Please type a plus sign (+) inside this box → [+]

PTO/SB/16 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Daniel Scott	BROWN VANSTONE	Mississauga, Canada Campbellville, Canada

☐ *Additional inventors are being named on the ____ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

VERIFIABLY TRAPDOOR-FREE ELLIPTI CURVE RANDOM NUMBER GENERATION

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☒ Customer Number 27871 → *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☐ Firm or Individual Name					
Address					
Address					
City		State		ZIP	
Country		Telephone		Fax	

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification *Number of Pages* 11
☒ Drawing(s) *Number of Sheets* 4
☐ Application Data Sheet. See 37 CFR 1.76
☐ CD(s), Number
☐ Other (specify)

Total # of sheets 15 = Application Size Fee $0.00

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☐ A check or money order is enclosed to cover the filing fees
☒ The Director is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 02-2553
☐ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($) $200.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.
☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully submitted,

SIGNATURE

TYPED or PRINTED NAME Sean X. Zhang

TELEPHONE 416 863 5839

Date January 21, 2005

REGISTRATION NO. 58,058 *(if appropriate)*

Docket Number: 67539/553

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

P1BLARGE/REV07

VERIFIABLY TRAPDOOR-FREE 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 from it. 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, crytographically secure pseudorandom bit generators have been developed in which the security of each generator relies on a presumed intractibility of the underlying number-theoretic 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 intractibility 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 module 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 module 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 in to 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 sl.

[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 mod 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 preventing an elliptic curve random number generator from admitting escrow keys, the method comprising the steps of choosing an arbitrary string and computing a hash of that string, converting the hash to a field element of the desired field, the field element regarded as the x-coordinate of a point Q on the elliptic curve, testing the x-coordinate for validity on the desired elliptic curve and if valid decompressing the x-coordinate to the point Q, wherein the choice of which is the two points is also derived from the hash value.

[0015] In another aspect of the present invention, the point Q is chosen of some canonical form such that its bit representation has some string that would be difficult to produce by relating the point Q with another point P on the elliptic curve.

[0016] In yet another aspect, the present invention provides a method of backup functionality for an elliptic curve random number generator wherein an escrow key is intentionally used, the method comprising the steps of computing an escrow key upon determination of a point Q of the elliptic curve, instituting an administrator and having the administrator store the escrow key, 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 sent for future determination of the state of the generator.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] The features of the invention will become more apparent in the following detailed description in which reference is made to the appended drawings wherein:

[0018] Figure 1 is a schematic representation of a cryptographic random number generation scheme.

[0019] Figure 2 is a flow chart illustrating a selection process for choosing elliptic curve points.

[0020] Figure 3 is schematic representation of an administrated cryptographic random number generation scheme.

[0021] Figure 4 is a flow chart illustrating an escrow key selection process.

[0022] Figure 5 is a flow chart illustrating a method for securely utilizing an escrow key.

DETAILED DESCRIPTION OF THE INVENTION

[0023] Referring therefore to figure 1, a cryptographic random number generator (ECRNG) 10 includes an authentic unit 12 for performing elliptic curve computations. The ECRNG also includes a secure register 14 to retain a state 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 preselected and assumed to be known. An output 20 is provided for communication of the random integer to a cryptographic module 22.

[0024] In operation, the ECRNG receives a bit string as a seed in the register 14. The seed is maintained secret and is selected to meet established cryptographic criteria.

[0025] 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 value in the register 14. The arithmetic unit 12 converts the x coordinate to an integer and truncates the value to obtain $r = t(z(sQ))$. This is provided to the output 20.

[0026] 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 to an integer u. The integer u is stored in the register for subsequent iterations.

[0027] To inhibit the establishment of a relationship between P and Q that provides a trap door, the point Q is selected as a verifiably random value. This is done most readily by ensuring that Q is derived from a hash value.

[0028] More precisely, one way to choose Q is as follows making reference to Figure 2. An arbitrary string is selected 202, its hash computed 204, the hash is then converted to a field element of the desired field 206, the field element regarded as the x-coordinate of Q, the x-coordinate is tested for validity on the desired elliptic curve 208 and the validity determined 210. If valid, the x-coordinate would be decompressed to a point Q 212, where the choice of which of the two possible points is also derived from the hash value.

[0029] The generation of Q from a bit string may be performed externally of the ECRNG 10, or, preferably, internally using the arithmetic unit 12.

[0030] A less preferred 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 these two methods 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.

[0031] To effectively prevent the existence of escrow keys, a verifiable Q should be accompanied with either a verifiable P or a pre-established P. A pre-established P is 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.

[0032] 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.

[0033] The value e may be regarded as an escrow key. If P and Q are established in a standard, and the entity who generated Q for the standard did so with knowledge of e (or indirectly via knowledge of d), then the entity will have an escrow key for every ECRNG that follows that standard.

[0034] Escrow keys are known to have advantages in some contexts. They can provide a backup functionality, as follows. If a cryptographic key is lost, then data encrypted under that key is also lost. But 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.

[0035] Escrow keys are also known to have disadvantages in other contexts. 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. Lost signing keys do not imply lost data, unlike encryption keys, so there is little need to backup signing keys. Forging signatures is not as useful to law enforcement agents as deciphering encrypted traffic. Escrow keys are sometimes called trapdoors, and are viewed by some with suspicion, especially those who value civil liberties over diligent law enforcement.

[0036] Figure 3 shows an administrated ECRNG 30 having like components to that shown in Figure 1 with the addition of an administrator 32 having an escrow key 34.

[0037] The administrated ECRNG 30 would be most useful for security administrators of organizations, who would have chosen P and Q such that they know an escrow key e such that Q = eP. They could then issue members of the organization instances of the ECRNG 12 in P and Q, thereby giving the administrator 32 an escrow key 34 that works for all the members of the organization.

[0038] This is most useful in its backup functionality for protecting against the loss of encryption keys. Escrow keys 34 could also be made member-specific the method of which is generally denoted as numeral 400 in Figure 4. When doing this, the point P will generally be chosen as the standard generator P for the desired elliptic curve 402, and the point Q will be determined as Q = dP 404, for some random integer d of appropriate size, with the escrow key e computed as $e = d^{-1} \bmod n$ 406, where n is the order of the generator P. The secure use of such an escrow key 34 is generally denoted by numeral 500 and illustrated in Figure 5. An administrator 32 would first need to be instituted 502 and an escrow key 34 would be chosen and stored 504 by the administrator 32 thereby providing a scheme 30 such as that shown in Figure 3.

[0039] In order for the escrow key to function with full effectiveness, the escrow administrator 32 needs 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 send to the administrator 32 such an output r 506. 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.

[0040] 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 organization), 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.

[0041] If the session was logged, then it may be recovered. Note that this does not compromise long-term private keys, just session keys, which should alleviate any concern regarding general suspicions related to escrows.

[0042] 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. The entire disclosures of all references recited above are incorporated herein by reference.

What is claimed is:

1. A method for preventing an elliptic curve random number generator from admitting escrow keys, said method comprising the steps of choosing an arbitrary string and computing a hash of said string; converting said hash to a field element of the desired field, said field element regarded as the x-coordinate of a point Q on the elliptic curve; testing said x-coordinate for validity on the desired elliptic curve, and if valid, decompressing said x-coordinate to the point Q; wherein the choice of which is the two points is also derived from said hash value.

2. A method for preventing an elliptic curve random number generator from admitting escrow keys, said method comprising the steps of choosing a point Q of some canonical form such that its bit representation has some string that would be difficult to produce by relating the point Q with another point P on the elliptic curve.

3. A method of backup functionality for an elliptic curve random number generator wherein an escrow key is intentionally used, said method comprising the steps of computing an escrow key upon determination of a point Q of the elliptic curve; instituting an administrator, and having said administrator store said escrow key; having members with an elliptic curve random number generator send to said administrator, an output r generated before an output value of said generator; said administrator logging said output sent for future determination of the state of said generator.

ABSTRACT

A system and method is provided for assuring that an elliptic curve random number generator does not admit 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. There is also provided an intentional use of escrow keys for back up functionality. An escrow key is computed, an administrator instituted who stores the escrow key, members with a generator sending to the administrator output of the generator and the administrator logging the output for future determination of the state of the generator.

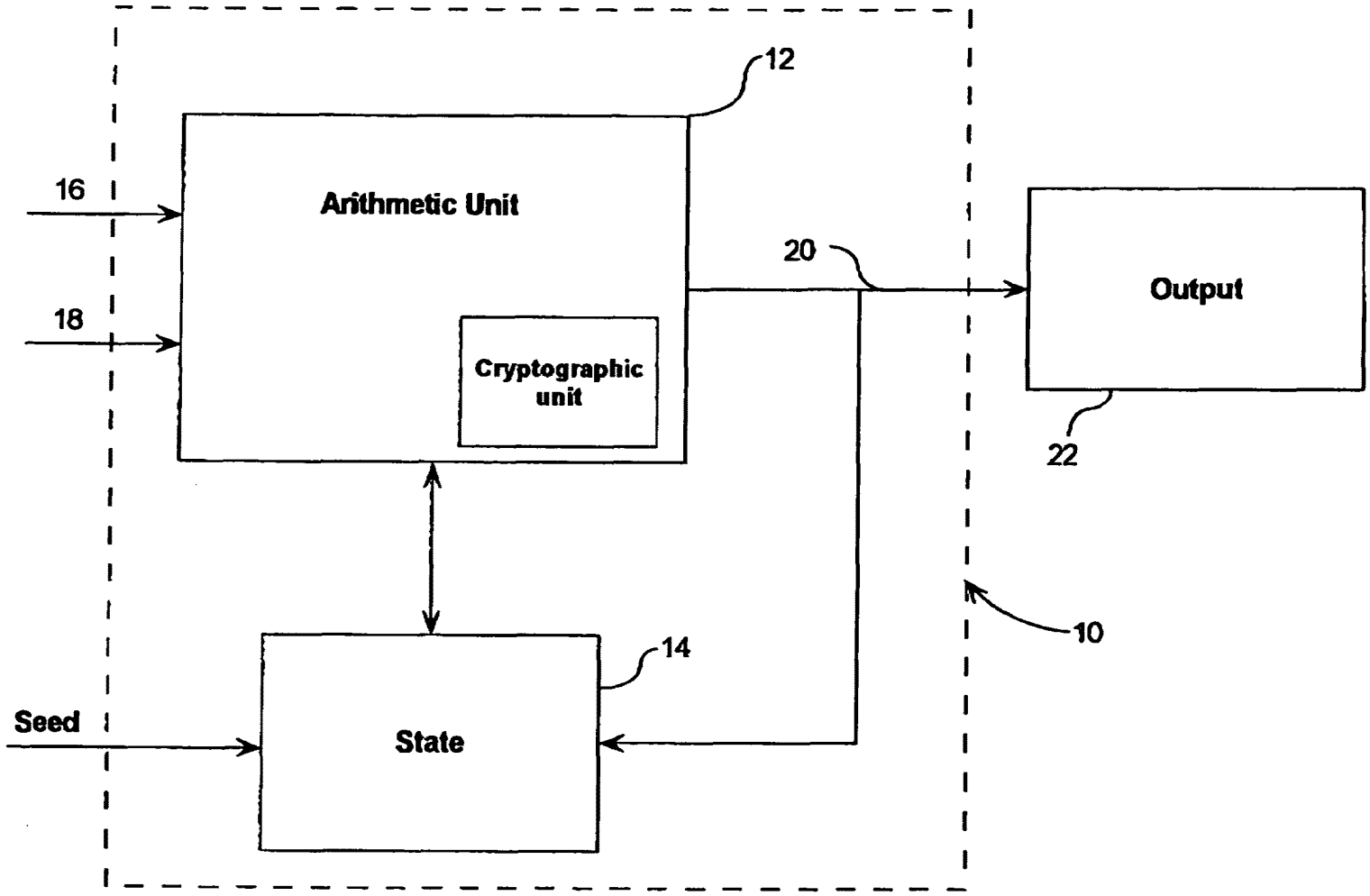

Figure 1

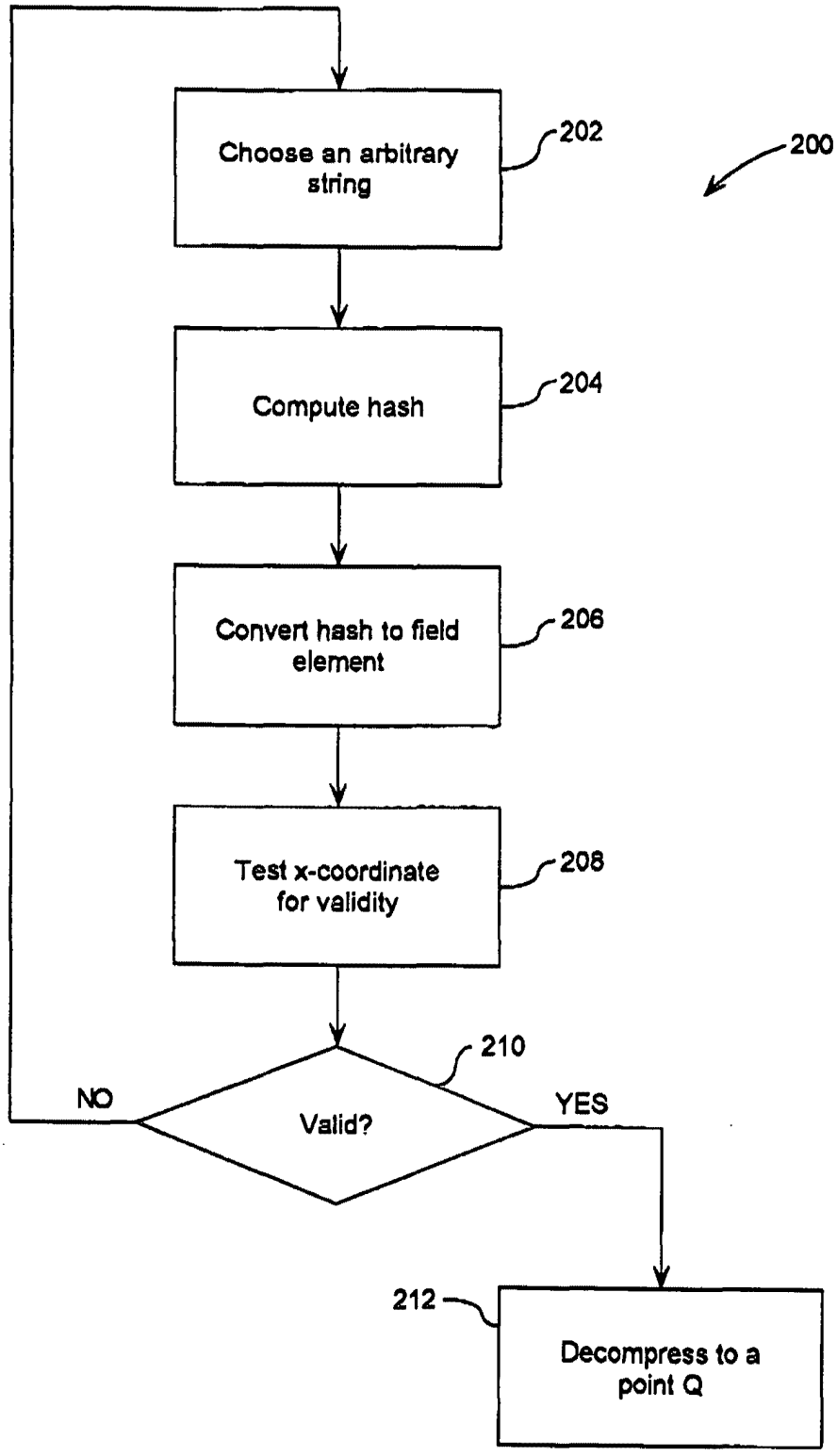

Figure 2

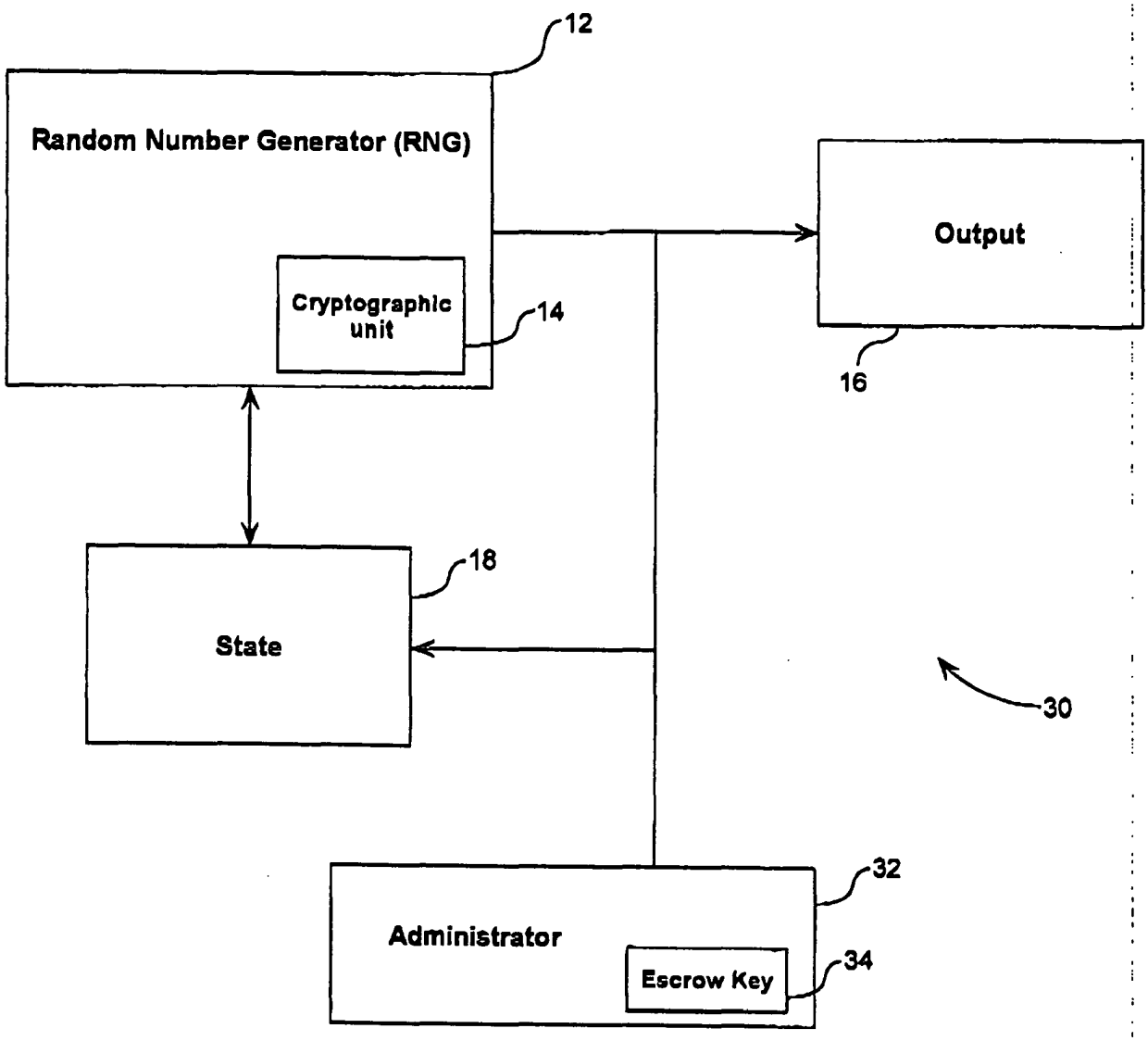

Figure 3

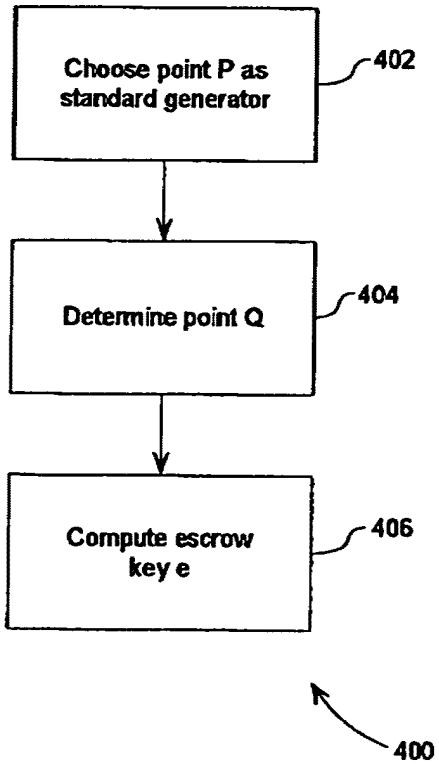

Figure 4

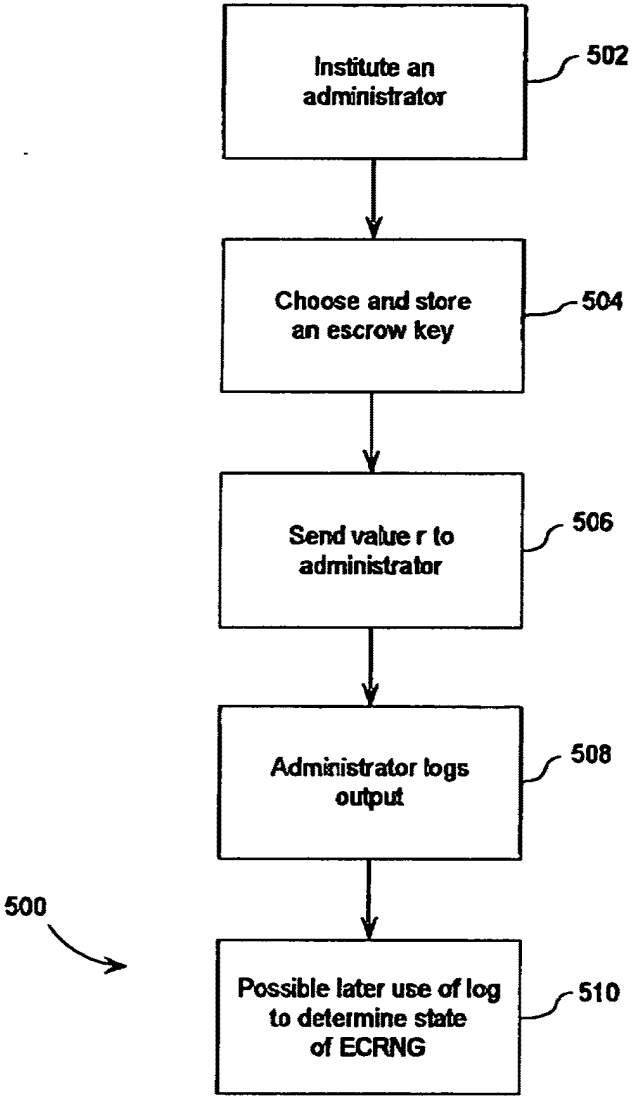

Figure 5

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

02/07/2005 EABUBAK1 00000159 022553 60644982
01 FC:1005 200.00 DA

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 489-267/69033

DEPARTMENT OF DEFENSE

ACCESS ACKNOWLEDGEMENT / SECRECY ORDER RECOMMENDATION

FOR PATENT APPLICATION

Application Serial No: DP60644982 Filing Date: 01/21/2005 Date Referred: 03/07/2005

I hereby acknowledge that the Department of Defense reviewers has inspected this application in administration of 35 USC 181 on befalf of the Agencies/Commands specified below. DoD reviewers will not divulge any information from this application for any purpose other than administration of 35 USC 181.

Defense Agency	Recommendation	Reviewer Name	Reviewer Command	Date Reviewed
NSA	Secrecy Not Recommended	Jennifer Ferragut	reviewer_command	02/02/2006

Type of Recommendations:	*SNR: Secrecy Not Recommended*
	SR: Secrecy Recommended
	NC: No Comment

Instructions to Reviewers:

1. All DoD personnel reviewering this application will be listed on this form regardless of whether they are making a secrecy order recommendation.

2. This form will be forwarded to USPTO once all assigned DoD entities have provided their secrecy order recommendation.

Time for Completion of Review:

Pursuant to 35 USC 184, the subject matter of this applicaiton may be filed in a foreign country for the purposed of filing a patent application without a license anytime after the expriation of six (6) months from filing date unless the application becomes the subject of a secrecy order.

The USPTO publishes patent application at 18 months from the earliest claimed filing date. The USPTO will delay the publication of a patent application made available to a defense agency under 35 USC 181 until no earlier than 6 months from the filing date or 90 days from the date of referral to that agency. This application will be cleared for publication 6 months from the filing date or 90 days from the above Date Referred, whichever is later, unless a response is is provided to the USPTO regarding the necessary recommendations as to the imposition of a secrecy order.

DoD Completion of Review: Final

Forwarded to USPTO: 02/07/2006 By: Laureen Darie