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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 March 2009 (12.03.2009)

PCT

(10) International Publication Number
WO 2009/030021 A1

(51) International Patent Classification:
H04L 9/14 (2006.01) *H04L 9/32* (2006.01)
H04L 9/30 (2006.01)

(21) International Application Number:
PCT/CA2008/001541

(22) International Filing Date:
3 September 2008 (03.09.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/935,855 4 September 2007 (04.09.2007) US

(71) Applicant *(for all designated States except US)*: **CERTICOM CORP.** [CA/CA]; 4th Floor, 5520 Explorer Drive, Mississauga, Ontario L4W 5L1 (CA).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BROWN, Daniel R.** [CA/CA]; 6033 Paddle Road, Mississauga, Ontario L5N 1X8 (CA). **CAMPAGNA, Matthew J.** [US/US]; 264 Old Sib Road, Ridgefield, Connecticut 06877 (US). **STRUIK, Marinus** [CA/CA]; 723 Carlaw Avenue, Toronto, Ontario M4K 3K8 (CA). **VANSTONE, Scott A.** [CA/CA]; 10140 Pineview Trail, Campbellville, Ontario N0P 1B0 (CA).

(74) Agents: **SLANEY, Brett J.** et al.; Box 25, Commerce Court West, 199 Bay Street, Toronto, Ontario M5L 1A9 (CA).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SIGNATURES WITH CONFIDENTIAL MESSAGE RECOVERY

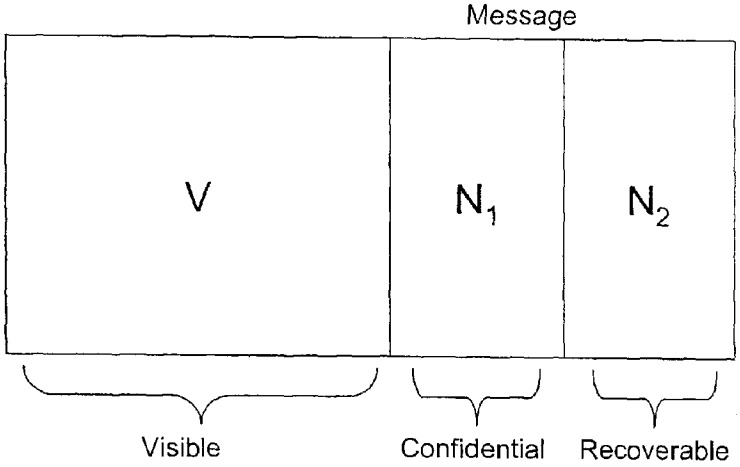

Figure 1

(57) Abstract: A portion of the signed message in an ECPVS is kept truly confidential by dividing the message being signed into at least three parts, wherein one portion is visible, another portion is recoverable by any entity and carries the necessary redundancy for verification, and at least one additional portion is kept confidential. The additional portion is kept confidential by encrypting such portion using a key generated from information specific to that verifying entity. In this way, any entity with access to the signer's public key can verify the signature by checking for a specific characteristic, such as a certain amount of redundancy in the one recovered portion, but cannot recover the confidential portion, only the specific entity can do so. Message recovery is also provided in an elliptic curve signature using a modification of the well analyzed ECDSA signing equation instead of, e.g. the Schnorr equation used in traditional PV signature schemes.

WO 2009/030021 A1

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

SIGNATURES WITH CONFIDENTIAL MESSAGE RECOVERY

TECHNICAL FIELD

[0001] The invention relates generally to cryptographic signatures and has particular utility in providing message recovery in such signatures.

BACKGROUND

[0002] In a public key cryptographic scheme, a public/private key pair is selected so that the problem of deriving the private key from the corresponding public key is equivalent to solving a computational problem that is believed to be intractable. One commonly used public key scheme is based on integer factorization in finite groups, in particular the RSA public key system for modulus $n = p \cdot q$, where p and q are primes.

[0003] Other public key schemes are based on the discrete logarithm problem in finite groups, in particular Diffie-Hellman key exchange and the ElGamal protocol in Z_p (p being a prime), and their variants such as the digital signature algorithm (DSA).

[0004] Elliptic curve public key schemes are based on the elliptic curve (EC) discrete logarithm problem, whose hardness is the basis for the security of EC cryptographic (ECC) schemes, including the EC digital signature algorithm (ECDSA). ECC is typically defined over two types of fields, F_p and F_{2^m}, or generally F_q, where the distinction is not important.

[0005] ECC public key schemes are often chosen for being particularly efficient and secure. For instance, it has been demonstrated that smaller parameters can be used in ECC than RSA or other discrete log systems at a given security level. As such, many solutions using ECC have been developed.

[0006] The Elliptic Curve Pintsov-Vanstone Signature (ECPVS) scheme, as presented in the ASC X9.92 Draft, provides a digital signature scheme with partial message recovery. PV signatures can be done in other discrete log implementations, however EC is considered most desirable. The ECPVS scheme has been used to provide a level of confidentiality by enabling a portion of the message being signed to be "hidden" within one of the resultant signature components. However, in order for the hidden portion to remain confidential, the public key of the signer needs to be kept secret. In a closed system, this may be convenient, however, keeping the public key secret is not the norm for public key systems.

[0007] The ECPVS scheme starts with a signer A having a private/public key pair (d_A, G_A) on an elliptic curve, where d_A is a long term private key and G_A is a restricted public key that is shared amongst a select group of verifiers. In the signing algorithm, A signs a message M = N || V, where N is the hidden portion of the message to be signed. The hidden portion has a predefined characteristic (such as a particular format), e.g. by containing a certain level of redundancy, and V is the plain text portion of the message. In ECPVS, the amount of redundancy or other characteristic can be chosen and thus upon recovering the hidden portion N when verifying the signature, the redundancy or other characteristic can be checked to verify the signature. The following summarizes ECPV signature generation.

[0008] 1. Generate an ephemeral key pair (k, Q), where Q = kG is a point on the elliptic curve, and k is a random integer $1 \leq k < n$, and n is the order of the group generated by the elliptic curve base point G.

[0009] 2. Construct a key k_1 = KDF(Q), where KDF is a key derivation function. In general, a key derivation function is used to derive a secret key from a secret value and/or some known information. In ECPVS, KDF takes as an input a point, Q, and possibly other information, and generates an encryption key k_1.

[0010] 3. Compute a first signature component *c* as $c = ENC_{k_1}(N)$, i.e. the encryption of the message N using a key k_1, where ENC is a suitable encryption scheme that takes as an input plaintext (e.g. N) and encrypts it with a key k_1 to produce ciphertext *c*.

[0011] 4. Compute an intermediate component *h* as *h* = Hash(*c* || V), where Hash is a suitable hash function, e.g. SHA1. If preferred, additional information that may be available or become available to parties verifying the signature (in other words information that the verifier needs 'on the side' for verification), e.g. a certificate or identifying information of the signer may be incorporated into *h*.

[0012] 5. Convert the intermediate component *h* to an integer *e*.

[0013] 6. Calculate a second signature component *s* using a suitable signature algorithm, such as the Schnorr algorithm, where: $s = e \cdot d_A + k \bmod n$.

[0014] 7. Output the signature as (*c*, *s*, V) or (*s*, *c*||V).

[0015] The following illustrates ECPV signature verification on a signature (s, $c||V$), when provided with A's genuine public key G_A.

[0016] 1. Compute the intermediate component h, using the component $c||V$ and using the same hash function used in the signing stage and any additional information, such as the identification information of the signer, where: $h = \text{Hash}(c||V)$.

[0017] 2. Convert h to an integer e.

[0018] 3. Compute a representation Q' of the ephemeral public key Q using the integer e, the public key of A, the base point G, and the signature component s, e.g. as $Q' = sG - eG_A$.

[0019] 4. Compute a decryption key k_1' using the same key derivation function KDF used in the signing stage, including the same additional information, namely as $k_1' = \text{KDF}(Q')$.

[0020] 5. Recover a representation N' of the hidden portion N by decrypting the component c using the key derived in step 4 and a complementary decryption function DEC, namely as $N' = DEC_{k_1'}(c)$.

[0021] 6. Check the specified characteristic (such as a particular format) of, e.g., redundancy contained in N'. If N' contains the necessary characteristic such as a certain amount of redundancy, then N' is a valid message and the signature is verified. If N' does not contain the necessary redundancy, then a null and invalid signature is returned.

[0022] The above scheme has been used to hide messages in the signature, in environments where it is reasonable to keep the public key G_A of A secret among a population of verifiers. This requires that the verifiers be trusted and/or controlled such that only they are able to use the public key and thus recover the portion N that is hidden in c. While in certain closed systems this may be plausible for providing confidentiality for the hidden portion to a group in the closed system, it is typically undesirable to have the public key be 'secret'. There is therefore a need to provide true confidentiality in such a system without having to make the public key secret.

SUMMARY OF THE INVENTION

[0023] In one aspect, there is provided a method for generating a signature on a message, the method comprising: dividing a message into a plurality of portions, a first portion of the message to be visible, a second portion of the message to be hidden and confidential such that only a specified entity can recover the second portion of the message; encrypting the second portion of the message using a first encryption key to generate a first signature component, the first encryption key being generated using information specific to the specified entity; generating a second signature component using the first signature component, the first portion of the message and a private key; and preparing the signature comprising the first and second signature components and the first portion of the message.

[0024] In another aspect, there is provided a method of verifying a signature on a message is provided, the message comprising a plurality of portions, a first portion of the message being visible, a second portion of the message hidden and confidential such that only a specified entity can recover the second portion of the message, the method comprising:

obtaining the signature having a first signature component encrypting the second portion of the message using a first encryption key, the first encryption key having been generated using information specific to the specified entity, having a second signature component generated using the first signature component, the first portion of the message and a private key, and having the first portion of the message; if the specified entity, generating a first decryption key using the information specific to the specified entity, a private key of the specified entity, the second signature component and a value derived from the combination of the first signature component and the first portion of the message; and using the first decryption key to decrypt the second portion of the message from the first signature component.

[0025] In yet another aspect, there is provided a method of generating a signature on a message is provided, the method comprising: dividing a message into a plurality of portions, a first portion of the message being visible and a second portion of the message to be hidden and recoverable by any entity; encrypting the second portion of the message using a first encryption key to generate a first signature component; generating a second signature component using the first signature component, the first portion of the message and an element derived from a private key as inputs to an elliptic curve digital signature algorithm

(ECDSA) signing equation; and preparing the signature comprising the first and second signature components and the first portion of the message.

[0026] In yet another aspect, there is provided a method of verifying a signature on a message is provided, the message comprising a plurality of portions, a first portion of the message being visible and a second portion of the message being hidden and recoverable by any entity; the method comprising: obtaining the signature having a first signature component encrypting the second portion of the message using a first encryption key, having a second component generated using the first signature component, the first portion of the message and an element derived from a private key as inputs to an ECDSA signing equation, and having the first portion of the message; computing a first decryption key using the first and second signature components, a public key of a signing entity and a value derived from a combination of the first signature component and the first portion of the message; using the first decryption key to decrypt the second portion of the message from the second signature component; and using the decrypted second portion of the message to verify the signature.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] Embodiments of the invention will now be described by way of example only with reference to the appended drawings wherein:

[0028] Figure 1 is a schematic diagram of a message having visible, confidential, and recoverable portions.

[0029] Figure 2 is a schematic diagram of a cryptographic communication system.

[0030] Figure 3 is a block diagram illustrating construction of an ECPV signature with confidential message recovery.

[0031] Figure 4 is a flow chart illustrating steps in preparing a signature constructed according to the diagram of Figure 3.

[0032] Figure 5 is a flow chart illustrating verification of the signature prepared in Figure 4 with partial message recovery.

[0033] Figure 6 is a flow chart illustrating verification of the signature prepared in Figure 4 with confidential message recovery.

[0034] Figure 7 is a block diagram illustrating construction of an Elliptic Curve Digital Signature with Recovery (ECDSR).

[0035] Figure 8 is a flow chart illustrating steps in preparing a signature constructed according to the diagram of Figure 7.

[0036] Figure 9 is a flow chart illustrating verification of the signature prepared in Figure 8.

[0037] Figure 10 is a block diagram illustrating construction of an ECDSR signature with confidential message recovery.

[0038] Figure 11 is a flow chart illustrating steps in preparing a signature constructed according to the diagram of Figure 10.

[0039] Figure 12 is a flow chart illustrating verification of the signature prepared in Figure 11 with partial message recovery.

[0040] Figure 13 is a flow chart illustrating verification of the signature prepared in Figure 11 with confidential message recovery.

DETAILED DESCRIPTION OF THE DRAWINGS

[0041] It has been recognized that a portion of the signed message can be kept confidential in a cryptographic signature by dividing the message being signed into at least three parts, wherein one portion is visible or plaintext, another portion is hidden and recoverable by any entity having access to the signer's public key and carries a specific characteristic such as a certain amount of redundancy, for verification, and at least one additional portion is also hidden but only recoverable by a specific verifying entity having the necessary secret value (i.e. providing confidential message recovery). The additional portion is kept confidential by encrypting such portion using a key generated from information specific to that verifying entity. In this way, any entity with access to the signer's public key can verify the signature by checking the specified characteristic of the one recovered portion,

but cannot recover the confidential portion, only the specific entity can do so, as the specific entity is the only one with the secret value needed to recover the confidential portion. The confidential message recovery can be implemented in elliptic curve fields or in other fields such as Z_p.

[0042] It has also been recognized that message recovery in an elliptic curve signature can be implemented using a modification of the well analyzed Elliptic Curve Digital Signature Algorithm (ECDSA) signing equation instead of, e.g. the Schnorr equation used in conventional PV signature schemes.

[0043] Turning now to Figure 1, a message 10 having a visible portion V, a confidential portion N_1 and a recoverable portion N_2 (having a certain amount of redundancy) is shown. It will be appreciated that the message 10 may include more than three parts.

[0044] Figure 2 shows a cryptographic communication system 12 in which a signing entity A signs the message 10 using a cryptographic unit 14 and has a private/public ECPVS key pair (d_A, G_A). The entity A may communicate with a first verifying entity B over a communication channel 18, and may communicate with any other verifying entity Z over a communication channel 20, which may be the same as channel 18 or a different communication link. The entity B also has a cryptographic unit 22 and has a private/public ECPVS key pair (d_B, G_B). It can also be seen in Figure 2 that the entity Z has a cryptographic unit 24 which is at least capable of verifying signatures and obtaining a copy of entity A's public key G_A.

[0045] In one embodiment, the entity A signs the message 10 shown in Figure 1 such that entity B is the legitimate recipient and thus can recover the confidential portion N_1 while entity B and any other entity Z can recover N_2 and verify the signature using such recovered data. The portion N_2 is given a certain characteristic, in the following, a certain amount of redundancy, such that by checking the redundancy of the recovered value, the signature can be verified. For example, N_2 could simply be a string of zeros of a certain length. Figures 3 and 4 illustrate the construction of a signature with confidential message recovery performed by entity A.

[0046] For signature generation, entity A uses its private key d_A, entity B's public key G_B, and signs the message 10, having plaintext V and portions N_1 and N_2, which will be

encrypted. Entity A generates an ephemeral key pair (k, Q) and then using k and the public key G_B, constructs a value $Q_B = kG_B$. The value Q_B is used to create an encryption key for encrypting the portion N_1 so that only entity B (or an entity having access to B's private key if applicable) can recover or unlock the confidential information contained in the portion N_1.

[0047] Two encryption keys are computed using a key derivation function: $k_1 = KDF(Q_B)$ and $k_2 = KDF(Q)$. Using the two encryption keys, the recoverable and confidential portions are then encrypted, using a suitable encryption scheme, to generate a pair of corresponding signature components: $c_1 = ENC_{k_1}(N_1)$ and $c_2 = ENC_{k_2}(N_2)$. The encryption scheme ENC takes as input plaintext (N_1, N_2) and encrypts the plaintext using a secret key (k_1, k_2) and produces ciphertext, (c_1, c_2) which are used as signature components.

[0048] An intermediate value h is then computed by hashing a combination (e.g. concatenation) of the pair of signature components c_1 and c_2 and the visible portion V: $h = Hash(c_1||c_2||V)$. Hash is a suitable hash function, e.g. SHA1, that may also incorporate additional information such as identity information of A into h. The value h is then converted into an integer e to be used in computing another signature component s.

[0049] The signature component s, as is done in ECPVS, can be computed using a suitable signing equation such as the Schnorr equation: $s = e \cdot d_A + k \bmod n$. The resultant signature ($s, c_1||c_2||V$) may then be output.

[0050] As discussed above, the portion N_2 can be recovered by entity B or any other entity Z using the public key of the signer A. Figure 5 illustrates a process of partial message recovery, which obtains a representation N_2' of the portion of the message N_2 having redundancy so that the redundancy can be checked to verify the signature. For the purpose of this illustration, it will be assumed that the verifying entity in Figure 5 is entity Z, which cannot recover N_1 because it does not possess the secret key d_B, and thus N_1 remains confidential with respect to entity Z.

[0051] As can be seen from Figure 5, the entity Z obtains the signature ($s, c_1||c_2||V$) and requires the public key G_A of the signing entity A to verify the signature. The intermediate value h is first computed using the same hash function, Hash, and the combination $c_1||c_2||V$ and any additional information used in creating h. The value h is then converted into an

integer e and a representation Q' of the ephemeral key Q is then computed using also the signature component s, the public key G_A, and the point G as: $Q' = sG - eG_A$. Having computed Q', entity Z then uses the same key derivation function KDF to obtain a decryption key $k_2' = KDF(Q')$. The decryption key k_2' and the signature component c_2 are then used, with the complementary decryption function DEC, to recover N_2' from c_2. Having recovered N_2', entity Z then checks for the characteristic, e.g. a certain amount of redundancy ,and accepts or rejects the signature on this basis. As such, if entity Z does not find the proper amount of redundancy, the signature is deemed to be invalid.

[0052] Turning now to Figure 6, a process is shown for entity B to both verify the signature and recover the confidential portion N_1. As can be seen in Figure 6, entity B obtains the signature $(s, c_1||c_2||V)$ and requires the public key G_A of the signing entity A and its own private key d_B, to verify the signature. As above, the intermediate value h is first computed using the same hash function Hash and the combination $c_1||c_2||V$ and any additional information used when creating h. The value h is then converted into an integer e and a representation Q' of the ephemeral key Q is then computed using the signature component s and the public key G_A as: $Q' = sG - eG_A$. It may be recalled from above that the value Q_B was computed using the public key of B, G_B. As such, entity B can compute a representation Q_B' of the value Q_B using its private key d_B, the signature component s, the integer e, the public key G_A, and the point G as: $Q_B' = d_B \cdot sG - d_B \cdot eG_A$. Having computed Q' and Q_B', entity B then uses the same key derivation function KDF to obtain decryption key $k_2' = KDF(Q')$ as above, and also to obtain decryption key $k_1' = KDF(Q_B')$. The decryption keys k_1' and k_2' and the signature components c_1 and c_2 are then used, with the complementary decryption function DEC, to recover N_1' and N_2' from c_1 and c_2 respectively. Having recovered N_1' and N_2', entity B then checks for the proper amount of redundancy in N_2', and accepts or rejects both N_1' and N_2' on this basis, since if the redundancy in N_2' is incorrect, the signature is invalid or has been compromised in some way.

[0053] It can therefore be seen that being able to specify a particular characteristic, which is then encrypted in the recoverable portion (e.g. N_2) in an ECPV signature enables one to check a predictable, recoverable output for verifying the signature. Also, using the public key of entity B to encrypt the confidential portion enables one to limit who/what can recover the confidential portion to a specific entity, in this example, entity B. It will be appreciated

that the embodiment of Figures 3 to 6 can also be implemented using a plurality of portions, e.g. N and V only, wherein the hidden portion N is computed as N_1 in the above and is also used to verify the signature. As such, in general, the message is divided into a plurality of portions.

[0054] As noted above, it has also been recognized that basic message recovery (on a message having a pair of portions, N and V) as provided by traditional ECPVS can also be provided by having inputs into a modification of the well analyzed ECDSA signing equation, hereinafter referred to as Elliptic Curve Digital Signature with Recovery (ECDSR). Previous uses of the ECDSA signing equation have not been able to provide message recovery. In the following embodiment, confidential message recovery is also provided in an ECDSA implementation, using the same principles discussed above for PV signatures.

[0055] Turning now to Figures 7 and 8, the construction of an ECDSR is shown. As can be seen in Figure 7, similar to traditional ECPV signatures, the message is divided into a visible portion V and a recoverable portion N. For signature generation, entity A uses its private key d_A, and signs the message 10, having plaintext V and recoverable portion N, which will be encrypted. Entity A generates an ephemeral key pair (k, Q) and using Q, constructs an encryption key $k_1 = KDF(Q)$, again using a suitable key derivation function. The encryption key k_1 is then used with a suitable encryption function ENC, to encrypt the portion N by computing a first signature component $c = ENC_{k_1}(N)$.

[0056] An intermediate value h is then computed by hashing a combination (e.g. concatenation) of the signature component c and the visible portion V where: $h = Hash(c \| V)$ and Hash is a suitable hash function that takes as an input additional information such as an identity string. The value h is then converted into an integer e, and the signature component c is converted to an integer C to be used in computing another signature component s.

[0057] The signature component s is computed using a modification of the ECDSA signing equation rather than using, e.g., the Schnorr equation, as sometimes used in ECPVS. In this way, s is computed as: $s = k^{-1}(e + C \cdot d_A) \bmod n$, where, e is the integer form of h, C is an integer derived from the signature component c that hides the portion N, and d_A is the long term private key of entity A. The resultant signature (c, s, V) may then be output.

[0058] As noted above, a modified version of the ECDSA signing equation is used in this embodiment. It may be noted that in the ECDSA signing algorithm an ephemeral point kP is generated, and the integer value $\bar{x}_1$ is then derived from the x-coordinate of kP. That value is used to generate a signature component r, which is used in the signing equation to compute $s = k^{-1}(e + dr)$, where e is the integer representation of the hashed message Hash(M), and d is the private key of the signer. The resulting signature is (r,s) for the message M. The verification computes point $X = es^{-1}P + rs^{-1}Q$, where Q is the public key of the signer and equal to dP. As $s = k^{-1}(e + dr)$, then $s^{-1} = k(e+dr)^{-1}$. When the value s^{-1} is substituted into the equation for X, the value $X = s^{-1}eP + s^{-1}rdP = s^{-1}(e+dr)P = k(e+dr)^{-1}(e+dr)P = kP$. Therefore, if $\bar{x}_1'$ is computed from the x-coordinate of the value X, and a value $v = \bar{x}_1' \bmod n$ is calculated, the signature can be validated if $v = r$. Any attempt to create a forged signature should result in a random point X being computed, and so the x-coordinate integer representation $\bar{x}_1'$ and thus the value v, should be different. It can thus be seen that in the ECDSR embodiment, the recipient should recover the same original plaintext value N, from the signature verification routine, i.e. where N' = N, since the verifier computes $R' = es^{-1}G + cs^{-1}G_A = es^{-1}G + Cs^{-1}d_AG = (es^{-1}+Cs^{-1}d_A)G = s^{-1}(e + Cd_A)G$. However, in this embodiment, $s = k^{-1}(e + Cd_A)$, and thus substituting $s^{-1} = k(e + Cd_A)^{-1}$, it can be seen that $s^{-1}(e + Cd_A)G = k(e + Cd_A)^{-1}(e+Cd_A)G = kG = R$. As such, the verifier can compute the same key $k_1 = KDF(R)$ used to perform the encryption and then compute $N' = DEC_{k1}(c)$. Now, the security comes in the fact that any signed message (c, s, V) will be able to obtain a value N' and the verifier verifies that N' looks like (i.e. has the characteristic) that they expect. In one example, the verifier looks for a certain amount of redundancy. In this case, any attempt to forge a signature would effectively make N' look random, and so not have the expected redundancy. Advantages to using this signing equation include: 1) high assurance that this signing equation is secure; and 2) existing ECDSA implementations can be taken advantage of to perform some of the calculations.

[0059] Turning now to Figure 9, verification of the signature (c, s, V) is shown, which can be done by any entity having access to the public key G_A of the entity A. For the purpose of this illustration, it will be assumed that the verifying entity in Figure 9 is entity Z.

[0060] As can be seen from Figure 9, the entity Z obtains the signature (c, s, V) and requires the public key G_A of the signing entity A to verify the signature. The intermediate

value h is first computed using the same hash function Hash and the combination c||V and any additional input information. The value h is then converted into an integer e. Since the ECDSA signing equation was used in Figure 8, a pair of components u_1 and u_2 are then computed as: $u_1 = es^{-1}$ and $u_2 = cs^{-1}$; and such components are then used to compute a value R′ using the public key G_A as: $R' = u_1G + u_2G_A$. The value R′ is then used, along with the same key derivation function KDF, to obtain a decryption key $k_1' = KDF(R')$. The decryption key k_1' is then used with the complimentary decryption function DEC to recover a representation N′ from signature component c as: $N' = DEC_{k_1}(c)$. The value N′ is then checked for the specified characteristic, e.g. for a certain amount of redundancy, and the signature accepted or rejected on this basis. As such, in this example, if entity Z does not find the proper amount of redundancy, the signature is deemed to be invalid.

[0061] It will be appreciated that for the embodiment of Figures 7-9, the equation for generating s in the signature generation algorithm of Figures 7 and 8 can instead be: $s = k^{-1}(C + e \cdot d_A) \bmod n$ (i.e. C and e are interchanged), where the equation for generating R′ would instead be: $R' = u_2G + u_1G_A$.

[0062] The ECDSR scheme discussed above can be extended to include confidential message recovery as shown in Figures 10-13. Turning first to Figures 10 and 11, ECDSR signature generation with confidential message recovery is shown.

[0063] For signature generation, entity A uses its private key d_A, entity B's public key G_B, and, as above, signs the message 10, having plaintext V and portions N_1 and N_2, which will be encrypted. Entity A generates an ephemeral key pair (k, Q) and then using k and the public key G_B, constructs a value $Q_B = kG_B$. The value Q_B is used to encrypt the portion N_1 so that only entity B (or an entity having access to B's private key if applicable) can recover or unlock the confidential information contained in the portion N_1.

[0064] Two encryption keys are then computed: $k_1 = KDF(Q_B)$ and $k_2 = KDF(Q)$. Using the two encryption keys, the recoverable and confidential portions are then encrypted using a suitable encryption scheme to generate a pair of corresponding signature components: $c_1 = ENC_{k_1}(N_1)$ and $c_2 = ENC_{k_2}(N_2)$. The encryption scheme E takes as input plaintext and a secret key and produces ciphertext, which create the signature components.

[0065] An intermediate value h is then computed by hashing a combination (e.g. concatenation) of the pair of signature components c_1 and c_2 and the visible portion V where: $h = Hash(c_1||c_2||V)$ and Hash is a suitable hash function that may also use additional information such as identification information of A, to create h. The value h is then converted into an integer e, and components c_1 and c_2 are converted to integers C_1 and C_2 respectively, to be used in computed another signature component s.

[0066] The signature component s in ECDSR with confidential message recovery uses the ECDSA signing equation with a combination of the integer representations C_1 and C_2 of the signature components c_1 and c_2 in place of the integer C as used above. In the embodiment of Figures 10 and 11, the signature component s is computed as: $s = k^{-1}(e + (C_1C_2)\cdot d_A) \bmod n$. The resultant signature $(s, c_1||c_2||V)$ may then be output. It may be noted that if there is concern with increasing the size due to, e.g. adding N_2, a compression algorithm could be applied to remove redundancy in a portion of the message.

[0067] Turning now to Figure 12, ECDSR with partial message recovery is shown, which enables any entity to recover the recoverable portion N_2 when having access to the public G_A of the signing entity A. As above, the specified characteristic, such as a certain amount of redundancy in the recovered value can be checked to verify the signature. For the purpose of this illustration, it will be assumed that the verifying entity in Figure 12 is entity Z, which cannot recover N_1 and thus N_1 remains confidential. This is because the value N1 is encrypted using a value derived from the public key G_B of entity B and thus can only be decrypted by having the private key d_B of entity B.

[0068] As can be seen from Figure 12, the entity Z obtains the signature $(s, c_1||c_2||V)$ and requires the public key G_A of the signing entity A to verify the signature. The intermediate value h is first computed using the same hash function Hash and the combination $c_1||c_2||V$ and any additional input information. The value h is then converted into an integer e. Since the ECDSA signing equation was used in Figure 11, a pair of components u_1 and u_2 are then computed as: $u_1 = es^{-1}$ and $u_2 = c_1c_2s^{-1}$ (i.e. using c_1c_2 in place of c); and such components are then used to compute a value Q' using the public key G_A as: $Q' = u_1G + u_2G_A$.

[0069] Having computed Q', entity Z then uses the same key derivation function KDF to obtain a decryption key $k_2' = KDF(Q')$. The decryption key k_2' and the signature component

c_2 are then used, with the complementary decryption function DEC, to recover N_2' from c_2. Having recovered N_2', entity Z then checks for the characteristic, e.g. a certain amount of redundancy and accepts or rejects the signature on this basis. As such, in this example, if entity Z does not find the proper amount of redundancy, the signature is deemed to be invalid.

[0070] Turning now to Figure 13, a process is shown for entity B to both verify the signature and recover the confidential portion. As can be seen in Figure 13, entity B obtains the signature $(s, c_1||c_2||V)$ and requires the public key G_A of the signing entity A and its own private key d_B, to verify the signature. As above, the intermediate value h is first computed using the same hash function Hash and the combination $c_1||c_2||V$ and any additional input information. The value h is then converted into an integer e and a representation Q' of the ephemeral key Q is then computed by first generating the values u_1 and u_2 and using the public key G_A as shown in Figure 12 and discussed in greater detail above.

[0071] In order to recover the confidential portion N_1, entity B also computes a representation Q_B' using the value Q' and its private key d_B, namely as: $Q_B' = d_B Q'$. Having computed Q_B' and Q', entity B then computes decryption keys k_1' and k_2' respectively, using the same key derivation function KDF, namely as: $k_1' = KDF(Q_B')$ and $k_2' = KDF(Q')$.

[0072] The decryption keys k_1' and k_2' and the signature components c_1 and c_2 are then used, with the complementary decryption function DEC, to recover N_1' and N_2' from c_1 and c_2 respectively. Having recovered N_1' and N_2', entity B then checks for the proper amount of redundancy in N_2', and accepts or rejects both N_1' and N_2' on this basis, since if the redundancy in N_2' is incorrect, the signature is invalid or has been compromised in some way.

[0073] It can therefore be seen that an ECDSR signature as discussed above can be used to provide both message recovery for verification, and confidential message recovery by enabling one to check a predictable output to verify the signature and using the public key of entity B to encrypt the confidential portion, which enables one to limit who can recover the confidential portion to a specific entity, in this example, entity B.

[0074] It will be appreciated that although the above examples are implemented in elliptic curve fields, the same principles may be applied to schemes in other fields such as Z_p.

[0075] For example, a discrete log implementation using El Gamal can be utilized. In such an implementation, the inputs are entity A's private key d_A; a public key (G_A, g, p), where $G_A = g^{d_A} \pmod{p}$, g being the generator and p being the group order; a message *m*; and entity B's public key G_B. The signature generation proceeds as follows:

[0076] (a) Generate ephemeral public key pair *k*, $r = g^k \pmod{p}$.

[0077] (b) Derive encryption key: *key* = KDF ($G^k_B \pmod{p}$).

[0078] (c) Encrypt message: $c = ENC_{key}(m)$.

[0079] (d) Compute $s = (Hash(c) - d_A r)k^{-1}) \pmod{p-1}$.

[0080] (e) Output c, (r,s).

[0081] The analogous signature verification algorithm, with a decryption routine for entity B outlined in steps (c) and (d) is as follows.

[0082] (a) Verify $0 < r < p$ and $0 < s < p-1$ or return INVALID.

[0083] (b) Verify $g^{Hash(c)} \equiv G^r_A r^s \pmod{p}$, or return INVALID.

[0084] (c) If B is performing verification, generate *key* = KDF($r^d_B \pmod{p}$), and decrypt $m=DEC_{key}(c)$.

[0085] (d) Return VALID and B obtains the message *m*.

[0086] The above El Gamal implementation can be of particular use in the application of Certificate Authority issuing a secret in a certificate, which the rightful owner could use for actions such as key-updating, certificate revocation, account management, or other key sharing applications.

[0087] It can be seen that the principles described above for providing message recovery (including confidential message recovery) can be extended to non-ECC implementations.

[0088] When implementing the above embodiments, there are several other extensions and/or variations that can be employed. One extension is that the ECDSR scheme can be

performed when N is an empty string. In this case, more of the existing ECDSA signing operations can be used.

[0089] Another extension is that the ECDSR scheme discussed above can be applied to a discrete log implementation using a modified DSA signature scheme.

[0090] It has also been recognized that the segmentation of the recoverable message can be extended to multiple directed messages. For example, if there are t receivers B_i with keys (d_{B_i}, G_{B_i}), then entity A can send a signature on the message $N_1||N_2||...||N_t||N_{t+1}||V$ where, as above, N_{t+1} carries the necessary redundancy and $c_i = ENC_{k_i}(N_i)$ where $k_i = KDF(k \cdot G_{B_i})$.

[0091] A Shamir sharing scheme can also be used to generate a t-threshold secret S, where t or more recipients with their respective portions are required to come together in order to reconstruct the secret S. In this implementation, each B_i is assigned a portion d_i for the secret S, which can be used as a private key. The signer then encrypts the message N_1 with the key $k_1 = KDF(k \cdot (SG))$, where S is the shared secret in a Shamir sharing scheme. When verifying the signature, the t recipients combine their portions d_i to create S for computing the decryption key k_1'.

[0092] Yet another extension allows a signed message to be verified by anyone, but requires all participants to be present, using a trusted system, to decrypt the message N_1. This can be done by having the signing entity A create a key $k_1 = KDF(k \cdot \sum_{i=1}^{2} G_{B_i})$, and encrypting the message by creating component c_1 as: $c_1 = ENC_{k_1}(N_1)$. Verification of the signature would proceed as discussed above, with $Q_B' = (\sum_{i=1}^{t} d_{B_i})(sG - G_A)$.

[0093] Yet another extension has the key agreement scheme for the private entity (e.g. entity B) be based on another scheme, such as using one-pass MQV. In this extension, the signing entity A generates a shared secret $z = MQV(d_A, k, G_A, G_B)$, and then uses z in a key derivation function $k_1 = KDF(z)$, along with additional information. On the verification side, there would be no change as to what occurs for verifiers other than entity B. Entity B however computes the analogous shared secret $z = MQV(d_B, d_B, G_A, Q')$, where Q′ is the same as that computed in Figure 6.

[0094] Additionally, the type of Diffie-Hellman (DH) key exchange in Figure 11 can be done with regular ECDSA signatures. In this case, the ephemeral signing key is used with the public key of the recipient. In this way, a confidentiality key can be created as: *key* = $KDF(kG_B)$, and the message encrypted as c = ENC(N, *key*). A traditional ECDSA signature is then computed on c, giving an (s,r) signature component pair. Anyone with entity A's public key G_A, can verify the signature (s,r) on c but only B can decrypt it.

[0095] A combination of any of the schemes discussed above, with any other one-pass exchange like one-pass MQV could also be implemented.

[0096] It can therefore be seen that a portion of the signed message can be kept confidential in an elliptic curve signature by dividing the message being signed into at least three parts, wherein one portion is visible or plaintext, another portion is hidden and recoverable by any entity having access to the signer's public key and carries the necessary redundancy for verification, and at least one additional portion is also hidden but only recoverable by a specific verifying entity having the necessary secret value (i.e. a confidential message recovery). The additional portion is kept confidential by encrypting such portion using a key generated from information specific to that verifying entity. In this way, any entity with access to the signer's public key can verify the signature by checking the redundancy of the one recovered portion, but cannot recover the confidential portion, only the specific entity can do so. It will be appreciated that the embodiment of Figures 3 to 6 can also be implemented using a plurality of portions, e.g. N and V only, wherein the hidden portion N is computed as N_1 in the above and is also used to verify the signature. As such, in general, the message is divided into a plurality of portions.

[0097] It can also be seen that message recovery in an elliptic curve signature can be implemented using a modification of the well analyzed ECDSA signing equation instead of, e.g. the Schnorr equation sometimes used in PV signature schemes.

[0098] 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.

Claims:

1. A method for generating a signature on a message, said method comprising:

dividing a message into a plurality of portions, a first portion of said message to be visible, a second portion of said message to be hidden and confidential such that only a specified entity can recover said second portion of the message;

encrypting said second portion of said message using a first encryption key to generate a first signature component, said first encryption key being generated using information specific to said specified entity;

generating a second signature component using said first signature component, said first portion of said message and a private key; and

preparing the signature comprising said first and second signature components and said first portion of said message.

2. The method according to claim 1, wherein said information specific to said specified entity is a public key.

3. The method according to claim 1, wherein said message comprises three or more portions including a plurality of portions to be hidden and confidential such that respective ones of a plurality of specified entities can recover corresponding ones of said plurality of portions to be hidden, each of said plurality of portions to be hidden being encrypted using respective encryption keys, each of said respective encryption keys being generated using information specific to a respective one of said plurality of specified entities.

4. The method according to claim 1, wherein said first encryption key is generated using information specific to a plurality of specified entities such that the presence of said plurality of specified entities is required to recover said second portion during verification of said signature.

5. The method according to according to claim 1, wherein said message comprises a third portion to be hidden and recoverable by any entity, said method further comprising encrypting said third portion of said message using a second encryption key to generate a

third signature component, said second signature component being generated using said third signature component and said signature including said third signature component.

6. The method according to claim 5 wherein said second encryption key is generated using public information available to said any entity such that said third portion can be recovered using a public key of a signer.

7. The method according to claim 1 wherein said signature is an elliptic curve signature and said second signature component is generated using either an ECPVS or an ECDSA signature scheme.

8. The method according to claim 1 wherein said signature is a discrete log signature.

9. A cryptographic processor configured for executing computer readable instructions for performing the method according to claim 1.

10. A computer readable medium comprising computer executable instructions for causing a cryptographic processor to perform the method according to claim 1.

11. A method of verifying a signature on a message is provided, said message comprising a plurality of portions, a first portion of said message being visible, a second portion of said message hidden and confidential such that only a specified entity can recover said second portion of said message, said method comprising:

obtaining said signature having a first signature component encrypting said second portion of said message using a first encryption key, said first encryption key having been generated using information specific to said specified entity, having a second signature component generated using said first signature component, said first portion of said message and a private key, and having said first portion of said message;

if said specified entity, generating a first decryption key using said information specific to said specified entity, a private key of said specified entity, said second signature component and a value derived from the combination of said first signature component and said first portion of said message; and

using said first decryption key to decrypt said second portion of said message from said first signature component.

12. The method according to claim 11, wherein said information specific to said specified entity is a public key, said method comprising obtaining said public key.

13. The method according to claim 11, wherein said message comprises three or more portions including a plurality of portions being hidden and confidential such that respective ones of a plurality of specified entities can recover corresponding ones of said plurality of portions being hidden and confidential, each of said plurality of portion hidden and confidential having been encrypted using respective encryption keys, each of said respective encryption keys being generated using information specific to a respective one of said plurality of specified entities, said method further comprising recovering a plurality of said portions being hidden and confidential.

14. The method according to claim 11, wherein said first encryption key has been generated for signing using information specific to a plurality of specified entities, said method further comprising obtaining and combining said information specific to said plurality of specified entities to generate said first encryption key for verification.

15. The method according to claim 11, wherein said message comprises a third portion intended to be hidden and recoverable by any entity; said signature having a third component encrypting said third portion of said message using a second encryption key, said method comprising: computing a second decryption key using said second signature component, a public key of a signing entity and a value derived from said first signature component and said first portion of said message; using said second decryption key to decrypt said third portion of said message from said third signature component; and verifying said signature using said third portion once recovered.

16. The method according to claim 15 further comprising checking said third portion for a predetermined characteristic.

17. The method according to claim 16 wherein said predetermined characteristic is redundancy in said third portion.

18. The method according to claim 15 wherein said second encryption key is generated using public information available to said any entity such that said third portion is recovered using a public key of a signer.

19. The method according to claim 11 wherein said signature is an elliptic curve signature and said second signature component is generated using either an ECPVS or an ECDSA signature scheme.

20. The method according to claim 11 wherein said signature is a discrete log signature.

21. A cryptographic processor configured for executing computer readable instructions for performing the method according to claim 11.

22. A computer readable medium comprising computer executable instructions for causing a cryptographic processor to perform the method according to claim 11.

23. A method of generating a signature on a message is provided, said method comprising:

dividing a message into a plurality of portions, a first portion of the message being visible and a second portion of said message to be hidden and recoverable by any entity;

encrypting said second portion of said message using a first encryption key to generate a first signature component;

generating a second signature component using said first signature component, said first portion of the message and an element derived from a private key as inputs to an elliptic curve digital signature algorithm (ECDSA) signing equation; and

preparing said signature comprising said first and second signature components and said first portion of said message.

24. The method according to claim 23 wherein said second portion of said message is an empty string.

25. A cryptographic processor configured for executing computer readable instructions for performing the method according to claim 23.

26. A computer readable medium comprising computer executable instructions for causing a cryptographic processor to perform the method according to claim 23.

27. A method of verifying a signature on a message is provided, said message comprising a plurality of portions, a first portion of said message being visible and a second portion of said message being hidden and recoverable by any entity; said method comprising:

obtaining said signature having a first signature component encrypting said second portion of said message using a first encryption key, having a second component generated using said first signature component, said first portion of said message and an element derived from a private key as inputs to an ECDSA signing equation, and having said first portion of said message;

computing a first decryption key using said first and second signature components, a public key of a signing entity and a value derived from a combination of said first signature component and said first portion of said message;

using said first decryption key to decrypt said second portion of said message from said second signature component; and

using the decrypted second portion of said message to verify said signature.

28. The method according to claim 27 wherein said second portion of said message is an empty string.

29. A cryptographic processor configured for executing computer readable instructions for performing the method according to claim 27.

30. A computer readable medium comprising computer executable instructions for causing a cryptographic processor to perform the method according to claim 27.

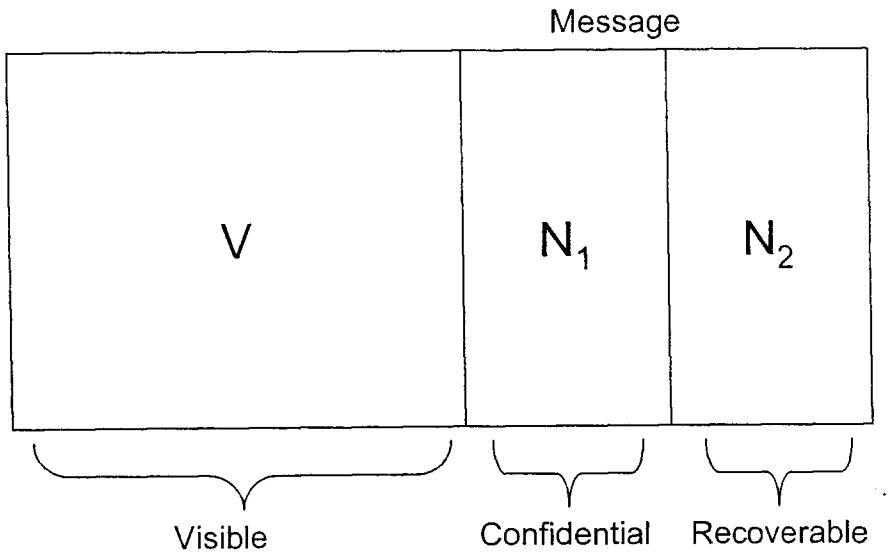

Figure 1

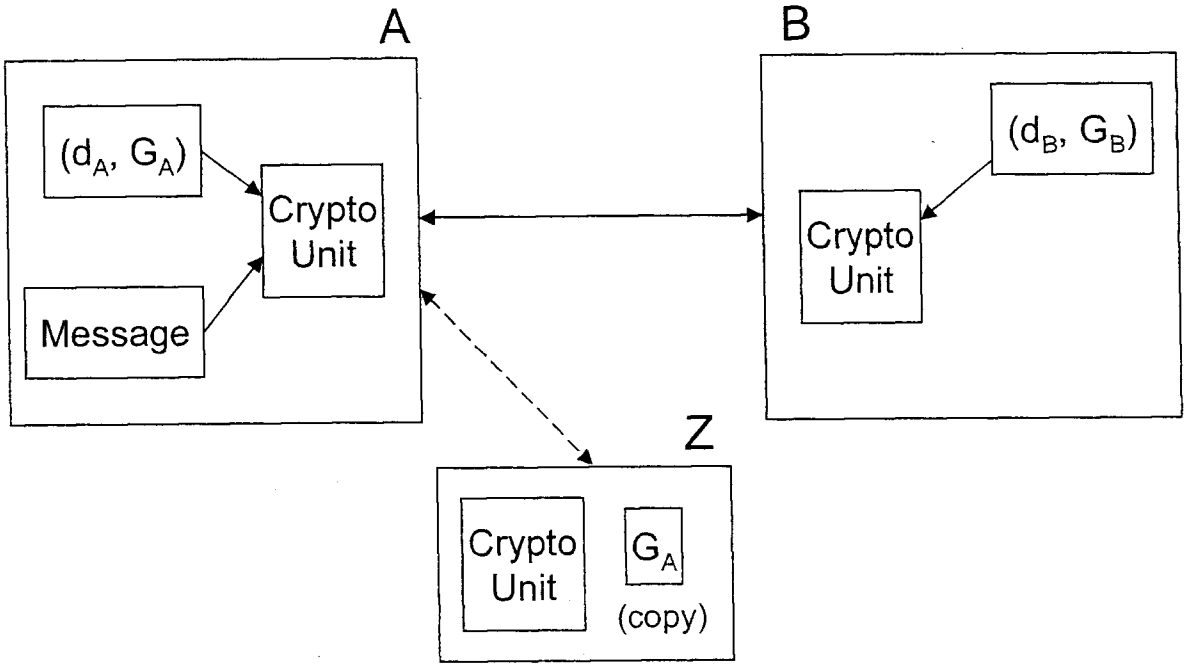

Figure 2

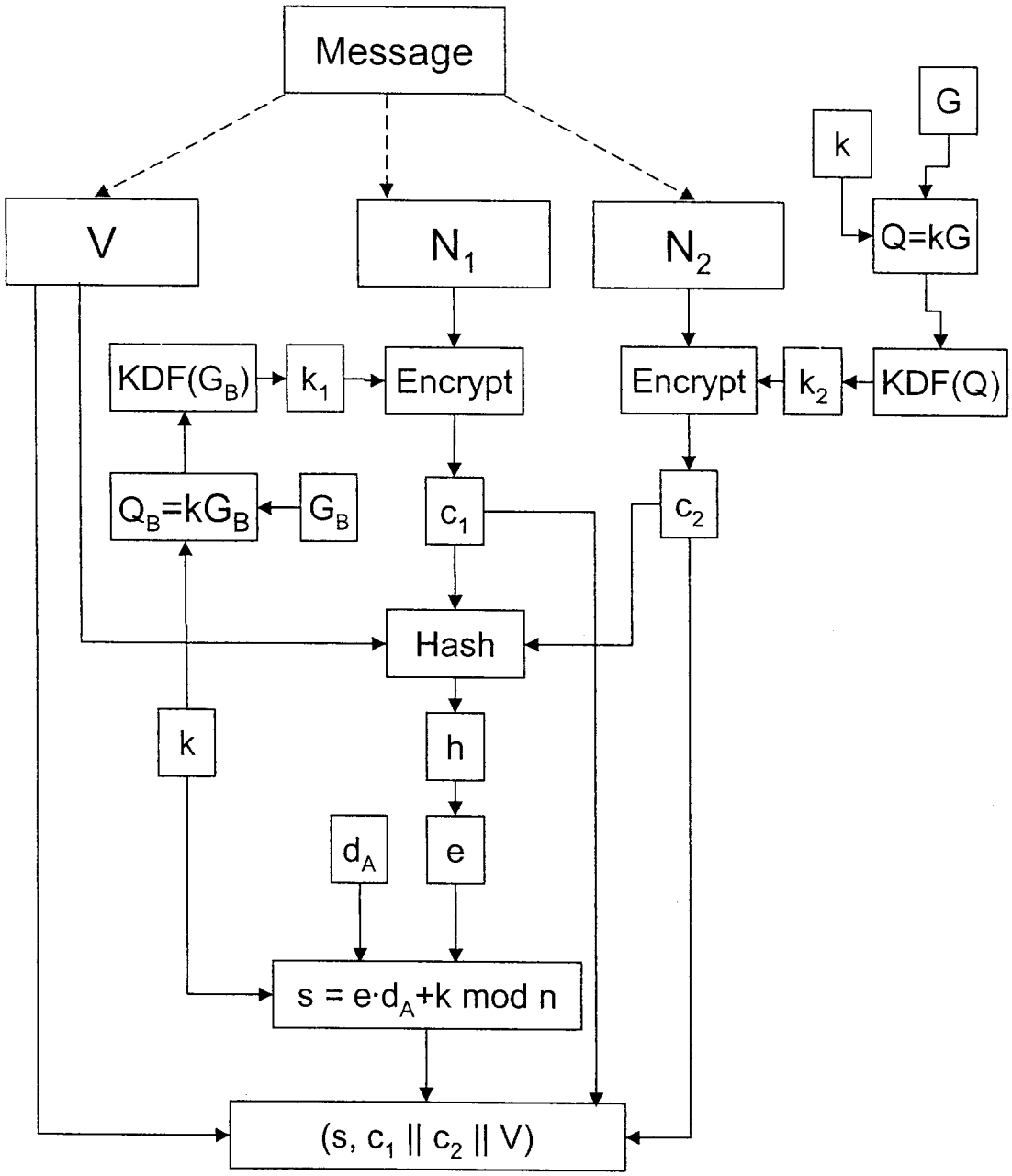
Message
V
N1
N2
G
k
Q=kG
KDF(QB)
k1
Encrypt
Encrypt
k2
KDF(Q)
QB=kGB
GB
c1
c2
Hash
k
h
dA
e
s = e·dA+k mod n
(s, c1 || c2 || V)

Figure 3

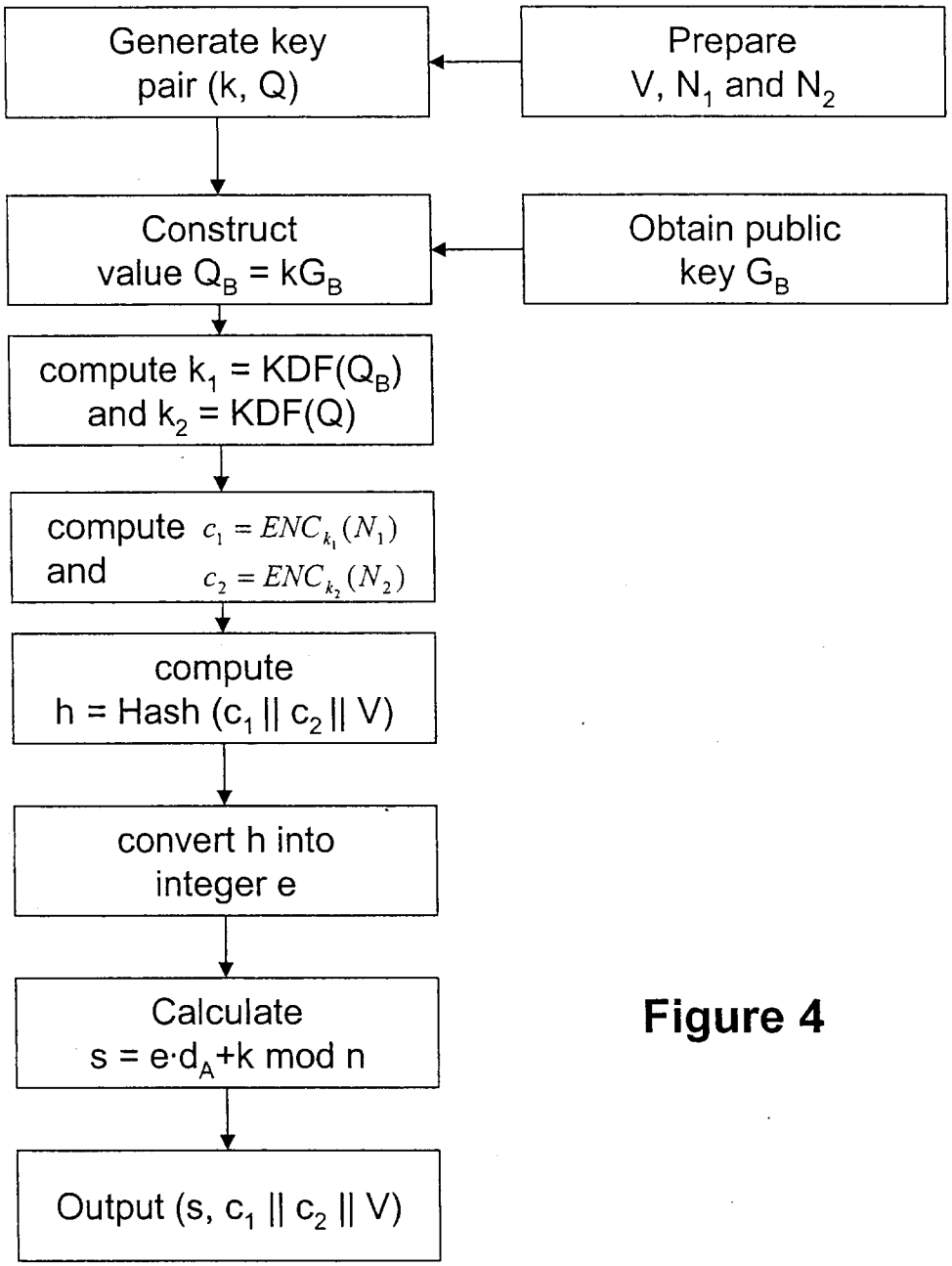
Generate key pair (k, Q)
Prepare V, N_1 and N_2
Construct value $Q_B = kG_B$
Obtain public key G_B
compute $k_1 = KDF(Q_B)$ and $k_2 = KDF(Q)$
compute $c_1 = ENC_{k_1}(N_1)$ and $c_2 = ENC_{k_2}(N_2)$
compute $h = Hash(c_1 || c_2 || V)$
convert h into integer e
Calculate $s = e \cdot d_A + k \bmod n$
Output $(s, c_1 || c_2 || V)$

Figure 4

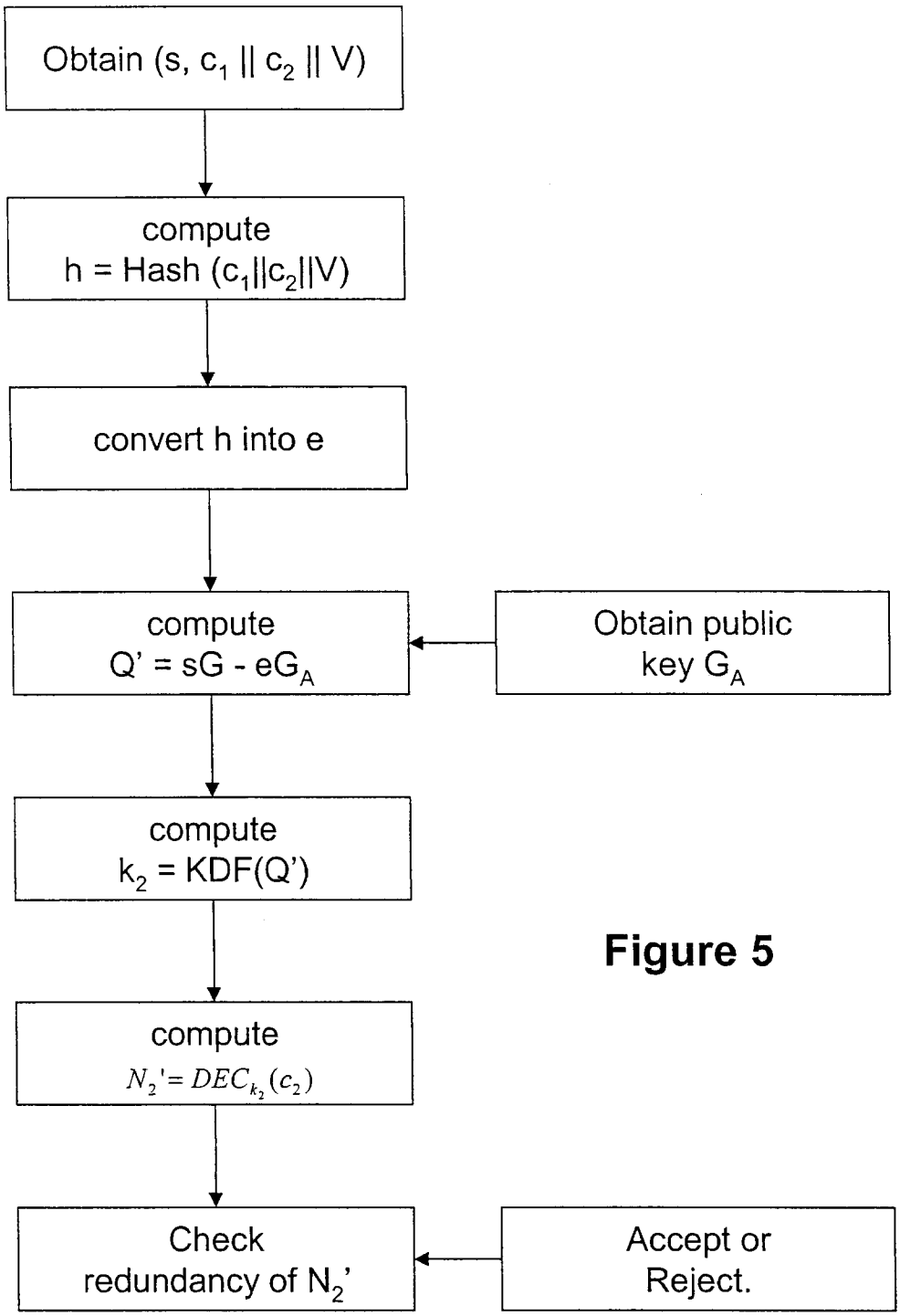
Obtain (s, c_1 || c_2 || V)
compute
h = Hash (c_1||c_2||V)
convert h into e
compute
Q' = sG - eG_A
Obtain public
key G_A
compute
k_2 = KDF(Q')
compute
$N_2' = DEC_{k_2}(c_2)$
Check
redundancy of N_2'
Accept or
Reject.

Figure 5

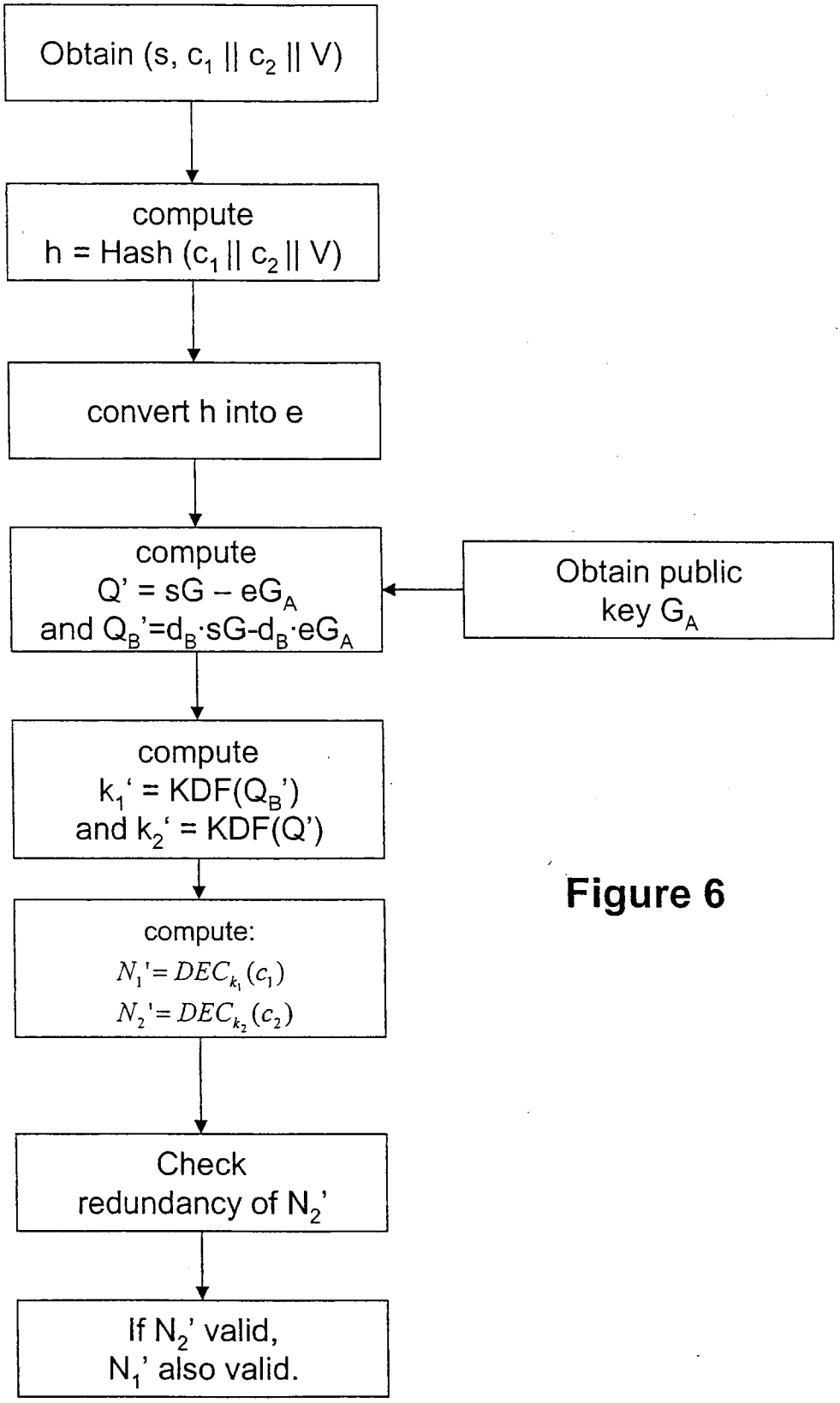
Obtain (s, c_1 || c_2 || V)
compute
h = Hash (c_1 || c_2 || V)
convert h into e
compute
Q' = sG – eG_A
and Q_B'=d_B·sG-d_B·eG_A
Obtain public
key G_A
compute
k_1' = KDF(Q_B')
and k_2' = KDF(Q')
compute:
$N_1' = DEC_{k_1}(c_1)$
$N_2' = DEC_{k_2}(c_2)$
Check
redundancy of N_2'
If N_2' valid,
N_1' also valid.

Figure 6

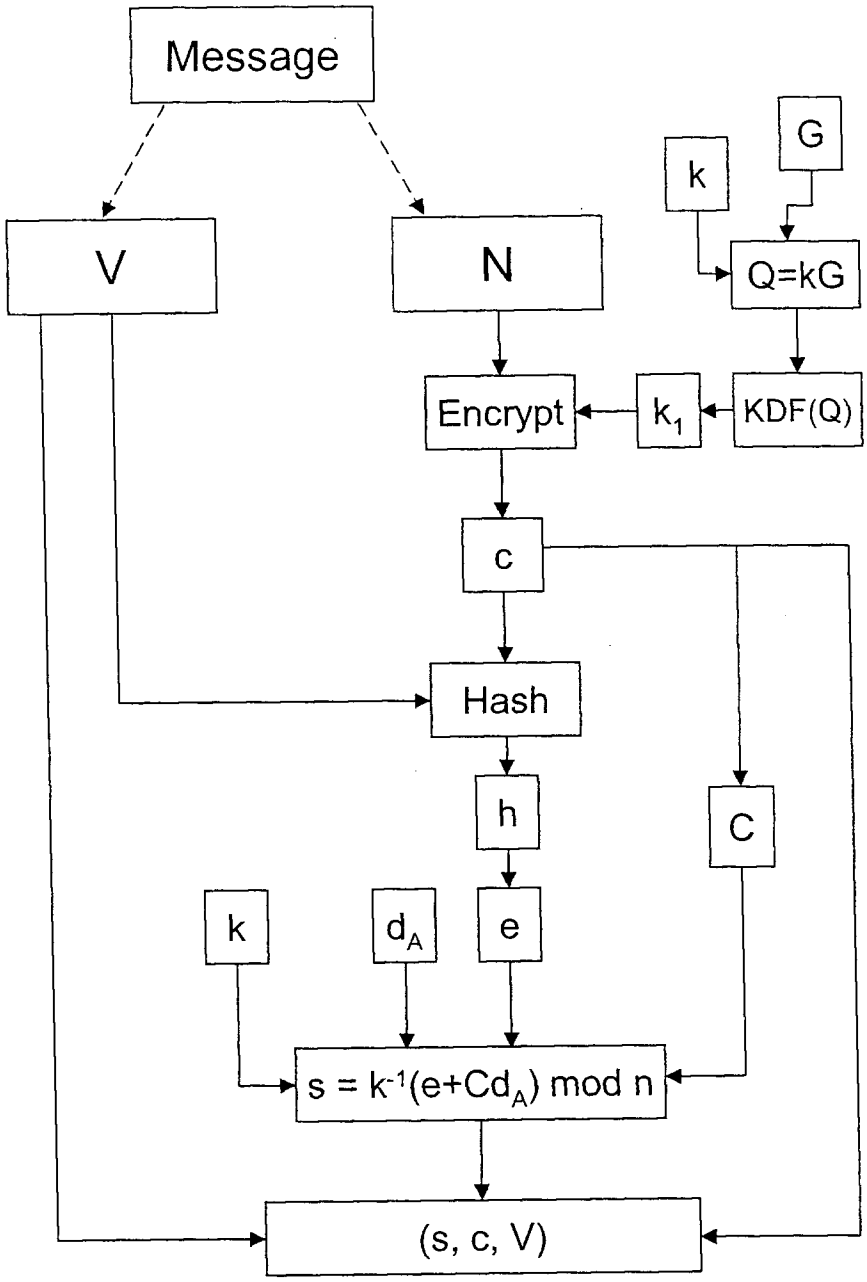
Message
V
N
k
G
Q=kG
Encrypt
k_1
KDF(Q)
c
Hash
h
C
k
d_A
e
$s = k^{-1}(e+Cd_A) \bmod n$
(s, c, V)

Figure 7

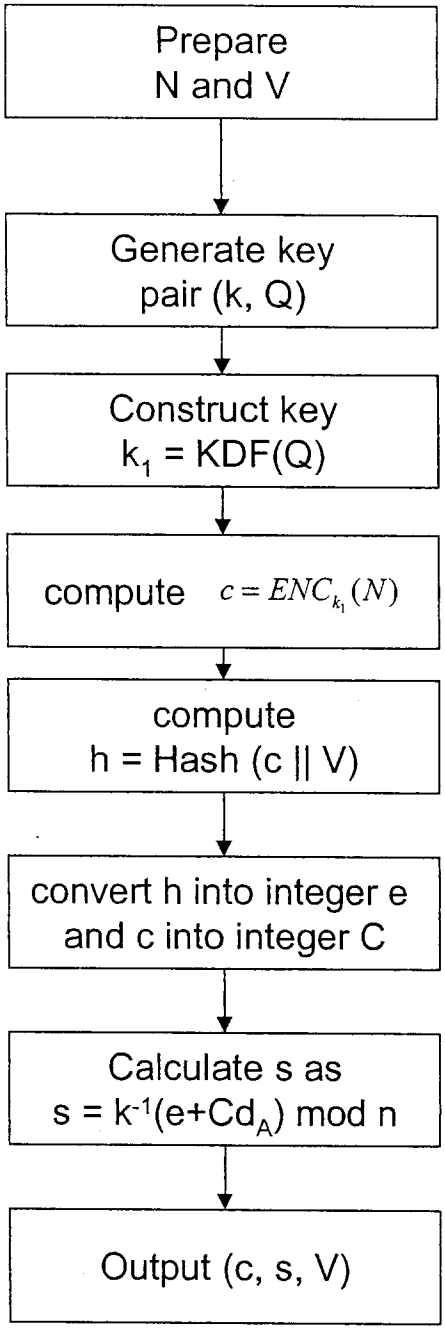
Prepare
N and V
Generate key
pair (k, Q)
Construct key
k_1 = KDF(Q)
compute $c = ENC_{k_1}(N)$
compute
h = Hash (c || V)
convert h into integer e
and c into integer C
Calculate s as
$s = k^{-1}(e+Cd_A) \bmod n$
Output (c, s, V)

Figure 8

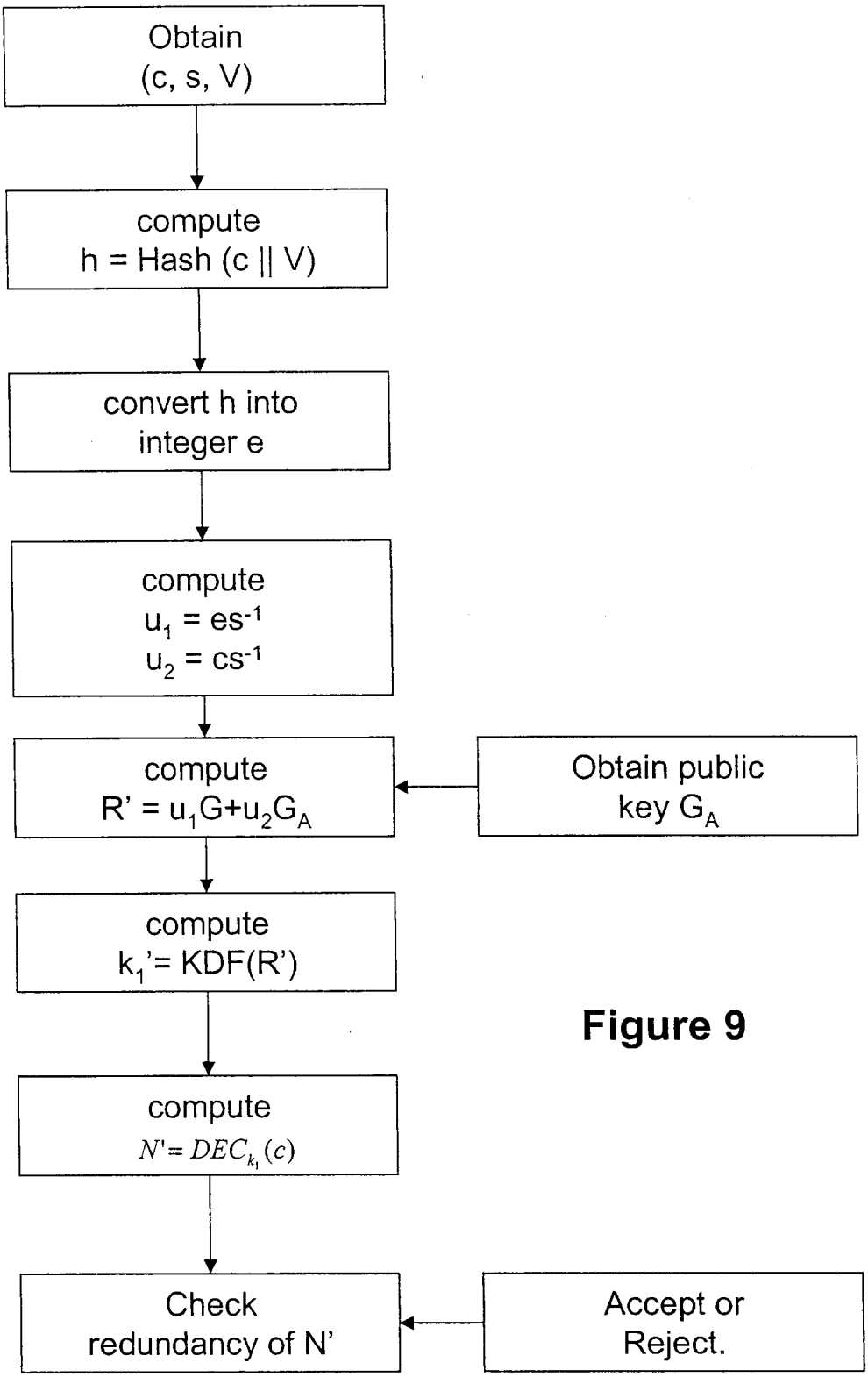
Obtain
(c, s, V)
compute
h = Hash (c || V)
convert h into
integer e
compute
$u_1 = es^{-1}$
$u_2 = cs^{-1}$
compute
$R' = u_1G+u_2G_A$
Obtain public
key G_A
compute
$k_1' = KDF(R')$
compute
$N' = DEC_{k_1}(c)$
Check
redundancy of N'
Accept or
Reject.

Figure 9

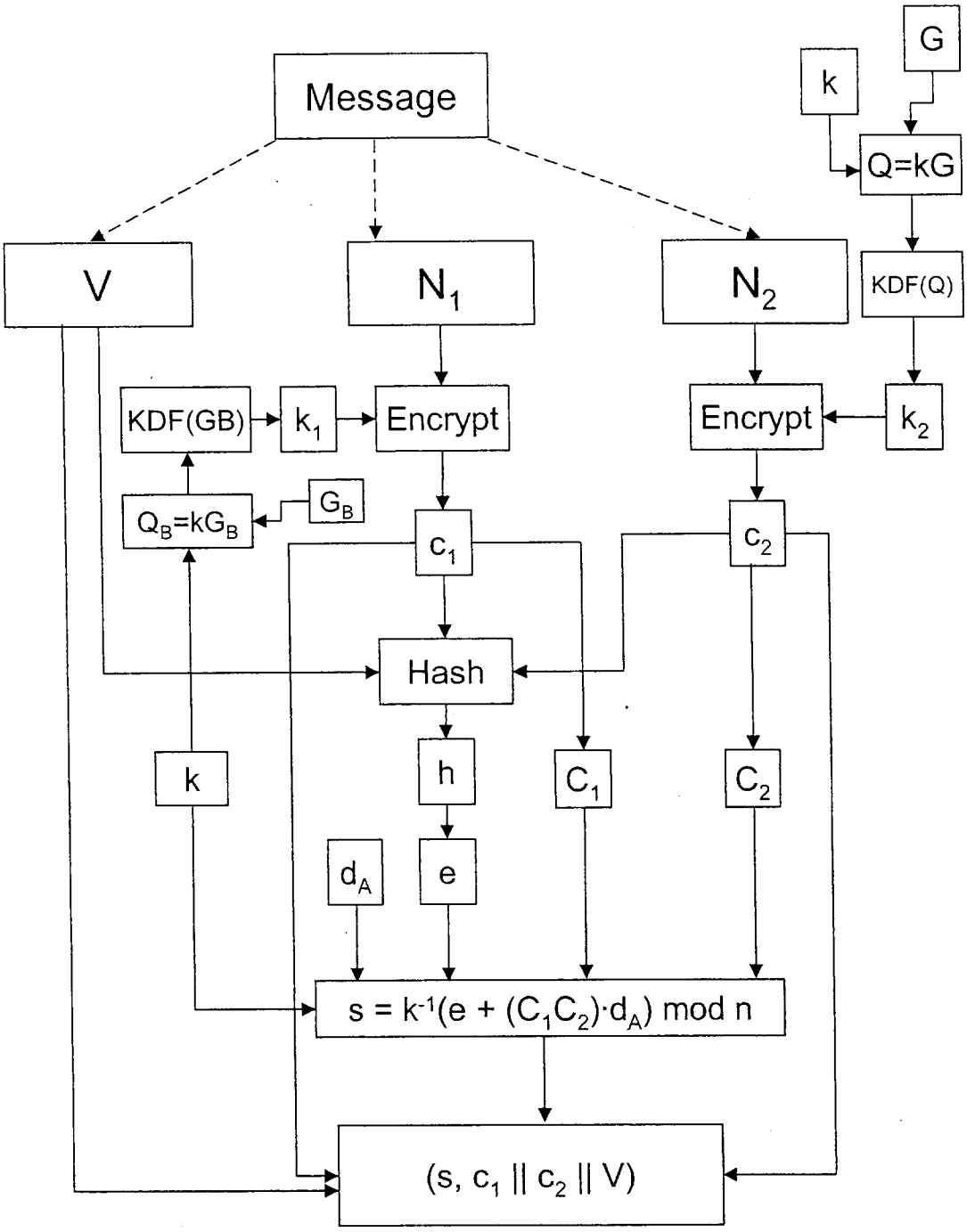
Message
G
k
Q=kG
V
N1
N2
KDF(Q)
KDF(GB)
k1
Encrypt
Encrypt
k2
QB=kGB
GB
c1
c2
Hash
k
h
C1
C2
dA
e
s = k-1(e + (C1C2)·dA) mod n
(s, c1 || c2 || V)

Figure 10

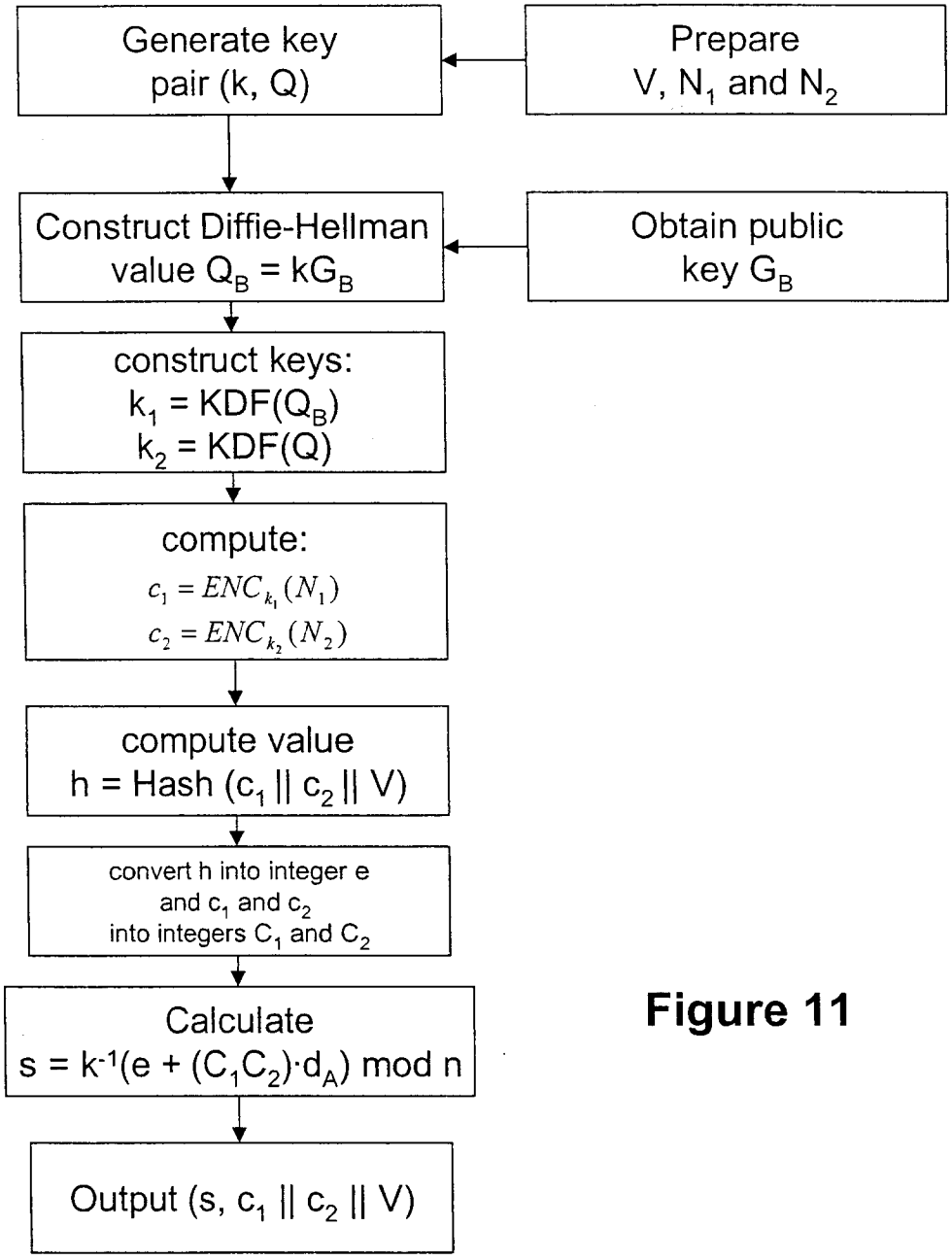
Generate key pair (k, Q)
Prepare V, N_1 and N_2
Construct Diffie-Hellman value $Q_B = kG_B$
Obtain public key G_B
construct keys: $k_1 = KDF(Q_B)$ $k_2 = KDF(Q)$
compute: $c_1 = ENC_{k_1}(N_1)$ $c_2 = ENC_{k_2}(N_2)$
compute value h = Hash (c_1 || c_2 || V)
convert h into integer e and c_1 and c_2 into integers C_1 and C_2
Calculate $s = k^{-1}(e + (C_1C_2)\cdot d_A) \bmod n$
Output (s, c_1 || c_2 || V)

Figure 11

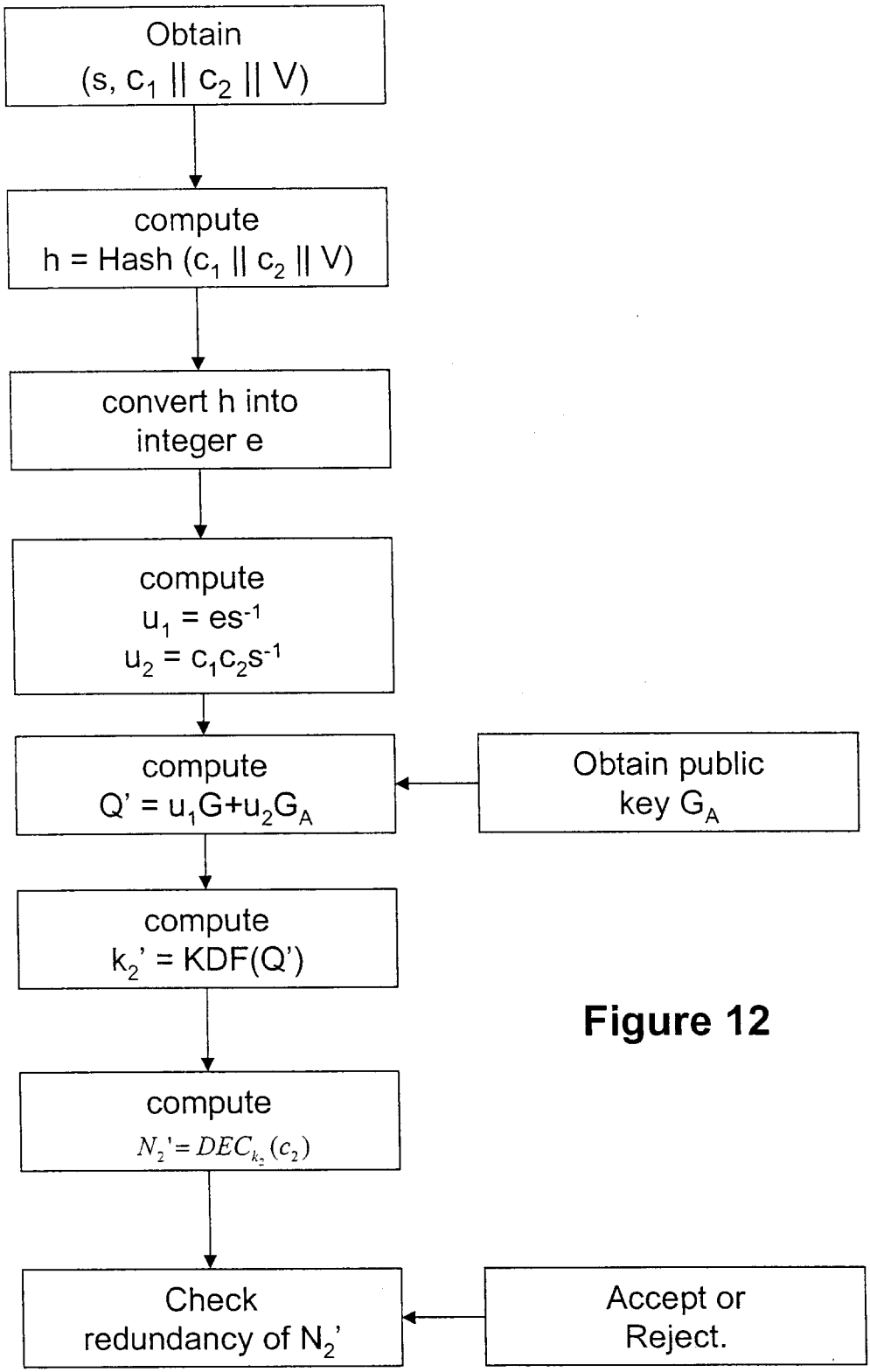
Obtain
(s, c_1 || c_2 || V)
compute
h = Hash (c_1 || c_2 || V)
convert h into
integer e
compute
$u_1 = es^{-1}$
$u_2 = c_1c_2s^{-1}$
compute
Q' = $u_1G+u_2G_A$
Obtain public
key G_A
compute
k_2' = KDF(Q')
compute
$N_2' = DEC_{k_2}(c_2)$
Check
redundancy of N_2'
Accept or
Reject.

Figure 12

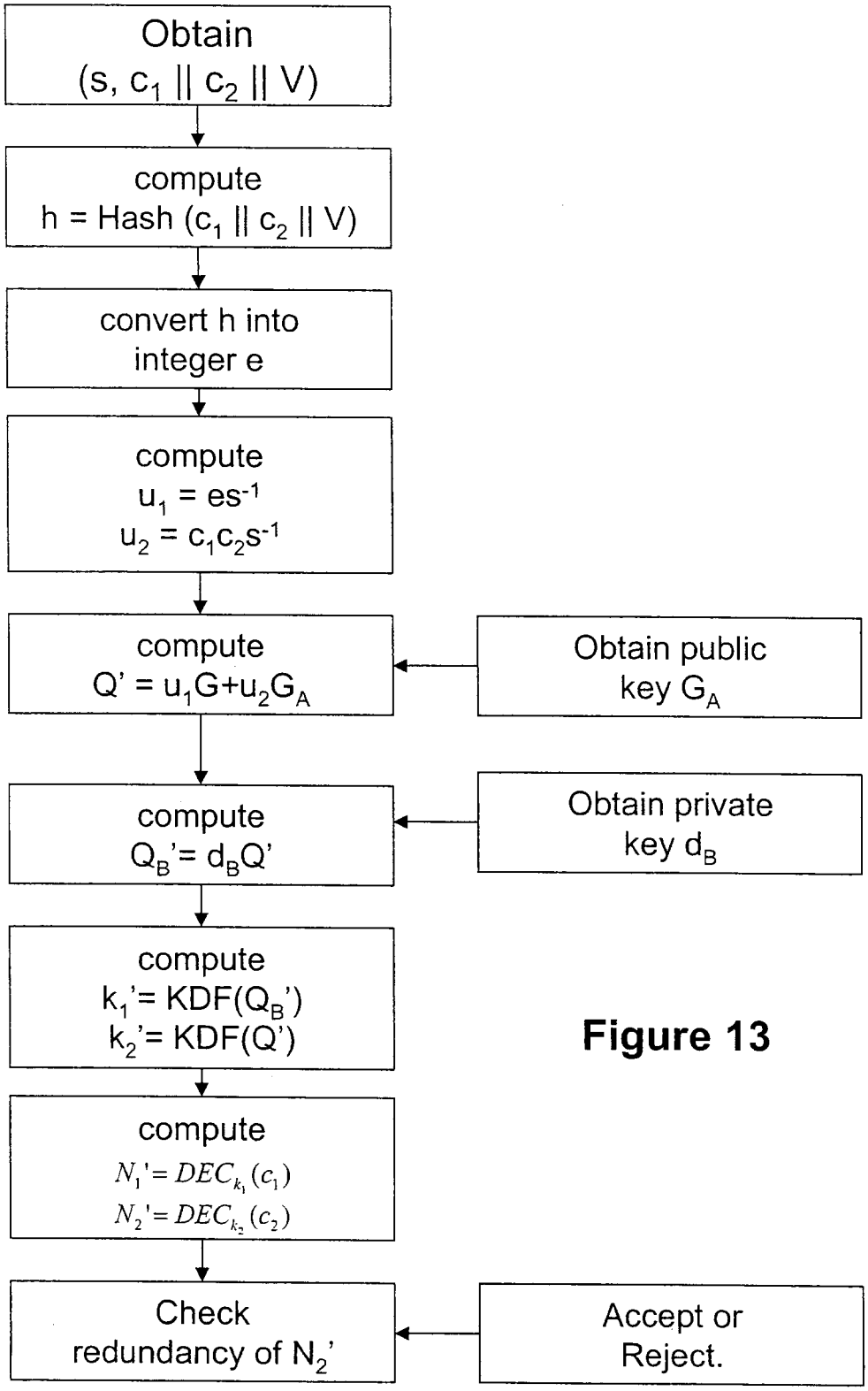
Obtain
(s, c_1 || c_2 || V)
compute
h = Hash (c_1 || c_2 || V)
convert h into
integer e
compute
$u_1 = es^{-1}$
$u_2 = c_1c_2s^{-1}$
compute
Q' = $u_1G+u_2G_A$
Obtain public
key G_A
compute
$Q_B' = d_BQ'$
Obtain private
key d_B
compute
$k_1' = KDF(Q_B')$
$k_2' = KDF(Q')$
compute
$N_1' = DEC_{k_1}(c_1)$
$N_2' = DEC_{k_2}(c_2)$
Check
redundancy of N_2'
Accept or
Reject.

Figure 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2008/001541

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***H04L 9/14*** (2006.01) , ***H04L 9/30*** (2006.01) , ***H04L 9/32*** (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: ***H04L 9/14*** (2006.01) , ***H04L 9/30*** (2006.01) , ***H04L 9/32*** (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)
QWEB, Canadian Patent Database
keywords: signature, message, encryption, ECDSA, private key, public key, first signature, second signature

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7,249,259 B1 Vanstone et al. 24 Jul. 2007 (24-07-2005) column 5, lines 10-30 column 3, lines 56-61 column 5, lines 10-30	1 to 30
A	WO 01/10078 A1 Coron et al. 8 Feb. 2001 (08-02-2001) ** See whole document **	1 to 30

[ ] Further documents are listed in the continuation of Box C.

[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
22 December 2008 (22-12-2008)	14 January 2009 (14-01-2009)

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	Francois Ziade 819- 994-7460

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2008/001541

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US7249259	24-07-2007	AT366007	15-07-2007
		AU777723	28-10-2004
		AU5655900	08-03-2001
		CA2317775	07-03-2001
		DE60035317	13-03-2008
		EP1083700	27-06-2007
		EP1830514	05-09-2007
		JP2001125482	11-05-2001
		US7249259	24-07-2007
		US2008141036	12-06-2008
WO0110078	08-02-2001	AU6575300	19-02-2001
		CN1377539	30-10-2002
		EP1205051	15-05-2002
		FR2797127	12-10-2001