(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 July 2006 (27.07.2006)

PCT

(10) International Publication Number
WO 2006/076804 A1

(51) International Patent Classification:
G06F 7/58 (2006.01) *H04L 9/28* (2006.01)

(21) International Application Number:
PCT/CA2006/000065

(22) International Filing Date: 23 January 2006 (23.01.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/644,982 21 January 2005 (21.01.2005) US

(71) Applicant *(for all designated States except US)*: **CERTICOM CORP.** [CA/CA]; 5520 Explorer Drive, 4th Floor, Mississauga, Ontario L5N 1X8 (CA).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **VANSTONE, Scott A.** [CA/CA]; 10140 Pineview Trail, P.o. Box 490, Campbellville, Ontario L0P 1B0 (CA). **BROWN, Daniel R.l.** [CA/CA]; 6033 Paddle Road, Mississauga, Ontario L5N 1X8 (CA).

(74) Agents: **ORANGE, John R.S.** et al.; Blake, Cassels & Graydon Llp, 199 Bay Street, Box 25, Commerce Court West, Toronto, Ontario M5L 1A9 (CA).

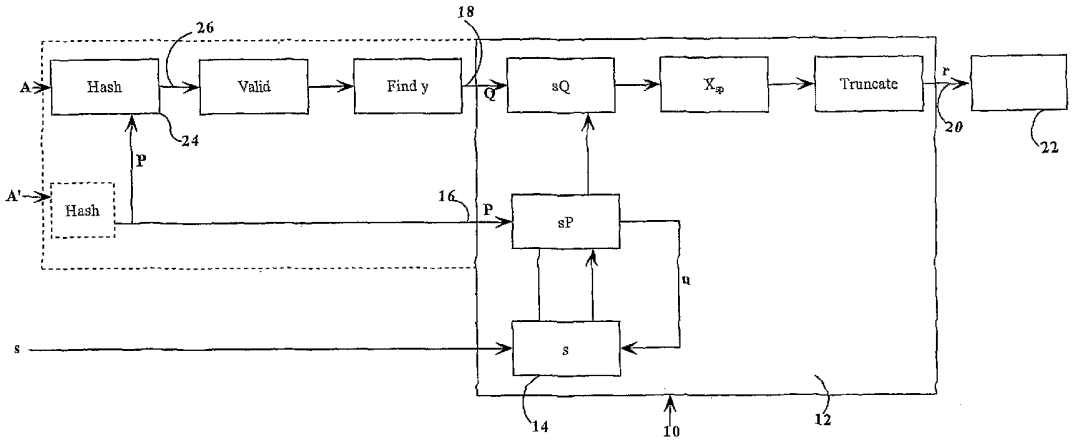
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: ELLIPTIC CURVE RANDOM NUMBER GENERATION



(57) Abstract: An elliptic curve random number generator avoids escrow keys by choosing a point Q on the elliptic curve as verifiably random. An arbitrary string is chosen and a hash of that string computed. The hash is then converted to a field element of the desired field, the field element regarded as the x-coordinate of a point Q on the elliptic curve and the x-coordinate is tested for validity on the desired elliptic curve. If valid, the x-coordinate is decompressed to the point Q, wherein the choice of which is the two points is also derived from the hash value. Intentional use of escrow keys can provide for back up functionality. The relationship between P and Q is used as an escrow key and stored by for a security domain. The administrator logs the output of the generator to reconstruct the random number with the escrow key.

WO 2006/076804 A1

ELLIPTIC CURVE RANDOM NUMBER GENERATION

FIELD OF THE INVENTION:

[0001] The present invention relates to systems and methods for cryptographic random number generation.

DESCRIPTION OF THE PRIOR ART

[0002] Random numbers are utilised in many cryptographic operations to provide underlying security. In public key infrastructures, for example, the private key of a key pair is generated by a random number generator and the corresponding public key mathematically derived therefrom. A new key pair may be generated for each session and the randomness of the generator therefore is critical to the security of the cryptographic system.

[0003] To provide a secure source of random numbers, cryptographically secure pseudorandom bit generators have been developed in which the security of each generator relies on a presumed intractability of the underlying number-theoretical problem. The American National Standards Institute (ANSI) has set up an Accredited Standards Committee (ASC) X9 for the financial services industry, which is preparing a American National Standard (ANS) X9.82 for cryptographic random number generation (RNG). One of the RNG methods in the draft of X9.82, called Dual_EC_DRBG, uses elliptic curve cryptography (ECC) for its security. Dual_EC_DRBG will hereinafter be referred to as elliptic curve random number generation (ECRNG).

[0004] Elliptic curve cryptography relies on the intractability of the discrete log problem in cyclic subgroups of elliptic curve groups. An elliptic curve E is the set of points (x, y) that satisfy the defining equation of the elliptic curve. The defining equation is a cubic equation, and is non-singular. The coordinates x and y are elements of a field, which is a set of elements that can be added, subtracted and divided, with the exception of zero. Examples of fields include rational

numbers and real numbers. There are also finite fields, which are the fields most often used in cryptography. An example of a finite field is the set of integers modulo *a* prime *q*.

[0005] Without the loss of generality, the defining equation of the elliptic curve can be in the Weierstrass form, which depends on the field of the coordinates. When the field F is integers modulo a prime $q > 3$, then the Weierstrass equation takes the form $y^2 = x^3 + ax + b$, where a and b are elements of the field F.

[0006] The elliptic curve E includes the points (x, y) and one further point, namely the point O at infinity. The elliptic curve E also has a group structure, which means that the two points P and Q on the curve can be added to form a third point $P + Q$. The point O is the identity of the group, meaning $P + O = O + P = P$, for all points P. Addition is associative, so that $P + (Q + R) = (P + Q) + R$, and commutative, so that $P + Q = Q + R$, for all points P, Q and R. Each point P has a negative point $-P$, such that $P + (-P) = O$. When the curve equation is the Weierstrass equation of the form $y^2 = x^3 + ax + b$, the negative of $P = (x, y)$ is determined easily as $-P = (x, -y)$. The formula for adding points P and Q in terms of their coordinates is only moderately complicated involving just a handful of field operations.

[0007] The ECRNG uses as input two elliptic curve points P and Q that are fixed. These points are not assumed to be secret. Typically, P is the standard generator of the elliptic curve domain parameters, and Q is some other point. In addition a secret seed is inserted into the ECRNG.

[0008] The ECRNG has a state, which may be considered to be an integer s. The state s is updated every time the ECRNG produces an output. The updated state is computed as $u = z(sP)$, where $z()$ is a function that converts an elliptic curve point to an integer. Generally, z consists of taking the x-coordinate of the point, and then converting the resulting field element to an integer. Thus u will typically be an integer derived from the x-coordinate of the point s.

[0009] The output of the ECRNG is computed as follows: $r = t(z(sQ))$, where t is a truncation function. Generally the truncation function removes the leftmost bits of its input. In the

ECRNG, the number of bits truncated depends on the choice of elliptic curve, and typically may be in the range of 6 to 19 bits.

[0010] Although P and Q are known, it is believed that the output r is random and cannot be predicted. Therefore successive values will have no relationship that can be exploited to obtain private keys and break the cryptographic functions. The applicant has recognised that anybody who knows an integer d such that $Q = dP$, can deduce an integer e such that $ed = 1 \bmod n$, where n is the order of G, and thereby have an integer e such that $P = eQ$. Suppose $U = sP$ and $R = sQ$, which are the precursors to the updated state and the ECRNG output. With the integer e, one can compute U from R as $U = eR$. Therefore, the output $r = t(z(R))$, and possible values of R can be determined from r. The truncation function means that the truncated bits of R would have to be guessed. The z function means that only the x-coordinate is available, so that decompression would have to be applied to obtain the full point R. In the case of the ECRNG, there would be somewhere between about $2^6 = 64$ and 2^{19} (i.e. about half a million) possible points R which correspond to r, with the exact number depending on the curve and the specific value of r.

[0011] The full set of R values is easy to determine from r, and as noted above, determination of the correct value for R determines $U = eR$, if one knows e. The updated state is $u = z(U)$, so it can be determined from the correct value of R. Therefore knowledge of r and e allows one to determine the next state to within a number of possibilities somewhere between 2^6 and 2^{19}. This uncertainty will invariably be eliminated once another output is observed, whether directly or indirectly through a one-way function.

[0012] Once the next state is determined, all future states of ECRNG can be determined because the ECRNG is a deterministic function. (at least unless additional random entropy is fed into the ECRNG state) All outputs of the ECRNG are determined from the determined states of the ECRNG. Therefore knowledge of r and e, allows one to determine all future outputs of the ECRNG.

[0013] It has therefore been identified by the applicant that this method potentially possesses a trapdoor, whereby standardizers or implementers of the algorithm may possess a piece of information with which they can use a single output and an instantiation of the RNG to

determine all future states and output of the RNG, thereby completely compromising its security. It is therefore an object of the present invention to obviate or mitigate the above mentioned disadvantages.

SUMMARY OF THE INVENTION

[0014] In one aspect, the present invention provides a method for computing a verifiably random point Q for use with another point P in an elliptic curve random number generator comprising computing a hash including the point P as an input, and deriving the point Q from the hash.

[0015] In another aspect, the present invention provides a method for producing an elliptic curve random number comprising generating an output using an elliptic curve random number generator, and truncating the output to generate the random number.

[0016] In yet another aspect, the present invention provides a method for producing an elliptic curve random number comprising generating an output using an elliptic curve random number generator, and applying the output to a one-way function to generate the random number.

[0017] In yet another aspect, the present invention provides a method of backup functionality for an elliptic curve random number generator, the method comprising the steps of computing an escrow key e upon determination of a point Q of the elliptic curve, whereby $P = eQ$, P being another point of the elliptic curve; instituting an administrator, and having the administrator store the escrow key e; having members with an elliptic curve random number generator send to the administrator, an output r generated before an output value of the generator; the administrator logging the output r for future determination of the state of the generator.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] An embodiment of the invention will now be described by way of example only with reference to the appended drawings wherein:

[0019] Figure 1 is a schematic representation of a cryptographic random number generation scheme.

[0020] Figure 2 is a flow chart illustrating a selection process for choosing elliptic curve points.

[0021] Figure 3 is a block diagram, similar to figure 1 showing a further embodiment

[0022] Figure 4 is flow chart illustrating the process implemented by the apparatus of Figure 3.

[0023] Figure 5 is a block diagram showing a further embodiment.

[0024] Figure 6 is a flow chart illustrating yet another embodiment of the process of Figure 2.

[0025] Figure 7 is schematic representation of an administrated cryptographic random number generation scheme.

[0026] Figure 8 is a flow chart illustrating an escrow key selection process.

[0027] Figure 9 is a flow chart illustrating a method for securely utilizing an escrow key.

DETAILED DESCRIPTION OF THE INVENTION

[0028] Referring therefore to Figure 1, a cryptographic random number generator (ECRNG) 10 includes an arithmetic unit 12 for performing elliptic curve computations. The ECRNG also includes a secure register 14 to retain a state value s and has a pair of inputs 16, 18 to receive a

pair of initialisation points P, Q. The points P, Q are elliptic curve points that are assumed to be known. An output 20 is provided for communication of the random integer to a cryptographic module 22. The initial contents of the register 14 are provided by a seed input S.

[0029] This input 16 representing the point P is in a first embodiment, selected from a known value published as suitable for such use.

[0030] The input 18 is obtained from the output of a one way function in the form of a hash function 24 typically a cryptographically secure hash function such as SHA1 or SHA2 that receives as inputs the point P. The function 24 operates upon an arbitrary bit string A to produce a hashed output 26. The output 26 is applied to arithmetic unit 12 for further processing to provide the input Q.

[0031] In operation, the ECRNG receives a bit string as a seed, which is stored in the register 14. The seed is maintained secret and is selected to meet pre-established cryptographic criteria, such as randomness and Hamming weight, the criteria being chosen to suit the particular application.

[0032] In order to ensure that d is not likely to be known (e.g. such that $P = dQ$, and $ed = 1 \bmod n$); one or both of the inputs 16, 18 is chosen so as to be verifiably random. In the embodiment of Figure 1, Q is chosen in a way that is verifiably random by deriving it from the output of a hash-function 24 (preferably one-way) whose input includes the point P. As shown in Figure 2 an arbitrary string A is selected at step 202, a hash H of A is computed at step 204 with P and optionally S as inputs to a hash-based function $F_H()$, and the hash H is then converted by the arithmetic unit 12 to a field element X of a desired field F at step 206. P may be pre-computed or fixed, or may also be chosen to be a verifiably random chosen value. The field element X is regarded as the x-coordinate of Q (thus a "compressed" representation of Q). The x-coordinate is then tested for validity on the desired elliptic curve E at step 208, and whether or not X is valid, is determined at step 210. If valid, the x-coordinate provided by element X is decompressed to provide point Q at step 212. The choice of which of two possible values of the y co-ordinate is generally derived from the hash value.

[0033] The points P and Q are applied at respective inputs 16, 18 and the arithmetic unit 12 computes the point sQ where s is the current value stored in the register 14. The arithmetic unit 12 converts the x-coordinate of the point (in this example point sQ) to an integer and truncates the value to obtain $r = t(z(sQ))$. The truncated value r is provided to the output 20.

[0034] The arithmetic unit 12 similarly computes a value to update the register 14 by computing sP, where s is the value of the register 14, and converting the x-coordinate of the point sP to an integer u. The integer u is stored in the register to replace s for the next iteration. {ditto above}

[0035] As noted above, the point P may also be verifiably random, but may also be an established or fixed value. Therefore, the embodiment of Figure 1 may be applied or retrofitted to systems where certain base points (e.g. P) are already implemented in hardware. Typically, the base point P will be some already existing base point, such as those recommended in Federal Information Processing Standard (FIPS) 186-2. In such cases, P is not chosen to be verifiably random.

[0036] In general, inclusion of the point P in the input to the hash function ensures that P was determined before Q is determined, by virtue of the one-way property of the hash function and since Q is derived from an already determined P. Because P was determined before Q, it is clearly understood that P could not have been chosen as a multiple of Q (e.g. where $P = eQ$), and therefore finding d is generally as hard as solving a random case of the discrete logarithm problem.

[0037] Thus, having a seed value S provided and a hash-based function $F()$ provided, a verifier can determine that $Q = F(S,P)$, where P may or may not be verifiably random. Similarly, one could compute $P = F(S,Q)$ with the same effect, though it is presumed that this is not necessary given that the value of P in the early drafts of X9.82 were identical to the base points specified in FIPS 186-2.

[0038] The generation of Q from a bit string as outlined above may be performed externally of the ECRNG 10, or, preferably, internally using the arithmetic unit 12. Where both P and Q

are required to be verifiably random, a second hash function 24 shown in ghosted outline in Figure 1 is incorporated to generate the coordinate of point P from the bit string A. By providing a hash function for at least one of the inputs, a verifiably random input is obtained.

[0039] It will also be noted that the output generated is derived from the x coordinate of the point sP. Accordingly, the inputs 16, 18 may be the x coordinates of P and Q and the corresponding values of sP and sQ obtained by using Montgomery multiplication techniques thereby obviating the need for recovery of the y coordinates.

[0040] An alternative method for choosing Q is to choose Q in some canonical form, such that its bit representation contains some string that would be difficult to produce by generating $Q = dP$ for some known d and P for example a representation of a name. It will be appreciated that intermediate forms between this method and the preferred method may also exist, where Q is partly canonical and partly derived verifiably at random. Such selection of Q, whether verifiably random, canonical, or some intermediate, can be called verifiable.

[0041] Another alternative method for preventing a key escrow attack on the output of an ECRNG, shown in Figures 3 and 4 is to add a truncation function 28 to ECRNG 10 to truncate the ECRNG output to approximately half the length of a compressed elliptic curve point. Preferably, this operation is done in addition to the preferred method of Figure 1 and 2, however, it will be appreciated that it may be performed as a primary measure for preventing a key escrow attack. The benefit of truncation is that the list of R values associated with a single ECRNG output r is typically infeasible to search. For example, for a 160-bit elliptic curve group, the number of potential points R in the list is about 2^{80}, and searching the list would be about as hard as solving the discrete logarithm problem. The cost of this method is that the ECRNG is made half as efficient, because the output length is effectively halved.

[0042] Yet another alternative method shown in Figure 5 and 6 comprises filtering the output of the ECRNG through another one-way function F_{H2},, identified as 34, such as a hash function to generate a new output. Again, preferably, this operation is performed in addition to the preferred method shown in Figure 2, however may be performed as a primary measure to prevent key escrow attacks. The extra hash is relatively cheap compared to the elliptic curve operations

performed in the arithmetic unit 12, and does not significantly diminish the security of the ECRNG.

[0043] As discussed above, to effectively prevent the existence of escrow keys, a verifiably random Q should be accompanied with either a verifiably random P or a pre-established P. A pre-established P may be a point P that has been widely publicized and accepted to have been selected before the notion of the ECRNG 12, which consequently means that P could not have been chosen as $P = eQ$ because Q was not created at the time when P was established.

[0044] Whilst the above techniques ensure the security of the system using the ECRNG by "closing" the trap door, it is also possible to take advantage of the possible interdependence of P and Q, namely where $P = eQ$, through careful use of the existence of e.

[0045] In such a scenario, the value e may be regarded as an escrow key. If P and Q are established in a security domain controlled by an administrator, and the entity who generates Q for the domain does so with knowledge of e (or indirectly via knowledge of d). The administrator will have an escrow key for every ECRNG that follows that standard.

[0046] Escrow keys are known to have advantages in some contexts. They can provide a backup functionality. If a cryptographic key is lost, then data encrypted under that key is also lost. However, encryption keys are generally the output of random number generators. Therefore, if the ECRNG is used to generate the encryption key K, then it may be possible that the escrow key e can be used to recover the encryption key K. Escrow keys can provide other functionality, such as for use in a wiretap. In this case, trusted law enforcement agents may need to decrypt encrypted traffic of criminals, and to do this they may want to be able to use an escrow key to recover an encryption key.

[0047] Figure 7 shows a domain 40 having a number of ECRNG's 10 each associated with a respective member of the domain 40. The domain 40 communicates with other domains 40a, 40b, 40c through a network 42, such as the internet. Each ECRNG of a domain has a pair of identical inputs P,Q. The domain 40 includes an administrator 44 who maintains in a secure manner an escrow key e.

[0048] The administrator 44 chooses the values of P and Q such that he knows an escrow key e such that $Q = eP$. Other members of the domain 40 use the values of P and Q, thereby giving the administrator 44 an escrow key e that works for all the members of the organization.

[0049] This is most useful in its backup functionality for protecting against the loss of encryption keys. Escrow keys e could also be made member-specific so that each member has its own escrow e' from points selected by the administrator 44.

[0050] As generally denoted as numeral 400 in Figure 8, the administrator initially selects a point P which will generally be chosen as the standard generator P for the desired elliptic curve 402. The administrator then selects a value d and the point Q will be determined as $Q = dP$ 404, for some random integer d of appropriate size. The escrow key e is computed as $e = d^{-1} \bmod n$ 406, where n is the order of the generator P and stored by the administrator.

[0051] The secure use of such an escrow key 34e is generally denoted by numeral 500 and illustrated in Figure 9. The administrator 44 is first instituted 502 and an escrow keys e would be chosen and stored 504 by the administrator44

[0052] In order for the escrow key to function with full effectiveness, the escrow administrator 44 needs direct access to an ECRNG output value r that was generated before the ECRNG output value k (i.e. 16) which is to be recovered. It is not sufficient to have indirect access to r via a one-way function or an encryption algorithm. A formalized way to achieve this is to have each member with an ECRNG 12 communicate with the administrator 44 as indicated at 46 in figure 7. and step 506 in figure 9. This may be most useful for encrypted file storage systems or encrypted email accounts. A more seamless method may be applied for cryptographic applications. For example, in the SSL and TLS protocols, which are used for securing web (HTTP) traffic, a client and server perform a handshake in which their first actions are to exchange random values sent in the clear.

[0053] Many other protocols exchange such random values, often called nonces. If the escrow administrator observes these nonces, and keeps a log of them 508, then later it may be able to determine the necessary r value. This allows the administrator to determine the

subsequent state of the ECRNG 12 of the client or server 510 (whoever is a member of the domain), and thereby recover the subsequent ECRNG 12 values. In particular, for the client who generally generates a random pre-master secret from which is derived the encryption key for the SSL or TLS session, the escrow key may allow recovery of the session key. Recovery of the session key allows recovery of the whole SSL or TLS session.

[0054] If the session was logged, then it may be recovered. This does not compromise long-term private keys, just session keys obtained from the output of the ECRNG, which should alleviate any concern regarding general suspicions related to escrows.

[0055] Whilst escrow keys are also known to have disadvantages in other contexts, their control within specific security domains may alleviate some of those concerns. For example, with digital signatures for non-repudiation, it is crucial that nobody but the signer has the signing key, otherwise the signer may legitimately argue the repudiation of signatures. The existence of escrow keys means the some other entity has access to the signing key, which enables signers to argue that the escrow key was used to obtain their signing key and subsequently generate their signatures. However, where the domain is limited to a particular organisation or part of an organisation it may be sufficient that the organisation cannot repudiate the signature. Lost signing keys do not imply lost data, unlike encryption keys, so there is little need to backup signing keys.

[0056] Although the invention has been described with reference to certain specific embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention as outlined in the claims appended hereto.

What is claimed is:

1. A method of computing a random number for use in a cryptographic operation comprising the steps of providing a pair of inputs to an elliptic curve random number generator with each input representative of at least one coordinate of an elliptic curve point and with at least one of said inputs being verifiably random.

2. A method according to claim 1 wherein said at least one input is obtained from an output of a hash function.

3. A method according to claim 2 wherein the other of said inputs is utilized as an input to said hash function.

4. A method according to claim 1 wherein said random number generator has a secret value and said secret value is used to compute scalar multiples of said points represented by said inputs.

5. A method according to claim 4 wherein one of said scalar multiples is used to derive said random number and the other of said scalar multiples is used to change said secret value for subsequent use.

6. A method according to claim 2 wherein said output of said hash function is validated as a coordinate of a point on an elliptic curve prior to utilization as said input.

7. A method according to claim 6 wherein another coordinate of said point is obtained from said one coordinate for inclusion as said input.

8. A method according to claim 7 wherein said other input is a representation of an elliptic curve point.

9. A method according to claim 5 wherein said random number is derived from said scalar multiple by selecting one coordinate of said point represented by said scalar multiple and truncating said coordinate to a bit string for use as said random number.

10. A method according to claim 9 wherein said one coordinate is truncated in the order of one half the length of a representation of an elliptic curve point representation.

11. A method according to claim 5 wherein said random number is derived from said scalar multiple by selecting one coordinate of said point represented by said scalar multiple and hashing said one coordinate to provide a bit string for use as said random number.

12. A method according to claim 1 wherein said verifiably random input is chosen to be of a canonical form whereby a predetermined relationship between said inputs is difficult to maintain.

13. A method of computing a random number for use in a cryptographic operation, said method comprising the steps of providing a pair of inputs, each representative of at least one coordinate of a pair of elliptic curve points to an elliptic curve random number generator, obtaining an output representative of at least one coordinate of a scalar multiple of an elliptic curve point and passing said output through a one way function to obtain a bit string for use as a random number.

14. A method according to claim 13 wherein said one way function is a hash function.

15. An elliptic curve random number generator having a pair of inputs each representative of at least one coordinate of a pair of elliptic curve points and an output for use as a random number in a cryptographic operation, at least one of said inputs being verifiably random.

16. An elliptic curve random number generator according to claim 15 wherein said one input is derived from an output of a one way function.

17. An elliptic curve random number generator according to claim 16 wherein said one way function is a hash function.

18. An elliptic curve random number generator according to claim 17 wherein the other of said inputs is provided as an input to said hash function.

19. A method of establishing an escrow key for a security domain within a network, said method comprising the steps of establishing a pair of points PQ as respective inputs to an elliptic curve random number generator with a relationship between said point such that $P = eQ$, storing said relationship e as an escrow key with an administrator and generating from said elliptic curve random number generator a random number for use in cryptographic operations within said domain.

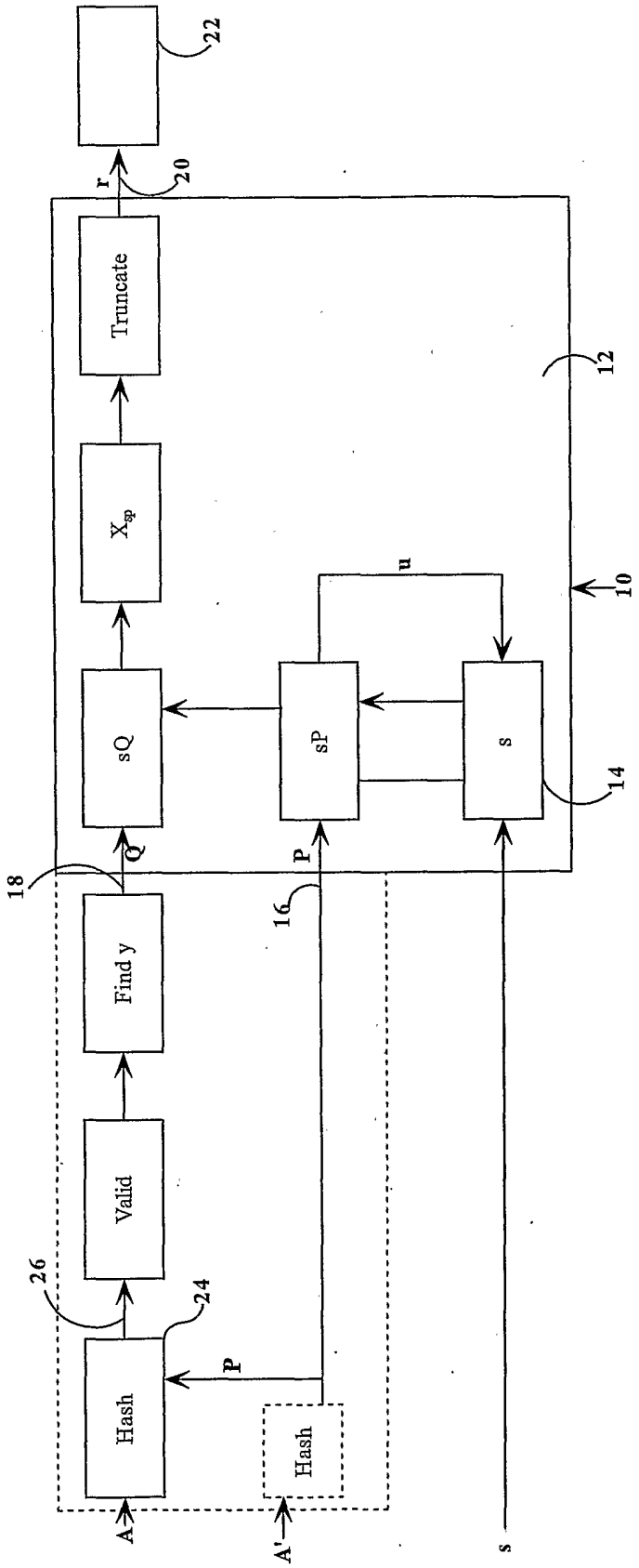

22
r
20
Truncate
12
X_{sp}
u
10
sQ
sP
s
14
Q
P
18
16
Find y
Valid
26
24
P
Hash
Hash
A
A'
s

FIGURE 1

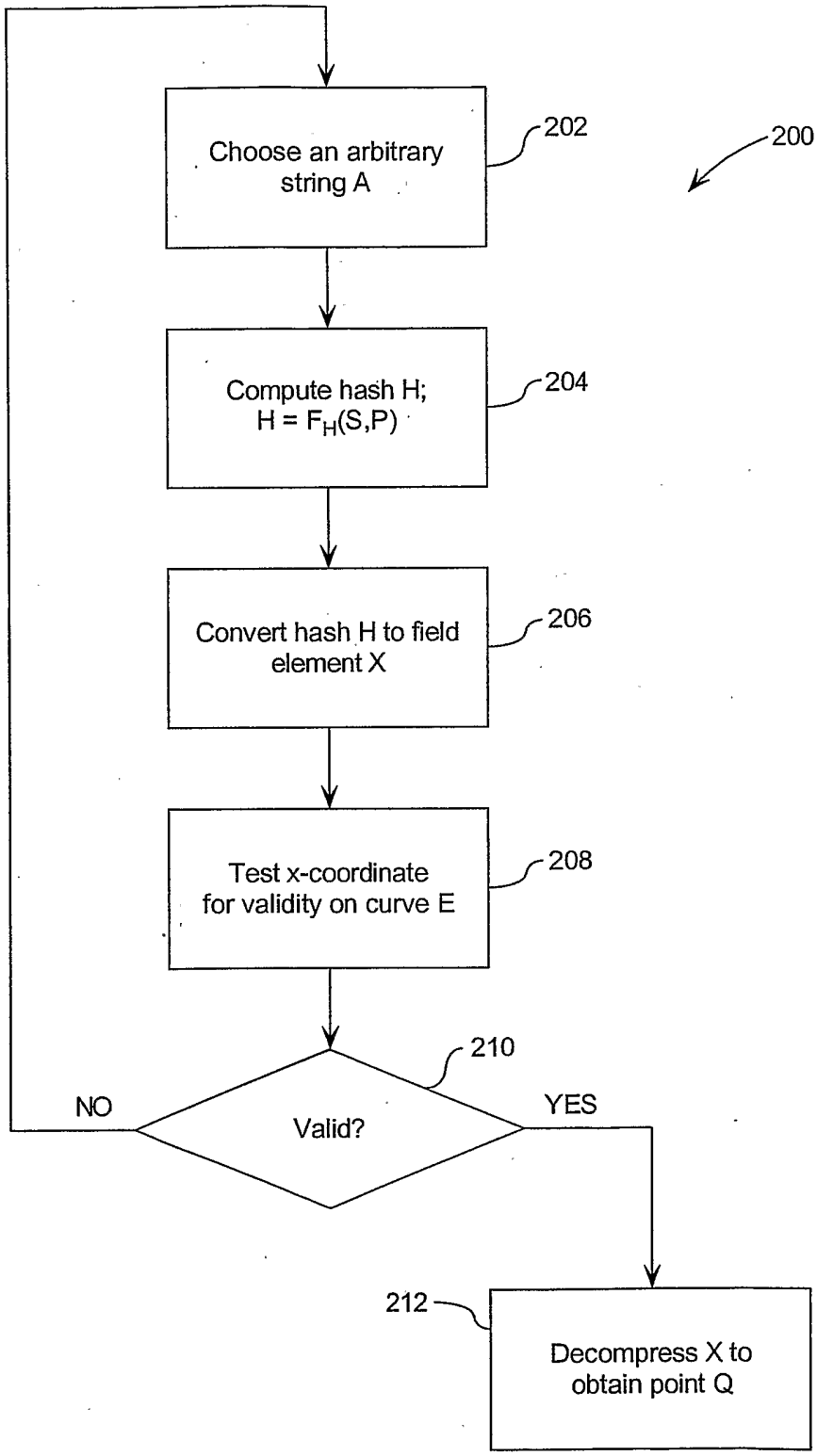

Choose an arbitrary string A
202
200
Compute hash H; H = F_H(S,P)
204
Convert hash H to field element X
206
Test x-coordinate for validity on curve E
208
210
Valid?
NO
YES
212
Decompress X to obtain point Q

Figure 2

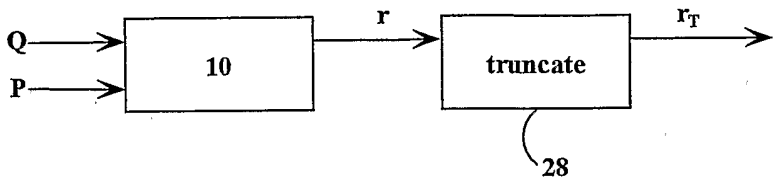


FIGURE 3

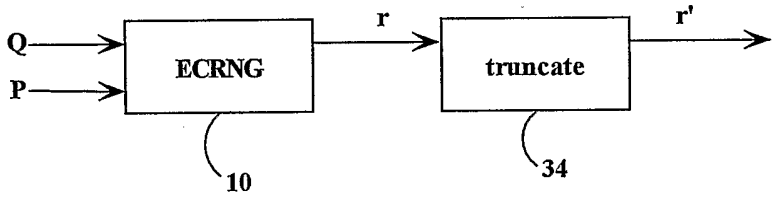


FIGURE 5

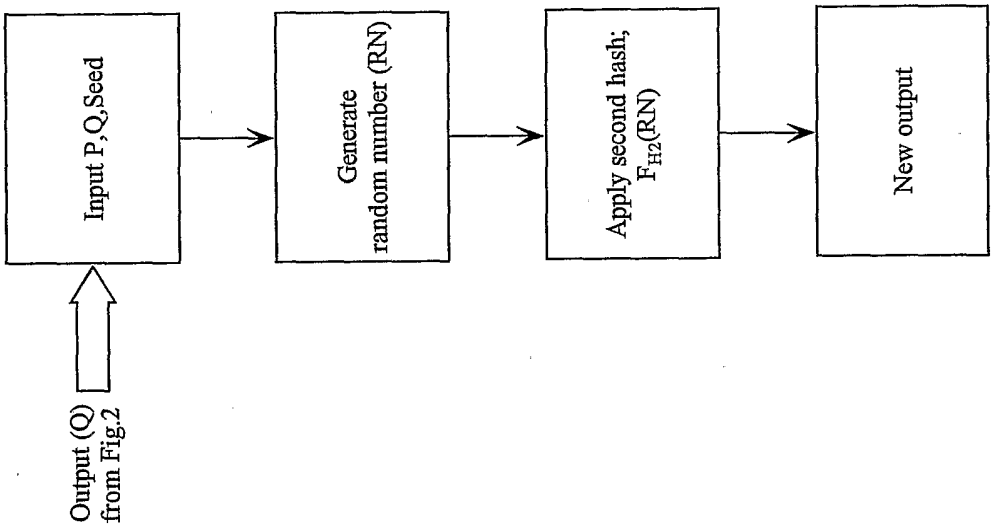


Figure 6

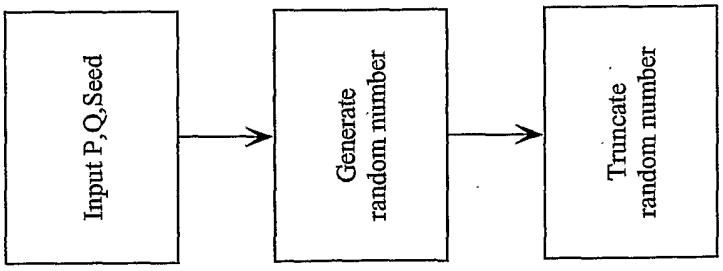


Figure 4

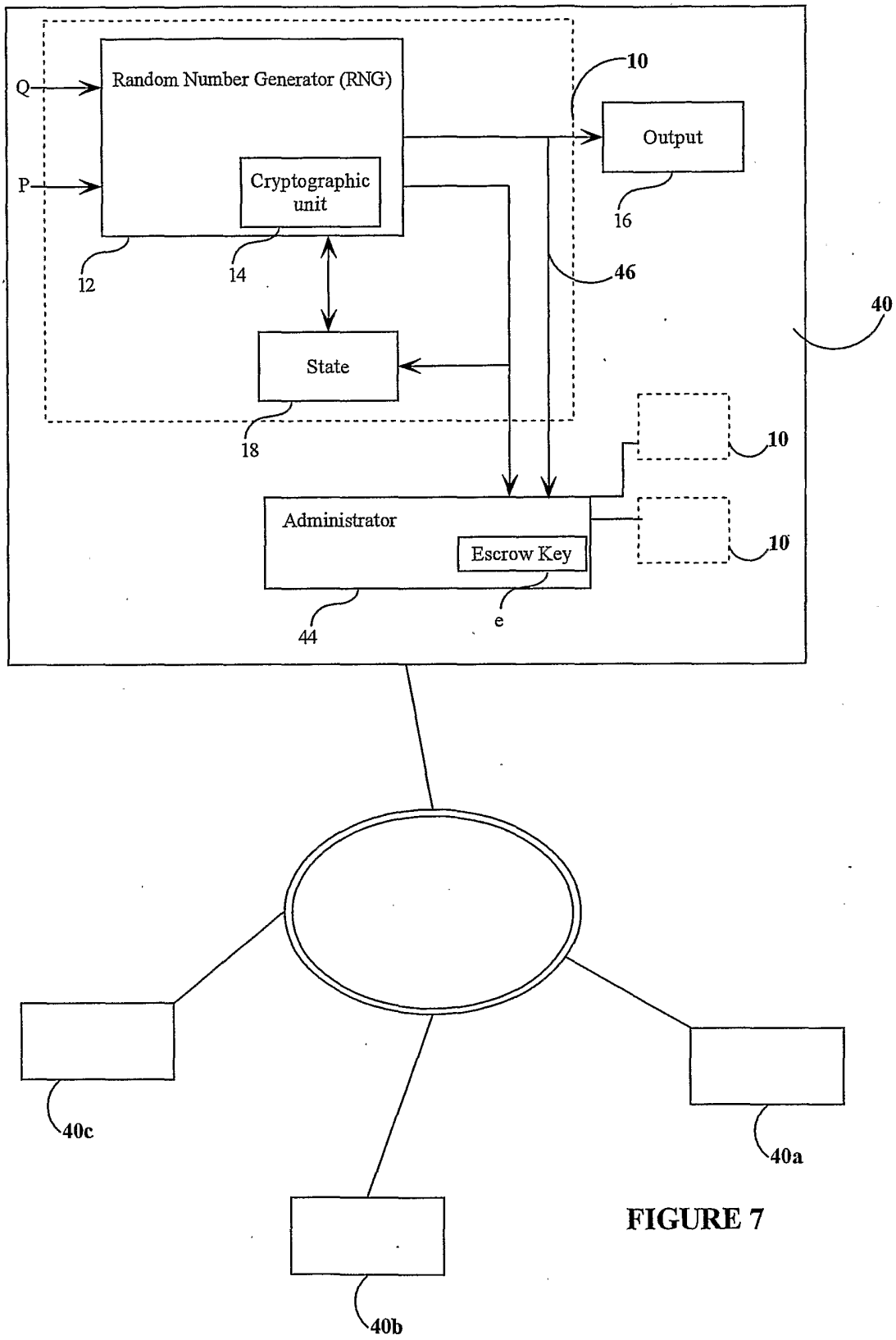

Q
P
Random Number Generator (RNG)
Cryptographic unit
10
Output
16
12
14
46
40
State
18
10
Administrator
Escrow Key
10
44
e
40c
40a
40b

FIGURE 7

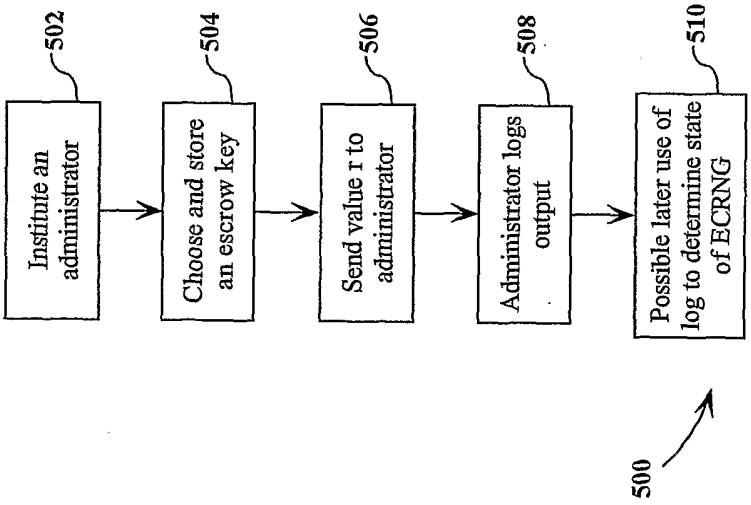


Figure 9

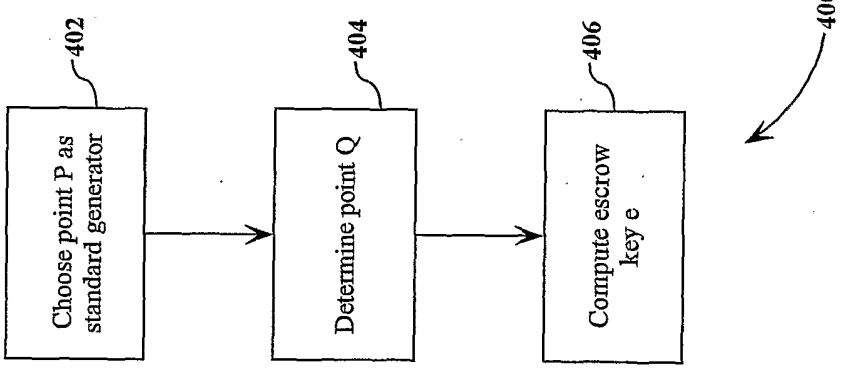


Figure 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2006/000065

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***G06F 7/58*** (2006.01) , ***H04L 9/28*** (2006.01)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC: *G06F 7/00* (2006.01); *G06F 7/58* (2006.01); *H04L 9/28* (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Databases used: Canadian Patent Database; USPTO West (full-text patent database, pre-grant publication, EPO/JPO abstracts); Esp@cenet; and, IEEE Xplore.
Search words used: elliptic curve, random number generator, cryptography, scalar multiple(s), verifiably random, escrow key, truncate, and hash function.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Categor	Citation of document, with indication, where appropriate, of the	Relevant to claim No.
Y	Lee et al., "Elliptic Curve Random Number Generation" *Electrical and Electronic Technology, 2001. Tencon. Proceedings of IEEE Region 10 International Conference on 19-22 August 2001.* Volume 1, pages 239 to 241. entire document	1-4, 12-19
Y	Kaliski, B S., "A Pseudo-Random Bit Generator Based On Elliptic Logarithms". *Advances in Cryptology, CRYPTO 1986.* Volume 263, pages 84 to 103, 1987. entire document	1-4, 12-19
Y	US 2004/0102242 (Poelmann) 27 May 2004 (27.05.04) abstract; paragraph 9	1-4, 12, 15-18

[X] Further documents are listed in the continuation of

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 April 2006 (10-04-2006)	1 May 2006 (01-05-2006)

Name and mailing address of the ISA/CA	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001(819)953-2476	Reid Mulligan (819) 934-7566

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the	Relevant to claim No.
Y	US 6,044,388 (Debellis et al.) 28 May 2000 (28.05.2000) abstract; column 5, lines 1-8	2, 3, 13, 14, 16-18
Y	US 6,243,467 (Reiter et al.) 5 June 2001 (5.06.2001) column 4, lines 61-67 and column 5, lines 1-11	4, 19

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/CA2006/000065

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US 2004/0102242	27-05-2004	AU 2003288680 A1 WO 2004046911 A2	15-06-2004 03-06-2004
US 6,044,388	28-03-2000	NONE	
US 6,243,467	05-06-2001	NONE	