requiring microprocessor instructions having a "form" "independent" of a "value" or "exponent.." *See e.g.,* Claims listed in footnote 7.

B. The '442 Patent-Claim 13

Claim 13 provides:

A method of computing a modular exponentiation involving a digital message, as part of a private key operation in an asymetric cryptosystems involving a sequence of processor instructions, comprising the steps of:

(a) obtaining a base representative of at least a portion of said message, said base to be cryptographically processed using an asymmetric cryptographic protocol involving a private key including a secret exponent and an associated modulus;

(b) storing in a memory **one or more values representing predefined positive integer powers of said base;**

(c) storing in an accumulator a value representing said base raised to an integer power;

(d) using a portion of said secret exponent, deriving an index referencing one of said previously stored values, by executing a sequence of processor instructions having a form independent of said exponent;

(e) multiplying, modulo said modulus, (i) a **value in said accumulator** and (ii) **said referenced value,** by executing a sequence of **processor instructions having a form independent of a value of said index;**

(f) storing the result of said step (e) in said accumulator;

(g) repeating said steps (d) through (g) until said accumulator contains a representation of said base raised to said exponent;

thereby cryptographically processing said base in a manner resistant to detection of said exponent by monitoring of said processor instructions.

1. "storing in a memory one or more values representing predefined positive integer powers of said base"

At the claim construction hearing, the parties stipulated that the phrase "storing in a memory one or more values representing predefined positive integer powers of said base" means **"storing in memory one or more values representing predefined positive integer powers of the base determined by step (a) of the method."** This stipulation is supported by the claim language and other parts of the specification:

Those skilled in the art will also recognize that another common technique for computing x ^y mod n uses a

table containing precomputed powers of x.

('442 patent, Col. 4:45-47.) Based on the intrinsic evidence and the stipulation of the parties, the Court adopts the above construction.

2. "multiplying, modulo said modulus, (i) a value in said accumulator and (ii) said referenced value"

The parties dispute the construction of "a value in said accumulator" disclosed in step (e) of the method. Step (e) discloses a modular multiplication using the value in the accumulator and one of the previously stored values referenced by an index. The value in the accumulator changes as the method loops through steps (d) through (g).

The Court finds that the Claim language requires no further construction.

III. CONCLUSION

In this Order the Court has construed some of the disputed words and phrases of the '442 Patent. Some of the words and phrases for which the by the parties initially requested construction are not addressed in this Order because the Court finds that they are the same or substantially the same as words and phrases construed in the First and Second Claim Construction Orders.FN10 To the extent a party contends that an omitted word or phrase should be addressed, a supplemental request for construction should be made on a timely basis.

FN10. For example, initially the parties requested construction of the phrases: "cryptographically processing" and "securing a computer processor against external monitoring" in Claims of the '442 Patent. These terms or substantially similar terms have been previously construed by the Court.

N.D.Cal.,2007.
Cryptography Research, Inc. v. Visa Intern. Service Assoc.