(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 292 872 B9

(12) CORRECTED EUROPEAN PATENT SPECIFICATION

(15) Correction information:
Corrected version no 1 (W1 B1)
Corrections, see
Description Paragraph(s) 30, 56
Claims EN 10, 16, 33, 40, 42
Claims FR 40
Numerous spelling errors of minor importance

(48) Corrigendum issued on:
15.09.2010 Bulletin 2010/37

(45) Date of publication and mention of the grant of the patent:
19.08.2009 Bulletin 2009/34

(21) Application number: **01944801.8**

(22) Date of filing: **11.06.2001**

(51) Int Cl.:
G06F 21/00 *(2006.01)*

(86) International application number:
PCT/CA2001/000833

(87) International publication number:
WO 2001/095068 (13.12.2001 Gazette 2001/50)

(54) A METHOD FOR THE APPLICATION OF IMPLICIT SIGNATURE SCHEMES

VERFAHREN ZUR ANWENDUNG VON IMPLIZITEN UNTERSCHRIFTEN

PROCEDE D'APPLICATION DE SYSTEMES DE SIGNATURE IMPLICITE

(84) Designated Contracting States:
DE FR GB

(30) Priority: **09.06.2000 US 589891**

(43) Date of publication of application:
19.03.2003 Bulletin 2003/12

(60) Divisional application:
09010612.1 / 2 148 465

(73) Proprietor: **Certicom Corp.**
Mississauga, ON L4W 5L1 (CA)

(72) Inventor: **VANSTONE, Scott, A.**
Campbellville, Ontario L0P 1B0 (CA)

(74) Representative: **Rickard, David John**
Ipulse
26 Mallinson Road
London
SW11 1BP (GB)

(56) References cited:
WO-A-99/49612

- **RIVEST R L: "CAN WE ELIMINATE CERTIFICATE REVOCATION LISTS?" FINANCIAL CRYPTOGRAPHY. INTERNATIONAL CONFERENCE, XX, XX, February 1998 (1998-02), pages 178-183, XP000997964**
- **YUNG-KAO HSU; SEYMOUR S: "Intranet security framework based on short-lived certificates" PROCEEDINGS SIXTH IEEE WORKSHOPS ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, IEEE COMPUT. SOC, 20 June 1997 (1997-06-20), pages 228-233, XP002202960 Cambridge, MA, USA ISBN: 0-8186-7967-0**

EP 1 292 872 B9

Description

[0001] This invention relates generally to cryptographic schemes, and more specially to implicit signature schemes.

BACKGROUND OF THE INVENTION

[0002] Various schemes of generating a public key in a secure digital communication system having at least one trusted entity and subscriber entities can be found in PCT publication No. WO 99/49612 A to QU et al. In such schemes, for each subscriber the trusted entity selects a unique identity distinguishing the subscriber, generates a public key reconstruction of the subscriber by mathematically combining a generator of the trusted entity with a private value of the subscriber, such that the pair of unique identity and public key reconstruction serves as the subscriber's implicit certificate, combines the implicit certificate information in accordance with a mathematical function to derive an entity information, generates a private key of the subscriber by signing the entity information and transmitting the private key to the subscriber, whereby the subscriber's public key may be reconstructed from the public information, the public key reconstruction and the unique identity.

[0003] A paper entitled "Intranet Security Framework based on Short-Lived Certificates" by Hsu et al. (Proceedings Sixth IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Comput. SOC, 20 June 1997, pages 228-233) describes an intranet security framework based on public key cryptography. It uses short-lived certificates to avoid and eliminate costly and difficult key management issues in the typical X.509 authentication framework. Specifically, it loosens the tightly coupled relationship between the security components and the application. Operationally, the framework creates the tasks of entity registration and certification so that certificates can be efficiently and securely delivered to the client.

[0004] Diffie-Hellman key agreement provided the first practical solution to the key distribution problem, in cryptographic systems. The key agreement protocol allows two parties never having met in advance or sharing key material to establish a shared secret by exchanging messages over an open (unsecured) channel. The security rests on the intractability of computing discrete logarithms or in factoring large integers.

[0005] With the advent of the Internet and such like, the requirement for large-scale distribution of public keys and public key certificates is becoming increasingly important to enable systems like Diffie-Hellman key agreement.

[0006] A number of vehicles are known by which public keys may be stored, distributed or forwarded over unsecured media without danger of undetectable manipulation. These vehicles include public-key certificates, identity-based systems, and implicit certificates. The objective of each vehicle is to make one party's public key available to others such that its authenticity and validity are verifiable.

[0007] A public-key certificate is a data structure consisting of a data part and a signature part. The data part contains cleartext data including as a minimum, a public key and a string identifying the party to be associated therewith. The signature part consists of the digital signature of a certification authority (CA) over the data part, effectively the encryption of the data with the CA's private key so it may be recovered with his public key, thereby binding the entities identity to the specified public key. The CA is a trusted third party whose signature on the certificate vouches for the authenticity of the public key bound to the subject entity.

[0008] Identity-based systems (ID-based system) resemble ordinary public-key systems, involving a private transformation and a public transformation, but parties do not have explicit public keys as before. Instead, the public key is effectively replaced by a party's publicly available identity information (e.g. name or network address). Any publicly available information, which uniquely identifies the party and can be undeniably associated with the party, may serve as identity information. Here a trusted CA is required to furnish each party with the private key corresponding to their public key.

[0009] An alternate approach to distributing public keys involves implicitly certified public keys. Here explicit user public keys exist, but they are to be reconstructed by the recipient rather than transported by explicitly signed public-key certificates as in certificate based systems. Thus implicitly certified public keys may be used as an alternative means for distributing public keys (e.g. Diffie-Hellman keys).

[0010] With a conventional certificate, the authenticity of the information must be verified to ensure that the sender and the sender's public key are bound to one another. With an implicit certification it is simply necessary to verify the sender's signature of the message using the implicit certificate. The primary advantage of implicit certificates is the computationally expense explicit certificate verification is not required as it is in certification schemes. Further, unconditionally trusted CAs are not required as they are in ID-based schemes.

[0011] An example of an implicitly certified public key mechanism is known as Gunther's implicitly-certified public key method. In this method:

1. A trusted server T selects an appropriate fixed public prime p and generator α of Z^*_p. T selects a random integer t, with $1 \leq t \leq p-2$ and $\gcd(t,p-1) = 1$, as its private key, and publishes its public key $u = \alpha^t \bmod p$, along with α, p.
2. T assigns to each party A a unique name or identifying string I_A and a random integer k_A with $\gcd(k_A, p-1) = 1$. T then computes $P_A = \alpha^{kA} \bmod p$. P_A is A's key reconstruction public data, allowing other parties to compute $(P_A)^a$ below.

3. Using a suitable hash function h, T solves the following equation for a:

$$H(I_A) \equiv t.P_A + k_A\, a(\mathrm{mod}\ p\text{-}1)$$

4. T securely transmits to A the pair (r,s) = (P_A,a), which is T's ElGamal signature on I_A. (a is A's private key for a Diffie-Hellman key-agreement)

5. Any other party can then reconstruct A's Diffie-Hellman public key P_A^a entirely from publicly available information (α, I_A, u, P_A,p) by computing:

$$P_A^a \equiv \alpha^{H(I_A)} u^{-P_A} \bmod p$$

[0012] Thus signing an implicit certificate needs one exponentiation operation, but reconstructing the ID-based implicitly-verifiable public key needs two exponentiations.

[0013] It is known that exponentiation in the group Z_p^* and its analog scalar multiplication of a point in E (F_q) is computationally intensive. An RSA scheme is extremely slow requiring successive squaring and multiplication operations. Elliptic curve (EC) cryptosystems are not only more robust but also more efficient by using doubling and adding operations. However, despite the resounding efficiency of EC systems over RSA type systems the computational requirement is still a problem particularly for computing devices having limited computing power such as "smart cards", pagers and such like.

[0014] Significant improvements have been made in the efficacy of certification protocols by adopting the protocols set out in Canadian patent application 2,232,936. In this arrangement, an implicitly-certified public key is provided by cooperation between a certifying authority, CA, and a correspondent A.

[0015] For each correspondent A, the CA selects a unique identity I_A distinguishing the entityA. The CA generates public data γ_A for reconstruction of a public key of correspondent A by mathematically combining a private key of the trusted party CA and a generator created by the CA with a private value of the correspondent A. The values are combined in a mathematically secure way such that the pair (I_A,γ_A) serves as correspondent A's implicit certificate. The CA combines the implicit certificate information (I_A,γ_A) in accordance with a mathematical function F(γ_A,I_A) to derive an entity information *f*. A private key *a* of the correspondent A is generated from *f* and the private value of the correspondent A. The correspondent A's public key may be reconstructed from the public information, the generator γ_A and the identity I_A relatively efficiently.

[0016] Certificates, implicit certificates, and ID-based systems provide assurance of the authenticity of public keys. However, it is frequently necessary to verify the status of the public key to ensure it has not been revoked by the CA.

[0017] Several solutions are known to this revocation problem, the most common bein the use of certificate revocation lists (CRLs). Each CA maintains a CRL which contains the serial number of revoked certificates and is signed by the CA using its private key. When a recipient receives a message that has been secured with a certificate, the recipient will recover the serial number, and check the CRL.

[0018] Typically, therefore, the correspondent A will sign a message *m* with a private key, *a*, and forward it together with a certificate from the CA that binds the sender A and the public key *a*P. The recipient B checks the certificate and verifies the signature on the message *m*. The correspondent B will then ask the CA whether the certificate is valid and receives a message signed by the CA confirming the status of the certificate at a particular time. The correspondent B will then verify the signature on the CA's message and proceed accordingly to accept or reject the message sent by correspondent A.

[0019] During this process it is necessary for correspondent A to perform one signature, for the CA to perform one signature, and for the recipient B to verify three signatures. CAs may also issue authorization or attributable certificates in addition to public-key certificates. In this case the certificate issued by the CA to the correspondent A has a certain expiry or has details such as a credit limit or access rights to certain programs.

[0020] However with each arrangement, verification of the certificates is necessary as the information contained in the certificate may change periodically, even within the life of the certificate.

[0021] Furthermore, a correspondent may wish to be recertified. This is particularly true if the correspondent has reason to believe that its implicit public key has been compromised. However, recertification is a costly process that requires the correspondent to regenerate its private key, securely communicate its private key with the CA, and regenerate the data for constructing and reconstructing the implicit public key.

[0022] Accordingly, there is a need for a technique that simplifies the verification and recertification of certificates issued by a certifying authority and it is an object of the present invention to provide a technique that obviates or mitigates the above disadvantages.

SUMMARY OF THE INVENTION

[0023] In accordance with an embodiment of the present invention there is provided a method of verifying a transaction over a data communication system between a first and second correspondent through the use of a certifying authority. The method comprises the following steps. One of the first and second correspondents advising the certifying authority that the transaction is to be validated. The certifying authority determines whether to validate the transaction requested by the first correspondent. Upon agreeing to validate said transaction, the certifying authority generates implicit signature components including transaction specific information. At least one of the implicit signature components is forwarded to the first correspondent for permitting the first correspondent to generate an ephemeral private key. At least one of the implicit signature components is forwarded to the second correspondent for permitting recovery of an ephemeral public key corresponding to the ephemeral private key. The first correspondent signs a message with the ephemeral private key and forwards the message to the second correspondent. The second correspondent attempts to verify the signature using the ephemeral public key and proceeds with the transaction upon verification.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] Embodiments of the present invention will now be described by way of example only with reference to the accompanying drawings in which

- **Figure 1** is a schematic representation of a data communication system;
- **Figure 2** is a flow chart illustrating the exchange of information conducted on the system of figure 1 in a first embodiment;
- **Figure 3** is a flow chart illustrating the exchange of information conducted on the system of figure 1 in a second embodiment;
- **Figure 4** is a flow chart showing a third embodiment of the system of Figure 1;
- **Figure 5** is a flow chart showing a fourth embodiment of the system of Figure 1;
- **Figure 6** is a flow chart showing a fifth embodiment of the system of Figure 1.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0025] Referring therefore to figure 1, a data communication system 10 includes a pair of correspondents A, B, respectively identified as 12, 14, interconnected by a communication link 16. The correspondent B, 14, is also connected by a communication link 18 to a certifying authority, CA, indicated at 20. It will be appreciated that the links 16,18 are typically telephone lines or wireless links allowing the parties to route messages to intended recipients.

[0026] Each of the correspondents, 12, 14 and certifying authority 20 incorporate cryptographic units 22 that perform public-key cryptographic functions under the control of cryptographic software that may be embodied on a data carrier or programmed in an integrated circuit. Such implementations are well known and need not be described in detail, except to the extent necessary to appreciate the operation of the exchange of messages. For the purpose of this description it is assumed that each of the units 22 implement an elliptic curve public-key cryptosystem (ECC) operating in a field defined over F(q) but it will be appreciated that other implementations, such as those using $Z_p^* F_p^*$, the multiplicative group, of integers modulo a prime may be used.

[0027] The parameters for the ECC are an underlying cubic curve and a defined point P on the curve of order n. The correspondent A has an identity, ID_A, a short term or ephemeral private key k and a corresponding public key kP. The CA 20 is advised of the public key kP and identity ID_A which conveniently remain the same for all correspondence originating from the correspondent A.

[0028] To initiate an exchange of a message, *m*, for example a transaction record, between correspondents A and B, the message is sent by correspondent A to correspondent B over the communication channel 16. The message *m* is sent in the clear or in any other manner that may be read by correspondent B.

[0029] The correspondent B advises the certifying authority CA 20 that he has received a message from correspondent A and may also include some additional information relating to the nature of the transaction. This may be performed on a dedicated channel or may be encrypted if the information is considered to be of a sensitive nature. Upon receiving the information from correspondent B, the CA 20 checks the record of correspondent A and, if in order, prepares to return to the correspondent B the implicit certificate components, 24, identified as s_i, γ_i and A_i.

[0030] The component A_i includes the identity of A, i.e. ID_A, typically a unique distinguishing name or identity, for example a name, address or phone number that is stored by the CA 20 and a time stamp, message or similar transaction specific information,

[0031] The CA 20 also generates a random integer r and computes a corresponding public key rP. The value of γ_i is then computed from the relationship that $\gamma_i = kP + rP$.

[0032] The value of s_i is then computed. s_i is a signature component computed from one of the number of signing equations having a complementary public key reconstruction equation. In the embodiment described, the signing equation is selected as $s_i = r - c \cdot H(A_i, \gamma_i) \pmod n$ where c is a long term secret key of the CA 20, and H indicates a secure hash function such as SHA 1 or SHA 2.

[0033] The CA 20 forwards s_i, γ_i, and A_i to correspondent B. Since A_i contains transaction specific information, the implicit signature components γ_i, s_i are also transaction specific. It is preferable, but not necessary, that the CA signs the signature components forwarded to correspondent B.

[0034] Correspondent B, upon receipt of the communication from the CA 20, forwards the certificate component s_i to the correspondent A. It is preferable, but not necessary, that correspondent B signs the certificate component sent to correspondent A. The correspondent A computes a transaction specific private key a_i from the relationship $a_i = k+s_i$. The message *m* is then signed according to a selected signature scheme that utilizes the computed private key a_i and the signature is returned to the correspondent B. For example, a Nyberg Rueppel signature scheme may be implemented between the correspondents A and B. The correspondent A selects an ephemeral key pair w; W where w is a randomly selected integer and W is a corresponding point wP.

[0035] The signature on the message m is R, S where $R = W_x + H(m)$ (mod n) and $S = w - a_iR$ (mod n) and W_x is the x coordinate of the point wP.

[0036] The correspondent B then recovers the value corresponding to the transaction specific public key, a_iP, from the values of γ_i and A_i received from the CA 20. For the signing equation exemplified above, the public key a_iP can be computed from $a_iP= \gamma_i\text{-}H(A_i,\gamma_i)\cdot cP$ (mod n), where cP is the public key of the CA 20, and checks the signature on the message m. The verification equation for a Nyberg Rueppel schemes requires the computation of $sP + R(a_iP)$ which is the point W on the curve. The x coordinate of the point is selected and $R\text{-}W_x$ is computed. The result should correspond to H(m), which can be computed and verified by B. If the signature verifies, the message m is accepted and the transaction completed.

[0037] The implementation described above maintains a relatively small size of certificate and reduces the work performed by the correspondents A and B. The CA 20 is required to perform one implicit signature per transaction and correspondent B only requires one implicit signature verification and two signature verifications per transaction. Whereas prior proposals would require the CA 20 to return a message to the correspondent B stating that correspondent A has a valid certificate, this is avoided in the present embodiment by sending transaction specific implicit certificate components.

[0038] As described above, a common key kP is used for each transaction by correspondent A but if preferred a different key kP may be used to inhibit tracing of transactions originating at correspondent A. In this case new values of kP are sent to the CA 20 offline with appropriate levels of security.

[0039] In the above embodiment a specific computation of s_i and the public key reconstruction equation is given. It will be appreciated that other forms of s_i may be used. For example $s_i = rH(A_i\gamma_i)\text{-} c$ (mod n) could be used with a corresponding change to the public key reconstruction equation such that $a_iP = H(A_i\gamma_1)\gamma_1 = cP.$ With this scheme, the correspondents A and B may utilize an ECDSA signature scheme to exchange the messages, m, in which the signature is R, S with the component S of the form $k^{-1}(E+RD)$ where

K is an ephemeral private key,

R is an integer derived from the x coordinate of the point kP,

E is a hash of the message m, and

D is a long term private key.

In this embodiment, the computed private, a_i, is used for the long term private key D with K and R computed for each communication in the normal manner. For a ECDSA scheme, the verification is performed by computing $u_1=ES^{-1}$ mod (n) and $u_2\text{-}RS^{-1}$ mod (n). A value corresponding to R is computed from $u_1P+u_2(a_iP)$ and compared with the received value of R. If they correspond, the signature is verified, the message is accepted and the transaction completed.

[0040] An alternative arrangement is shown in figure 3, wherein like numerals with a prefix "1" refer to similar components as those of Figure 1, in which the originator of the message, correspondent A, communicates directly with the CA 120 who has previously been provided with the identity ID_A and the public key kP. In this arrangement the correspondent A notifies the CA 120 that a certificate is required. The CA 120 generates a certificate with components s_i, γ_i, A_i as before. The correspondent A then computes the transaction specific private key $a_i = k + s_i$ and uses it to sign the message *m*. The signed message is forwarded together with the explicit signature components γ_i and A_i to the correspondent B.

[0041] The correspondent B recovers the public key a_iP from A_i and γ_i and checks the signature on the message m. The transaction specific information in the component A_i is checked to determine if it is as expected. Verification of the transaction specific information after it has been recovered is known in the art and depends on the type of information being verified. If both the signature and the information are verified then the transaction is accepted.

[0042] Alternately, the CA 120 could send s_i to correspondent A and γ_i, A_i to correspondent B. Correspondent A can then sign message *m* using the private key $d_s = a + s_i$ and forward the message and signature to correspondent B.

[0043] The above protocol may also be used to provide implicit attributable certificates as shown in figure 4, wherein like numerals with a prefix "2" refer to similar components as those of Figure 1. Initially the values of ID_A and kP are transferred to the CA 220 from correspondent A. A request is then sent from correspondent A to the CA 220 to gain access to a particular application controlled by B.

[0044] The CA 220 generates a certificate including A_i, γ_i and s_i with A_i including the ID_A and an indication that the correspondent A can use a particular application and sends the certificate to A. A value of $a_i = k + s_i$ is generated by the correspondent A and used to sign the message m. The signed message is forwarded to correspondent B together with γ_i and A_i who recovers the corresponding public key a_iP. The signature is then checked and, if it verifies, access is given to the application. If the signature does not verify, the request is returned.

[0045] The above implicit attributable certificate is efficient in that it only requires one signed certificate and by using different public keys per application is hard to trace to a particular user. Moreover, the identity and the specific attributable certificate can be incorporated into one certificate rather than the two normally required.

[0046] Yet an alternate embodiment, similar to that illustrated in figure 3, is shown in figure 5. The CA 120 has a private key, c, and a public key, Q_C = cP. In order to acquire a certificate, correspondent A first generates a random integer, a. Integer a is used to compute a value aP, which is sent to the CA 120 along with correspondent A's identity, ID_A or, alternately, A_i (which may contain ID_A).

[0047] Upon receiving aP and ID_A from correspondent A, the CA 120 generates a random integer c_A and uses it to calculate correspondent A's certificate, $\gamma_A = aP + c_AP$. The CA 120 also calculates a signature component s_A of a suitable form. In the preferred embodiment, $s_A = H(\gamma_A \| ID_A \| cP)c + c_A (\text{mod } n)$. As an alternative, s_A could be computed from $s_A = H(\gamma_A \| ID_A \| cP)c_A + c (\text{mod } n)$. The certificate, γ_A and s_A are sent to correspondent A. Correspondent A's private key then becomes $d = a + s_A$, and its public key becomes $Q_A = d$P. Correspondent A's public key can be derived from the certificate according to the appropriate public key reconstruction equation, i.e. in the preferred embodiment $Q_A = h(\gamma_A \| ID_A \| cP)Q_C + \gamma_A$.

[0048] Therefore, if correspondent A wants to sign a message, m, to send to correspondent B, correspondent A does so using the private key, d. Correspondent A then sends the signed message along with the certificate, γ_A, and identification, ID_A. Upon receiving the information sent from correspondent A, correspondent B uses the certificate and identification along with the CA's public key, Q_C, for deriving correspondent A's public key, Q_A. The message is accepted if the signature is verified using correspondent A's derived public key, Q_A.

[0049] In the present embodiment, it is possible for the CA to efficiently recertify correspondent A. The CA generates a random number, $\overline{c_A}$ and computes $\overline{c_A}P$. Using the original value of aP received from correspondent A, the CA generates a new certificate, $\overline{\gamma_A} = \overline{c_A}P + aP$ and a new $\overline{s_A} = H(\overline{\gamma_A} \| ID_A \| cP)c + \overline{c_A} (\text{mod } n)$. The certificate, $\overline{\gamma_A}$, and $\overline{s_A}$ are sent to correspondent A. Therefore, correspondent A has a new private key, $\overline{d} = a + \overline{s_A}$, and a new certificate, $\overline{\gamma_A}$. Therefore, correspondent A's new public key, Q_A, can be derived according to $\overline{Q_A} = H(\overline{\gamma_A} \| ID_A \| cP)Q_C + \overline{\gamma_A}$.

[0050] Using such a recertification process can recertify correspondent A without requiring correspondent A to change its private key. However, this scheme requires sufficient bandwidth to send both s_A and γ_A *to* correspondent A. Furthermore, for each correspondent (such as correspondent A), the CA has to perform a point multiplication to obtain the new certificate, γ_A.

[0051] However, it is possible to make a modification to the recertification process as described above such that it is more efficient and requires less bandwidth. In the following example illustrated in figure 6, the CA recertifies all correspondents (including correspondent A). Also, it is assumed that correspondent A has been previously certified, acquired the certificate, γ_A, from the CA and determined the private key $d = a + s_A$.

[0052] The CA certifies the correspondents at the expiration of a certification period. For an i^{th} certification period, the CA generates a random value k_i and computes the value $Q_i = k_iP$. For each correspondent such as correspondent A, the CA computes $r_i = H(\gamma_A \| ID_A \| cP \| k_iP \| i)$and then $s_{A_i} = r_ic + k_i + c_A (\text{mod } n)$. Again, the CA could use other equations to produce s_{A_i}, for example $s_{A_i} = r_ic_A + c + k_i (\text{mod } n)$ with a corresponding public key reconstruction equation. Since the certificate does not change, it is only necessary for the CA to send s_{A_i} to correspondent A. The private key for correspondent A becomes $d_i = a + s_{A_i}$ and the certificate remains γ_A. The CA makes Q_i and i publicly available.

[0053] Therefore, it is possible to reconstruct correspondent A's public key, d_iP, by computing r_i, and then calculating $d_iP = r_iQ_C + \gamma_A + Q_i$. Correspondent A communicates with correspondent B similarly to the situation previously described. If correspondent A wants to sign a message to send to correspondent B, correspondent A does so using the private key, d_i. Correspondent A then sends the signed message along with the certificate, γ_A, and identification ID_A. Upon receiving the information sent from correspondent A, correspondent B uses the certificate and identification along with the CA's public keys, Q_C and Q_i, for deriving r_i. The values r_i, Q_c, Q_i, and γ_A are then used for deriving correspondent A's public key. The message is accepted if the signature is verified using correspondent A's derived public key.

[0054] Thus it can be seen that correspondent A's certificate does not change. Therefore, the CA is only required to send s_i and i to correspondent A for recertification, which requires essentially half the bandwidth of sending s_A and γ_A as in the previous example. Further, although the CA has to calculate $Q_i = k_iP$ for the ith certification period, the calculation is amortized over all the correspondents. That is, the CA only has to do one point multiplication for all the correspondents (for the calculation of Q_i). The CA also has to perform one modular multiplication for each correspondent (while calculating s_{A_i}). This results in a more efficient process than previously described wherein the CA has to perform one point multiplication and one modular multiplication for each correspondent.

[0055] Since the recertification scheme described above is not a costly operation for the CA, the CA could recertify correspondents more frequently than if traditional schemes are implemented. Therefore, one application of this recertification scheme is to replace revocation lists. Instead of providing a list of revoked certificates, the CA recertifies only those certificates that are still valid and have not been revoked.

[0056] In an alternate embodiment, the certificates as described in the previous embodiments are embedded into an RSA modulus itself. For an RSA encryption algorithm, correspondent A is required to provide a public key pair, (n, e), where n is the modulus and e is the public exponent. The modulus is defined as $n = pq$ where p and q are large prime numbers. The public exponent is selected as $1 < e < \phi$, where $\phi = (p-1)(q-1)$. It has been shown that a portion of the modulus can be set aside to have a predetermined value without increasing the vulnerability of the key. This method is described in detail in U.S. serial no. 08/449,357 filed May 24, 1995.

[0057] Embedding the certificate into the modulus reduces the bandwidth requirements since the certificate is included as part of the modulus instead of in addition to it. This implementation is particularly useful for a CA who signs using RSA and certifies using ECC. For example, a 2048-bit RSA modulus can easily contain a 160-bit ECC certificate.

Claims

1. A method of verifying a transaction over a data communication system between a first and second correspondent (12, 14) through the use of a certifying authority (20), said method comprising the steps of:

a) one of said first and second correspondents (12,14) advising said certifying authority (20) that a transaction is to be validated;
b) said certifying authority (20) determining whether to validate the transaction requested by said first or second correspondent (12, 14);
c) upon agreeing to validate said transaction, said certifying authority (20) generating implicit signature components (s_i, γ_i, A_i) including transaction specific information;
d) forwarding to said first correspondent (12) at least one of said implicit signature components (s_i) for permitting said first correspondent (12) to generate an ephemeral private key;
e) forwarding to said second correspondent (14) at least one of said implicit signature components (γ_i, A_i) for permitting recovery of an ephemeral public key ($\alpha_i P$) corresponding to said ephemeral private key (α_i);
f) said first correspondent (12) signing a message (m) with said ephemeral private key and forwarding said message (m) to said second correspondent (14) and
g) said second correspondent (14) attempting to verify said signature using said ephemeral public key ($\alpha_i P$) and proceeding with said transaction upon verification.

2. A method as defined in claim 1, wherein said second correspondent (14) advises said certification authority (20) that said transaction is to be validated upon receiving an initial message from said first correspondent (12).

3. A method as defined in claim 2, wherein said at least one of said implicit signature components (s_i) is forwarded to said second correspondent (14) by said certifying authority (20).

4. A method as defined in claim 3, wherein said at least one of said implicit signature components (s_i) is forwarded to said first correspondent (12) by said second correspondent (14).

5. A method as defined in claim 4, wherein said generated implicit signature components includes:

a) γ_i, where γ_i = kP + rP, and where k is a long term private key of said first correspondent (12), r is a random integer generated by said certification authority (20), and P is a point on a curve; and
b) s_i, where $s_i = r - c \cdot H(A_i, \gamma_i)$; and where c is a long term private key of said certifying authority (20), A_i includes at least one distinguishing feature ID_A of said first correspondent (12) and said transaction specific information, and H indicates a secure hash function;

wherein said long term public key kP of said first correspondent (12) is sent to said certifying authority (20) prior to said verification of said transaction.

6. A method as defined in claim 5, wherein A_i, γ_i, and s_i are forwarded to said second correspondent (14) and s_i is forwarded to said first correspondent (12).

7. A method as defined in claim 5, wherein said distinguishing feature ID_A includes at least one of a name of said first correspondent (12), a telephone number of said first correspondent (12), and an address of said first correspondent (12).

8. A method as defined in claim 5, wherein said transaction specific information includes at least one of a time of said transaction and a date of said transaction.

9. A method as defined in claim 6, wherein said ephemeral private key is generated according to $a_i = k+s_i$, where a_i is said ephemeral private key.

10. A method as defined in claim 9, wherein said ephemeral public key is recovered according to $a_iP=\gamma_i-H(A_{i,}\gamma_i) \cdot cP$, where a_iP is said ephemeral public key and cP is said certifying authority's (20) public key.

11. A method as defined in claim 10, wherein said certifying authority (20) verifies the validity of a request attributed to said first correspondent (12).

12. A method as defined in claim 10, wherein said ephemeral private key α_i is a transaction specific private key and said ephemeral public key $\alpha_i P$ is a transaction specific public key.

13. A method as defined in claim 2, wherein said first correspondent (12) advises said certification authority (20) that a request is to be validated.

14. A method as defined in claim 13, wherein said at least one of said implicit signature components (s_i) is forwarded to said first correspondent (12) by said certifying authority (20).

15. A method as defined in claim 14, wherein said at least one of said implicit signature components (s_i) is forwarded to said second correspondent (14) by said first correspondent (12).

16. A method as defined in claim 15, wherein said generated implicit signature components include:

a) γ_i, where γ_i = kP + rP, and where k is a long term private key of said first correspondent (12), r is a random integer generated by said certification authority (20), and P is a point on a curve; and
b) s_i, where $s_i = r - c \cdot H(A_i,\gamma_i)$, and where c is a long term private key of said certifying authority (20), A_i includes at least one distinguishing feature of said first correspondent (12) and said transaction specific information, and H indicates a secure hash function;

wherein said long term public key kP of said first correspondent (12) is sent to said certifying authority (20) prior to said verification of said transaction.

17. A method as defined in claim 16, wherein A_i, γ_i, and s_i are forwarded to said first correspondent (12), and A_i and γ_i are forwarded to said second correspondent (14).

18. A method as defined in claim 16, wherein said distinguishing feature ID_A includes at least one of a name of said first correspondent (12), a telephone number of said first correspondent (12), and an address of said first correspondent (12).

19. A method as defined in claim 16, wherein said transaction specific information includes at least one of a time of said transaction and a date of said transaction.

20. A method as defined in claim 17, wherein said ephemeral private key is generated according to $a_i = k+s_i$, where a_i is said ephemeral private key.

21. A method as defined in claim 20, wherein said ephemeral public key is recovered according to $a_iP=\gamma_i-H(A_i,\gamma_i)\cdot cP$, where a_iP is said ephemeral public key and cP is said certifying authority's (20) public key.

22. A method as defined in claim 21, wherein said certifying authority (20) verifies the validity of a request attributed to said first correspondent (12).

23. A method as defined in claim 21, wherein said ephemeral private key α_i is a transaction specific private key and said ephemeral public key α_iP is a transaction specific public key.

24. A method as defined in claim 15, wherein said generated implicit signature components A_i,γ_i,s_i include a parameter for indicating a predetermined permission for said first correspondent (12), said second correspondent (14) granting access to said first correspondent (12) according to said predetermined permission upon verification of said signature.

25. A method as defined in claim 15, wherein said generated implicit signature components include:

a) γ_A, where $\gamma_A = aP + c_AP,$ and where aP is a long term public key of said first correspondent (12), c_A is a random integer generated by said certifying authority (20), and P is a point on a curve; and.
b) s_A, where $s_A = h(\gamma_A \parallel A_i \parallel cP)c + c_A(\mathrm{mod}\ n)$, and where A_i includes at least one distinguishing feature of said first correspondent (12) and said transaction specific information, where c is a long term private key of said certifying authority (20), n is a large prime number, and h indicates a secure hash function.

26. A method as defined in claim 25, wherein γ_A and s_A are forwarded to said first correspondent (12), and A_i and γ_A are forwarded to said second correspondent (14) by said first correspondent (12).

27. A method as defined in claim 25, wherein said distinguishing feature ID_A is includes at least one of a name of said first correspondent (12), a telephone number of said first correspondent (12), and an address of said first correspondent (12).

28. A method as defined in claim 25, wherein said transaction specific information includes at least one of a time of said transaction and a date of said transaction.

29. A method as defined in claim 26, wherein said ephemeral private key is generated according to $d = a + s_A,$ where d is said ephemeral private key.

30. A method as defined in claim 29, wherein said ephemeral public key is recovered according to $Q_A = h(\gamma_A \| A_i \| Q_C)Q_C + \gamma_A$, where Q_A is said ephemeral public key and Q_C is said certifying authority's (20) long term public key.

31. A method as defined in claim 30, wherein said certifying authority (20) recertifies said implicit signature components attributed to said first correspondent (12) by changing said random integer, c_A.

32. A method as defined in claim 30, wherein said ephemeral private key is a transaction specific private key and said ephemeral public key is a transaction specific public key.

33. A method as defined in claim 15, wherein said generated implicit signature components include:

a) i, where i is a certification period;
b) s_A, where $s_{A_i} = r_i c + k_i + c_A \pmod n$, n is a large prime number, c is a long term private key of said certifying authority (20), c_A and k_i are random integers, and $r_i = h(\gamma_A \| A_i \| cP \| k_i P \| i)$, where A_i includes at least one distinguishing feature of said first correspondent (12) and said transaction specific information, P is a point on a curve, and h indicates a secure hash function;

wherein $\gamma_A = aP + c_A P$, and where aP is a long term public key of said first correspondent (12) and γ_A has previously been determined by said certifying authority (20) and forwarded to said first correspondent (12).

34. A method as defined in claim 33, wherein i and s_A are forwarded to said first correspondent (12), and A_i and γ_A are forwarded to said second correspondent (14) by said first correspondent (12).

35. A method as defined in claim 33, wherein said distinguishing feature ID_A is includes at least one of a name of said first correspondent (12), a telephone number of said first correspondent (12), and an address of said first correspondent (12).

36. A method as defined in claim 33, wherein said transaction specific information includes at least one of a time of said transaction and a date of said transaction.

37. A method as defined in claim 34, wherein said ephemeral private key is generated according to $d_i = a + s_{A_i}$, where d_i is said ephemeral private key.

38. A method as defined in claim 37, wherein said ephemeral public key is recovered according to $Q_A = r_i Q_C + \gamma_A + Q_i$, where Q_A is said ephemeral public key, Q_i is said certifying authority's (20) certification period public key, and Q_C is said certifying authority's (20) long term public key.

39. A method as defined in claim 38, wherein said certifying authority (20) recertifies said implicit signature components attributed to said first correspondent (12) for each certification period, i, by changing said random integer, k_i.

40. A method for certifying a correspondent (12, 14) in a data communication system through the use of a certifying authority (20) having control of a certificate's validity, said method comprising the steps of:

a) said certifying authority (20) generating a first random number (c_A);
b) generating transaction specific implicit signature components γ_A, s_i, based on said first random number (c_A);
c) publishing a public key Q_c of said certifying authority (20) for use in verifying said correspondent (12, 14);
d) forwarding said transaction specific implicit signature components from said certifying authority (20) to said correspondent (12, 14);

wherein said certifying authority (20) recertifies said correspondent's (12, 14) transaction specific implicit signature components by changing said value of said first random number (c_A).

41. A method as defined in claim 40, wherein c_A is said first random number generated by said certifying authority (20) and said transaction specific implicit signature components include:

a) γ_A, where $\gamma_A = aP + c_A P$, and where aP is a long term public key of said correspondent (12, 14) and P is a point on a curve; and
b) s_A, where $s_A = h(\gamma_A \| A_i \| cP)c + c_A \pmod n$, and where c is a long term private key of said certifying authority (20), n is a large prime number, A_i is an identifier of said correspondent (12, 14) and includes at least one distinguishing feature of said correspondent (12, 14) and transaction specific information, and h indicates a secure hash function;

42. A method as defined in claim 41, wherein said correspondent (12,14) is recertified by forwarding said transaction specific implicit signature components γ_A, s_A for said first random number (c_A) having said changed value from said certifying authority (20) to said correspondent (12,14).

43. A method as defined in claim 40, wherein said first random number (c_A) has said value for one certification period, said value being changed for other of said certifications periods.

44. A method as defined in claim 43, wherein k_i is said first random number generated by said certifying authority (20) for an *i*th certification period and said transaction specific implicit signature components include:

c) *i*, where *i* is a current certification period;
d) s_A, where $s_{A_i} = r_i c + k_i + c_A (\mathrm{mod}\ n)$, n is a large prime number, c is a long term private key of said certifying authority (20), c_A is a second random number, and $r_i = h(\gamma_A \parallel A_i \parallel cP \parallel k_i P \parallel i)$, where A_i includes at least one distinguishing feature of said correspondent (12, 14) and transaction specific information, P is a point on a curve, and h indicates a secure hash function;

wherein $\gamma_A = aP + c_A P$, and where aP is a long term public key of said correspondent (12, 14) and γ_A has previously been determined by said certifying authority (20) and forwarded to said correspondent (12, 14).

45. A method as defined in claim 44, wherein said published information further includes $k_i P$ and *i*.

46. A method as defined in claim 45, wherein said correspondent (12, 14) is recertified by forwarding said transaction specific implicit signature components for said first random number having said changed value from said certifying authority (20) to said correspondent (12, 14).

Patentansprüche

1. Verfahren zum Verifizieren einer Transaktion über ein Datenkommunikationssystem zwischen einem ersten und einem zweiten Teilnehmer (12, 14) unter Verwendung einer Zertifizierungsautorität (20), wobei das Verfahren die Schritte umfasst:

a) einer der ersten und zweiten Teilnehmer (12, 14) meldet der Zertifizierungsautorität (20), dass eine Transaktion zu validieren ist,
b) die Zertifizierungsautorität (20) ermittelt, ob die von dem ersten oder zweiten Teilnehmer (12, 14) geforderte Transaktion zu validieren ist,
c) nach der Zustimmung, die Transaktion zu validieren, generiert die Zertifizierungsautorität (20) implizite Signaturkomponenten (s_i, γ_i, A_i), die transaktionsspezifische Informationen enthalten,
d) Übermitteln von mindestens einer Komponente der impliziten Signaturkomponenten (s_i) an den ersten Teilnehmer (12) zum Zulassen, dass der erste Teilnehmer (12) einen kurzlebigen privaten Schlüssel generiert,
e) Übermitteln wenigstens einer Komponente der impliziten Signaturkomponenten (γ_i, A_i) an den zweiten Teilnehmer (14) zum Zulassen der Rücksendung eines kurzlebigen öffentlichen Schlüssels ($\alpha_i P$) entsprechend dem kurzlebigen privaten Schlüssel (α_i),
f) der erste Teilnehmer (12) signiert eine Nachricht (m) mit dem kurzlebigen privaten Schlüssel und übermittelt die Nachricht (m) an den zweiten Teilnehmer (14), und
g) der zweite Teilnehmer (14) versucht die Signatur unter Verwendung des kurzlebigen öffentlichen Schlüssels ($\alpha_i P$) zu verifizieren und macht nach der Verifikation mit der Transaktion weiter.

2. Verfahren nach Anspruch 1, wobei der zweite Teilnehmer (14) der Zertifizierungsautorität (20) meldet, dass die Transaktion nach dem Empfang einer Initialnachricht von dem ersten Teilnehmer (12) zu validieren ist.

3. Verfahren nach Anspruch 2, wobei die zumindest eine Komponente der impliziten Signaturkomponenten (s_i) durch die Zertifizierungsautorität (20) an den zweiten Teilnehmer (14) übermittelt wird.

4. Verfahren nach Anspruch 3, wobei die zumindest eine Komponente der impliziten Signaturkomponenten (s_i) durch den zweiten Teilnehmer (14) an den ersten Teilnehmer (12) übermittelt wird.

5. Verfahren nach Anspruch 4, wobei die erzeugten impliziten Signaturkomponenten enthalten:

a) γ_i, wobei $\gamma_i = kP + rP$ und k ein langlebiger privater Schlüssel des ersten Teilnehmers (12) ist, r ein Random Integer ist, das durch die Zertifizierungsautorität (20) generiert wurde, und P ein Punkt auf einer Kurve ist, und
b) s_i, wobei $s_i = r - c \cdot H(A_i, \gamma_i)$ und c ein langlebiger privater Schlüssel der Zertifizierungsautorität (20) ist, A_i wenigstens ein Unterscheidungsmerkmal ID_A des ersten Teilnehmers (12) und der transaktionsspezifischen Informationen ist und H eine sichere Hash-Funktion bezeichnet,

wobei der langlebige öffentliche Schlüssel kP des ersten Teilnehmers (12) vor der Verifikation der Transaktion an die Zertifizierungsautorität (20) gesendet wird.

6. Verfahren nach Anspruch 5, wobei A_i, γ_i und s_i an den zweiten Teilnehmer (14) übermittelt werden und s_i an den ersten Teilnehmer (12) übermittelt wird.

7. Verfahren nach Anspruch 5, wobei das Unterscheidungsmerkmal ID_A wenigstens eines der nachfolgenden Elemente enthält: ein Name des ersten Teilnehmers (12), eine Telefonnummer des ersten Teilnehmers (12) und eine Adresse des ersten Teilnehmers (12).

8. Verfahren nach Anspruch 5, wobei die transaktionsspezifische Informationen wenigstens die Zeit der Transaktion oder das Datum der Transaktion enthält.

9. Verfahren nach Anspruch 6, wobei der kurzlebige private Schlüssel gemäß $a_i = k + s_i$ generiert wird, wobei a_i der kurzlebige private Schlüssel ist.

10. Verfahren nach Anspruch 9, wobei der kurzlebige öffentliche Schlüssel gemäß $a_iP = \gamma_i - H(A_i,\gamma_i) \cdot cP$ wiederhergestellt wird, wobei a_iP der kurzlebige öffentliche Schlüssel und cP der öffentliche Schlüssel der Zertifizierungsautorität (20) ist.

11. Verfahren nach Anspruch 10, wobei die Zertifizierungsautorität (20) die Validität einer Anfrage, die dem ersten Teilnehmer (12) zuzuordnen ist, verifiziert.

12. Verfahren nach Anspruch 10, wobei der kurzlebige private Schlüssel (α_i) ein transaktionsspezifischer privater Schlüssel ist und der kurzlebige öffentliche Schlüssel (α_iP) ein transaktionsspezifischer öffentlicher Schlüssel ist.

13. Verfahren nach Anspruch 2, wobei der erste Teilnehmer (12) der Zertifizierungsautorität (20) meldet, dass eine Anfrage zu validieren ist.

14. Verfahren nach Anspruch 13, wobei zumindest eine Komponente der impliziten Signaturkomponenten (s_i) durch die Zertifizierungsautorität (20) an den ersten Teilnehmer (12) übermittelt wird.

15. Verfahren nach Anspruch 14, wobei wenigstens eine Komponente der impliziten Signaturkomponenten (s_i) durch den ersten Teilnehmer (12) an den zweiten Teilnehmer (14) übermittelt wird.

16. Verfahren nach Anspruch 15, wobei die generierten impliziten Signaturkomponenten enthalten:

a) γ_i, wobei $\gamma_i = kP + rP$ und k ein langlebiger privater Schlüssel des ersten Teilnehmers (12) ist, r ein Random Integer ist, das von der Zertifizierungsautorität (20) generiert ist, und P ein Punkt auf einer Kurve ist, und

b) s_i, wobei $s_i = r - c \cdot H(A_i, \gamma_i)$ und c ein langlebiger privater Schlüssel der Zertifizierungsautorität (20) ist, A_i wenigstens ein Unterscheidungsmerkmal des ersten Teilnehmers (12) und die transaktionsspezifischen Informationen enthält und H eine sichere Hash-Funktion bezeichnet,

wobei der langlebige öffentliche Schlüssel (kP) des ersten Teilnehmers (12) vor der Verifikation der Transaktion an die Zertifizierungsautorität (20) gesendet wird.

17. Verfahren nach Anspruch 16, wobei A_i, γ_i und s_i an den ersten Teilnehmer (12) übermittelt werden, und A_i und γ_i an den zweiten Teilnehmer (14) übermittelt werden.

18. Verfahren nach Anspruch 16, wobei das Unterscheidungsmerkmal ID_A wenigstens eines von ein Name des ersten Teilnehmers (12), eine Telefonnummer des ersten Teilnehmers (12) und eine Adresse des ersten Teilnehmers (12) enthält.

19. Verfahren nach Anspruch 16, wobei die transaktionsspezifische Informationen wenigstens eines von die Zeit der Transaktion und das Datum der Transaktion enthält.

20. Verfahren nach Anspruch 17, wobei der kurzlebige private Schlüssel gemäß $a_i = k + s_i$ generiert wird und a_i der kurzlebige private Schlüssel ist.

21. Verfahren nach Anspruch 20, wobei der kurzlebige öffentliche Schlüssel gemäß $a_iP = \gamma_i - H(A_i,\gamma_i) \cdot cP$ wiederhergestellt wird und a_iP der kurzlebige öffentliche Schlüssel ist und cP der öffentliche Schlüssel der Zertifizierungsautorität (20) ist.

22. Verfahren nach Anspruch 21, wobei die Zertifizierungsautorität (20) die Validität einer Anfrage, die dem ersten Teilnehmer (12) zuzuordnen ist, verifiziert.

23. Verfahren nach Anspruch 21, wobei der kurzlebige private Schlüssel α_i ein transaktionsspezifischer privater Schlüssel ist und der kurzlebige öffentliche Schlüssel α_iP ein transaktionsspezifischer öffentlicher Schlüssel ist.

24. Verfahren nach Anspruch 15, wobei die generierten impliziten Signaturkomponenten A_i, γ_i, s_i einen Parameter zum Kennzeichnen einer vorbestimmten Zulassung für den ersten Teilnehmer (12) enthalten, wobei der zweite Teilnehmer (14) einen Zugang zu dem ersten Teilnehmer (12) gemäß der vorbestimmten Zulassung nach Verifikation der Signatur gewährt.

25. Verfahren nach Anspruch 15, wobei die erzeugten impliziten Signaturkomponenten enthalten:

a) γ_A, wobei $\gamma_A = aP + c_AP$ und aP ein langlebiger öffentlicher Schlüssel des ersten Teilnehmers (12) ist, c_A ein Random Integer ist, das von der Zertifizierungsautorität (20) generiert ist, und P ein Punkt auf einer Kurve ist, und
b) s_A, wobei $s_A = h(\gamma_A \| A_i \| cP)c + c_A \pmod{n}$ und A_i wenigstens ein Unterscheidungsmerkmal des ersten Teilnehmers (12) und die transaktionsspezifischen Informationen enthält, wobei c ein langlebiger privater Schlüssel der Zertifizierungsautorität (20) ist, n eine große Primzahl ist und h eine sichere Hash-Funktion bezeichnet.

26. Verfahren nach Anspruch 25, wobei γ_A und s_A an den ersten Teilnehmer (12) übermittelt werden und A_i und γ_A durch den ersten Teilnehmer (12) an den zweiten Teilnehmer (14) übermittelt werden.

27. Verfahren nach Anspruch 25, wobei das Unterscheidungsmerkmal ID_A zumindest eines von ein Name des ersten Teilnehmers (12), eine Telefonnummer des ersten Teilnehmers (12) und eine Adresse des ersten Teilnehmers (12) enthält.

28. Verfahren nach Anspruch 25, wobei die transaktionsspezifische Informationen wenigstens eines von Zeit der Transaktion und das Datum Transaktion enthält.

29. Verfahren nach Anspruch 26, wobei der kurzlebige private Schlüssel gemäß $d = a + s_A$ generiert wird, wobei d der kurzlebige private Schlüssel ist.

30. Verfahren nach Anspruch 29, wobei der kurzlebige öffentliche Schlüssel gemäß $Q_A = h(\gamma_A \| A_i \| Q_C)Q_C + \gamma_A$ wiederhergestellt wird, wobei Q_A der kurzlebige öffentliche Schlüssel und Q_C der langlebige öffentliche Schlüssel der Zertifizierungsautorität (20) ist.

31. Verfahren nach Anspruch 30, wobei die Zertifizierungsautorität (20) die impliziten Signaturkomponenten, die dem ersten Teilnehmer (12) zugeordnet sind, durch Ändern des Random Integer c_A neu zertifiziert.

32. Verfahren nach Anspruch 30, wobei der kurzlebige private Schlüssel ein transaktionsspezifischer privater Schlüssel ist und der kurzlebige öffentliche Schlüssel ein transaktionsspezifischer öffentlicher Schlüssel ist.

33. Verfahren nach Anspruch 15, wobei die erzeugten impliziten Signaturkomponenten enthalten:

a) i, wobei i eine Zertifizierungsperiode ist,
b) s_A, wobei $s_{A_i} = r_ic + k_i + c_A \pmod{n}$ und n eine große Primzahl ist, c ein langlebiger privater Schlüssel der Zertifizierungsautorität (20) ist, c_A und k_i Random Integers sind und $r_i = h(\gamma_A \| A_i \| cP \| k_i P \| i)$ ist, wobei A_i zumindest ein Unterscheidungsmerkmal des ersten Teilnehmers (12) und die transaktionsspezifischen Informationen enthält, P ein Punkt auf einer Kurve ist und h eine sichere Hash-Funktion bezeichnet,

wobei $\gamma_A = aP + c_AP$ ist und aP ein langlebiger öffentlicher Schlüssel des ersten Teilnehmers (12) ist und γ_A zuvor durch die Zertifizierungsautorität (20) festgelegt und an den ersten Teilnehmer (12) übermittelt wurde.

34. Verfahren nach Anspruch 33, wobei i und s_A an den ersten Teilnehmer (12) übermittelt werden und A_i und γ_A durch den ersten Teilnehmer (12) an den zweiten Teilnehmer (14) übermittelt werden.

35. Verfahren nach Anspruch 33, wobei das Unterscheidungsmerkmal ID_A zumindest eines von ein Name des ersten Teilnehmers (12), eine Telefonnummer des ersten Teilnehmers (12) und eine Adresse des ersten Teilnehmers (12) enthält.

36. Verfahren nach Anspruch 33, wobei die transaktionsspezifischen Informationen mindestens eines von Zeit der Transaktion und Datum der Transaktion enthält.

37. Verfahren nach Anspruch 34, wobei der kurzlebige private Schlüssel gemäß $d_i = a + s_{A_i}$ generiert wird, wobei d_i der kurzlebige private Schlüssel ist.

38. Verfahren nach Anspruch 37, wobei der kurzlebige öffentliche Schlüssel gemäß $Q_A = r_iQ_C + \gamma_A + Q_i$ wiederhergestellt wird, wobei Q_A der kurzlebige öffentliche Schlüssel ist, Q_i der für eine Zertifizierungsperiode gültige öffentliche Schlüssel der Zertifizierungsautorität (20) ist und Q_C der langlebige öffentliche Schlüssel der Zertifizierungsautorität (20) ist.

39. Verfahren nach Anspruch 38, wobei die Zertifizierungsautorität (20) die impliziten Signaturkomponenten, die dem ersten Teilnehmer (12) zugeordnet sind, für jede Zertifizierungsperiode i durch Veränderung des Random Integers k_i neu zertifiziert.

40. Verfahren zum Zertifizieren eines Teilnehmers (12, 14) in einem Datenkommunikationssystem unter Verwendung einer Zertifizierungsautorität (20), die Kontrolle über die Validität eines Zertifikats hat, wobei das Verfahren die Schritte umfasst:

a) die Zertifizierungsautorität (20) erzeugt eine erste Zufallszahl (c_A),
b) Erzeugen von transaktionsspezifischen impliziten Signaturkomponenten γ_A, s_i, die auf der ersten Zufallszahl (c_A) basieren,

c) Veröffentlichen eines öffentlichen Schlüssels Q_c der Zertifizierungsautorität (20) zur Verwendung bei der Verifikation des Teilnehmers (12, 14),

d) Übermitteln der transaktionsspezifischen impliziten Signaturkomponenten von der Zertifizierungsautorität (20) zu dem Teilnehmer (12, 14),

wobei die Zertifizierungsautorität (20) die transaktionsspezifischen impliziten Signaturkomponenten von dem Teilnehmer (12, 14) durch Verändern des Werts der ersten Zufallszahl (c_A) neu zertifiziert.

41. Verfahren nach Anspruch 40, wobei c_A die erste Zufallszahl ist, die durch die Zertifizierungsautorität (20) generiert wurde, und die transaktionsspezifischen impliziten Signaturkomponenten enthalten:

a) γ_A, wobei $\gamma_A = aP + c_AP$ und aP ein langlebiger öffentlicher Schlüssel des Teilnehmers (12, 14) und P ein Punkt auf einer Kurve ist, und

b) s_A, wobei $s_A = h(\gamma_A \| A_i \| cP)c + c_A \pmod n$ und c ein langlebiger privater Schlüssel der Zertifizierungsautorität (20) ist, n eine große Primzahl ist, A_i ein Identifier des Teilnehmers (12, 14) ist und wenigstens ein Unterscheidungsmerkmal des Teilnehmers (12, 14) und die transaktionsspezifischen Informationen enthält, und h eine sichere Hash-Funktion bezeichnet.

42. Verfahren nach Anspruch 41, wobei der Teilnehmer (12, 14) neu zertifiziert wird durch Übermitteln der transaktionsspezifischen impliziten Signaturkomponenten γ_A, s_A für die erste Zufallszahl (c_A), die den veränderten Wert von der Zertifizierungsautorität (20) hat, an den Teilnehmer (12, 14).

43. Verfahren nach Anspruch 40, wobei die erste Zufallszahl (c_A) den Wert für eine Zertifizierungsperiode hat, wobei der Wert für andere Zertifizierungsperioden verändert ist.

44. Verfahren nach Anspruch 43, wobei k_i die erste Zufallszahl ist, die durch die Zertifizierungsautorität (20) für eine i-te Zertifizierungsperiode generiert ist, und die transaktionsspezifischen impliziten Signaturkomponenten enthalten:

c) i, wobei i eine momentane Zertifizierungsperiode ist,

d) s_A, wobei $s_{A_i} = r_ic + k_i + c_A \pmod n$, n eine große Primzahl ist, c ein langlebiger privater Schlüssel der Zertifizierungsautorität (20) ist, c_A eine zweite Zufallszahl ist und $r_i = h(\gamma_A \| A_i \| cP \| k_iP \| i)$, wobei A_i wenigstens ein Unterscheidungsmerkmal des Teilnehmer (12, 14) und die transaktionsspezifischen Informationen enthält, P ein Punkt auf einer Kurve ist und h eine sichere Hash-Funktion bezeichnet,

wobei $\gamma_A = aP + c_AP$ und aP ein langlebiger öffentlicher Schlüssel des Teilnehmers (12, 14) ist und γ_A zuvor durch die Zertifizierungsautorität (20) festgelegt und an den Teilnehmer (12, 14) übermittelt wurde.

45. Verfahren nach Anspruch 44, wobei die veröffentlichten Informationen ferner k_iP und i enthalten.

46. Verfahren nach Anspruch 45, wobei der Teilnehmer (12, 14) neu zertifiziert wird durch Übermitteln der transaktionsspezifischen impliziten Signaturkomponenten für die erste Zufallszahl, die einen veränderten Wert von der Zertifizierungsautorität (20) hat, an den Teilnehmer (12, 14).

Revendications

1. Procédé pour vérifier une transaction via un système de communication de données entre un premier et un second correspondant (12, 14) par l'intermédiaire de l'utilisation d'une autorité (20) de certification, ledit procédé comprenant les étapes consistant à:

a) l'un parmi ledit premier et ledit second correspondant (12, 14) avise ladite autorité de certification (20) qu'une transaction doit être validée;

b) ladite autorité de certification (20) détermine s'il s'agit de valider la transaction requise par ledit premier ou ledit second correspondant (12);

c) lorsqu'elle donne son accord pour valider ladite transaction, ladite autorité de certification (20) génère des composantes de signature implicite (s_i, γ_i, A_i) qui incluent des informations spécifiques à la transaction;

d) l'envoi audit premier correspondant (12) de l'une au moins desdites composantes de signature implicite (s_i) pour permettre audit premier correspondant (12) de générer une clé privée éphémère;

e) l'envoi audit second correspondant (14) de l'une au moins desdites composantes de signature implicite (γ_i, A_i) pour permettre la récupération d'une clé publique éphémère (α_i, P) correspondant à ladite clé privée éphémère (α_i);

f) ledit premier correspondant (12) signe un message (m) avec ladite clé privée éphémère et envoie ledit message (m) audit second correspondant (14), et

g) ledit second correspondant (14) tente de vérifier ladite signature en utilisant ladite clé publique éphémère (α_i, P) et procède à ladite transaction en cas de vérification.

2. Procédé selon la revendication 1, dans lequel ledit second correspondant (14) avise ladite autorité de certification (20) que ladite transaction doit être validée lors de la réception d'un message initial provenant dudit premier correspondant (12).

3. Procédé selon la revendication 2, dans lequel ladite au moins une composante de signature implicite (s_i) est envoyée audit second correspondant (14) par ladite autorité de certification (20).

4. Procédé selon la revendication 3, dans lequel ladite au moins une composante de signature implicite (s_i) est envoyée audit premier correspondant (12) par ledit second correspondant (14).

5. Procédé selon la revendication 4, dans lequel lesdites composantes de signature implicite générées incluent :

a) γ_i, tel que $\gamma_i = kP + rP$, et dans laquelle k est une clé privée à long terme dudit premier correspondant (12), r est un entier aléatoire généré par ladite autorité de certification (20), et P est un point sur une courbe ; et
b) s_i, tel que $s_i = r - c{\cdot}H(A_i, \gamma_i)$, et dans laquelle c est une clé privée à long terme de ladite autorité de certification (20), A_i inclut au moins une caractéristique distinctive ID_A dudit premier correspondant (12) et ladite information spécifique à la transaction, et H indique une fonction de compression sécurisée ;

dans lequel ladite clé publique à long terme kP dudit premier correspondant (12) est envoyée à ladite autorité de certification (20) avant ladite vérification de ladite transaction.

6. Procédé selon la revendication 5, dans lequel A_i, γ_i, et s_i sont envoyés audit second correspondant (14) et s_i est envoyé audit premier correspondant (12).

7. Procédé selon la revendication 5, dans lequel ladite caractéristique distinctive ID_A inclut au moins un élément parmi un nom dudit premier correspondant (12), un numéro de téléphone dudit premier correspondant (12) et une adresse dudit premier correspondant (12).

8. Procédé selon la revendication 5, dans lequel ladite information spécifique à la transaction inclut au moins un paramètre parmi l'heure de ladite transaction et la date de ladite transaction.

9. Procédé selon la revendication 6, dans lequel ladite clé privée éphémère est générée en accord avec $a_i = k + s_i$, dans laquelle a_i est ladite clé privée éphémère.

10. Procédé selon la revendication 9, dans lequel ladite clé publique éphémère est récupérée selon $a_iP = \gamma_i - H(A_i, \gamma_i){\cdot}cP$, dans laquelle a_iP est ladite clé publique éphémère et cP est ladite clé publique de ladite autorité de certification (20).

11. Procédé selon la revendication 10, dans lequel ladite autorité de certification (20) vérifie la validité d'une requête attribuée audit premier correspondant (12).

12. Procédé selon la revendication 10, dans lequel ladite clé privée éphémère a_i est une clé privée spécifique à la transaction et ladite clé publique éphémère a_iP est une clé publique spécifique à la transaction.

13. Procédé selon la revendication 2, dans lequel ledit premier correspondant (12) avise ladite autorité de certification (20) qu'une requête doit être validée.

14. Procédé selon la revendication 13, dans lequel ladite au moins une composante de signature implicite (s_i) est envoyée audit premier correspondant (12) par ladite autorité de certification (20).

15. Procédé selon la revendication 14, dans lequel ladite au moins une composante de signature implicite (s_i) est envoyée audit second correspondant (14) par ledit premier correspondant (12).

16. Procédé selon la revendication 15, dans lequel lesdites composantes de signature implicite générées incluent:

a) γ_i, tel que $\gamma_i = kP + rP$, et dans laquelle k est une clé privée à long terme dudit premier correspondant (12), r est un entier aléatoire généré par ladite autorité de certification (20), et P est un point sur une courbe; et
b) s_i, tel que $s_i = r - c{\cdot}H(A_i, \gamma_i)$, et dans laquelle c est une clé privée à long terme de ladite autorité de certification (20), A_i inclut au moins une caractéristique distinctive dudit premier correspondant (12) et ladite information spécifique à la transaction, et H indique une fonction de compression sécurisée;

dans lequel ladite clé publique à long terme kP dudit premier correspondant (12) est envoyée à ladite autorité de certification (20) avant ladite vérification de ladite transaction.

17. Procédé selon la revendication 16, dans lequel A_i, γ_i, et s_i sont envoyés audits premiers correspondant (12), et A_i et γ_i, sont envoyés audit second correspondant (14).

18. Procédé selon la revendication 16, dans lequel ladite caractéristique distinctive ID_A inclut au moins un élément parmi un nom dudit premier correspondant (12), un numéro de téléphone dudit premier correspondant (12), et une adresse dudit premier correspondant (12).

19. Procédé selon la revendication 16, dans lequel lesdites informations spécifiques à la transaction incluent au moins un élément parmi l'heure de ladite transaction et la date de ladite transaction.

20. Procédé selon la revendication 17, dans lequel ladite clé privée éphémère est générée selon $a_i = k + s_i$, dans laquelle a_i est ladite clé privée éphémère.

21. Procédé selon la revendication 20, dans lequel ladite clé publique éphémère est récupérée selon $a_iP = \gamma_i - H(A_i, \gamma_i) \cdot cP$, dans laquelle a_iP est ladite clé publique éphémère et cP est ladite clé publique de ladite autorité de certification (20).

22. Procédé selon la revendication 21, dans lequel ladite autorité de certification (20) vérifie la validité d'une requête attribuée audit premier correspondant (12).

23. Procédé selon la revendication 21, dans lequel ladite clé privée éphémère a_i est une clé privée spécifique à la transaction, et ladite clé publique a_iP éphémère est une clé publique spécifique à la transaction.

24. Procédé selon la revendication 15, dans lequel lesdites composantes de signature implicite générées A_i, γ_i, et s_i incluent un paramètre pour indiquer une permission prédéterminée pour ledit premier correspondant (12), ledit second correspondant (14) accordant un accès audit premier correspondant (12) selon ladite permission prédéterminée lors de la vérification de ladite signature.

25. Procédé selon la revendication 15, dans lequel lesdites composantes de signature implicite générées incluent:

a) γ_A, telle que $\gamma_A = aP + c_AP$, et dans laquelle aP est une clé publique à long terme dudit premier correspondant (12), c_A est un entier aléatoire généré par ladite autorité de certification (lever) et P est un point sur une courbe; et
b) s_A, tel que $s_A = h(\gamma_A \| A_i \| cP)c + c_A (mod\ n)$, et dans laquelle A_i inclut au moins une caractéristique distinctive dudit premier correspondant (12) et lesdites informations spécifiques à la transaction, c est une clé privée à long terme de ladite autorité de certification, n est un nombre premier important, et h indique une fonction de compression sécurisée.

26. Procédé selon la revendication 25, dans lequel γ_A et s_A sont envoyés audit premier correspondant (12), et sont A_i et γ_A sont envoyés audit second correspondant (14) par ledit premier correspondant (12).

27. Procédé selon la revendication 25, dans lequel ladite caractéristique distinctive ID_A inclut un élément au moins parmi un nom dudit premier correspondant (12), un numéro de téléphone dudit premier correspondant (12), et une adresse dudit premier correspondant (12).

28. Procédé selon la revendication 25, dans lequel lesdites informations spécifiques à la transaction incluent un élément au moins parmi l'heure de ladite transaction et la date de ladite transaction.

29. Procédé selon la revendication 26, dans lequel ladite clé privée éphémère est générée selon $d = a + sA$, dans laquelle d est ladite clé privée éphémère.

30. Procédé selon la revendication 29, dans lequel ladite clé publique éphémère est récupérée selon $Q_A = h(\gamma_A \| A_i \| Q_C)Q_C + \gamma_A$, dans laquelle Q_A est ladite clé publique éphémère et Q_C est ladite de clé publique à long terme de ladite autorité de certification (20).

31. Procédé selon la revendication 30, dans lequel ladite autorité de certification (20) certifie à nouveau lesdites composantes de signature implicite attribuées audit premier correspondant (12) en changeant ledit entier aléatoire, c_A.

32. Procédé selon la revendication 30, dans lequel ladite clé privée éphémère est une clé privée spécifique à la transaction, et ladite clé publique éphémère est une clé publique spécifique à la transaction.

33. Procédé selon la revendication 15, dans lequel lesdites composantes de signature implicite générée incluent :

à) i, tel que i est une période de certification ;
b) s_A, tel que $s_A = r_ic + k_i + c_A (mod\ n)$, dans laquelle n est un nombre premier important, c est une clé privée à long terme de ladite autorité de certification (20), c_A et k_i sont des entiers aléatoires, et $r_i = h(\gamma_A \| A_i \| cP \| k_iP \| i)$, dans laquelle A_i inclut au moins une caractéristique distinctive dudit premier correspondant (12) et lesdites informations spécifiques à la transaction, P est un point sur une courbe, et h indique une fonction de compression sécurisée ;

dans lequel $\gamma_A = aP + c_AP$, et dans laquelle aP est une clé publique à long terme dudit premier correspondant (12), et γ_A a été précédemment déterminé par ladite autorité de certification (20) et envoyé audit premier correspondant (12).

34. Procédé selon la revendication 33, dans lequel i et s_A sont envoyés audit premier correspondant (12) et A_i et γ_A sont envoyés audit second correspondant (14) par ledit premier correspondant (12).

35. Procédé selon la revendication 33, dans lequel ladite caractéristique distinctive ID_A inclut un élément au moins parmi un nom dudit premier correspondant (12), un numéro de téléphone dudit premier correspondant (12), et une adresse dudit premier correspondant (12).

36. Procédé selon la revendication 33, dans lequel lesdites informations spécifiques à la transaction incluent un élément au moins parmi l'heure de ladite transaction et la date de ladite transaction.

37. Procédé selon la revendication 34, dans lequel ladite clé privée éphémère est générée selon $d_i = a + s_A$, dans laquelle d_i est ladite clé privée éphémère.

38. Procédé selon la revendication 37, dans lequel ladite clé publique éphémère est récupérée selon $Q_A = r_iQ_C + \gamma_A + Q_i$, dans laquelle Q_A est ladite clé publique éphémère, Q_i est ladite clé publique pour une période de certification de l'autorité de certification (20), et Q_C est ladite clé publique à long terme de ladite autorité de certification (20).

39. Procédé selon la revendication 38, dans lequel ladite autorité de certification (20) certifie à nouveau lesdites composantes de signature implicite attribuées audit premier correspondant (12) pour chaque période de certification i, en changeant ledit entier aléatoire, k_i.

40. Procédé pour certifier un correspondant (12, 14) dans un système de communication de données en utilisant une autorité de certification (20) ayant le contrôle sur la validité d'un certificat, ledit procédé comprenant les étapes consistant à :

a) ladite autorité de certification (20) génère un premier nombre aléatoire (c_A);
b) génération de composantes de signature implicite spécifiques à la transaction (γ_A, s_i) en se basant sur ledit premier nombre aléatoire (c_A);
c) publication d'une clé publique Q_C de ladite autorité de certification (20) à utiliser pour vérifier ledit correspondant (12, 14) ;
d) fourniture desdites composantes de signature implicite spécifiques à la transaction depuis ladite autorité de certification (20) audit correspondant (12, 14) ;

dans lequel ladite autorité de certification (20) certifie à nouveau lesdites composantes de signature implicite spécifiques à la transaction dudit correspondant (12, 14) en changeant ladite valeur dudit premier nombre aléatoire (c_A).

41. Procédé selon la revendication 40, dans lequel c_A est ledit premier nombre aléatoire généré par ladite autorité de certification (20) et lesdites composantes de signature implicite spécifiques à la transaction incluent :

a) γ_A, tel que $\gamma_A = aP + c_AP$, et dans laquelle aP est une clé publique à long terme dudit correspondant (12, 14) et P est un point sur une courbe ; et
b) s_A, tel *que* $s_A = h(\gamma_A \| A_i \| cP)c + c_A(mod\ n)$, et dans laquelle c est une clé privée à long terme de ladite autorité de certification (20), n est un nombre premier important, A_i est un identificateur dudit correspondant (12, 14) et inclut au moins une caractéristique distinctive dudit correspondant (12, 14) et des informations spécifiques à la transaction, et h indique une fonction de compression sécurisée.

42. Procédé selon la revendication 41, dans lequel ledit correspondant (12, 14) est à nouveau certifié en envoyant lesdites composantes de signature implicite spécifiques à la transaction (γ_A, s_A) pour ledit premier nombre aléatoire (c_A) qui possède ladite valeur changée, depuis ladite autorité de certification (20) vers ledit correspondant (12, 14).

43. Procédé selon la revendication 40, dans lequel ledit premier nombre aléatoire a une valeur pour une période de certification, ladite valeur étant changée pour d'autres périodes de certification.

44. Procédé selon la revendication 43, dans lequel k_i est ledit premier nombre aléatoire généré par ladite autorité de certification (20) pour une i-ème période de certification et lesdites composantes de signature implicite spécifiques à la transaction incluent :

c) i, tel que i est une période de certification courante ;
d) s_A, tel que $s_A = r_ic + k_i + c_A(mod\ n)$, n est un nombre premier important, c est une clé privée à long terme de ladite autorité de certification (20), c_A est un second nombre aléatoire, et $r_i = h(\gamma_A \| A_i \| cP \| k_iP \| i)$, dans laquelle A_i inclut au moins une caractéristique distinctive dudit correspondant (12, 14) et des informations spécifiques à la transaction, P est un point sur une courbe, et h indique une fonction de compression sécurisée ;

dans lequel $\gamma_A = aP + c_AP$, dans laquelle aP est une clé publique à long terme pour ledit correspondant (12, 14) et γ_A a été déterminé précédemment par ladite autorité de certification (20) et envoyé audit correspondant (12, 14).

45. Procédé selon la revendication 44, dans lequel ladite information publiée inclut en outre k_iP et i.

46. Procédé selon la revendication 45, dans lequel ledit correspondant (12, 14) est à nouveau certifié en envoyant lesdites composantes de signature implicite spécifiques à la transaction pour ledit premier nombre aléatoire ayant ladite valeur changée depuis ladite autorité de certification (20) vers ledit correspondant (12, 14).

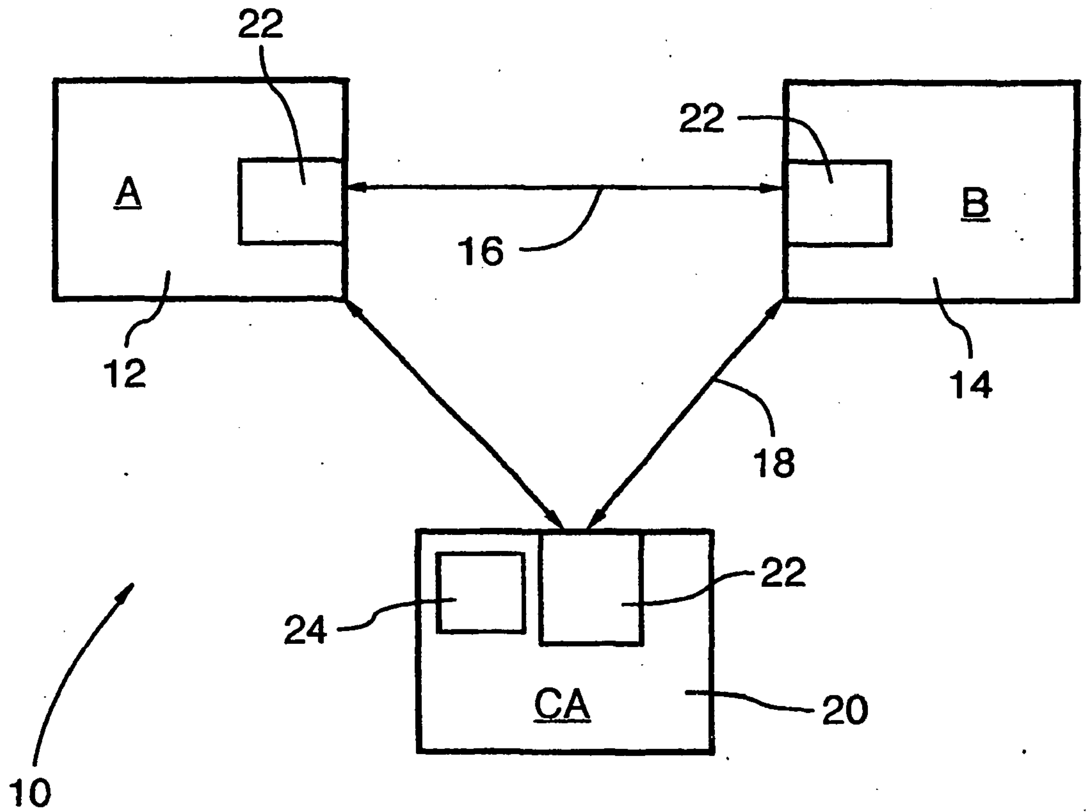

FIG.1

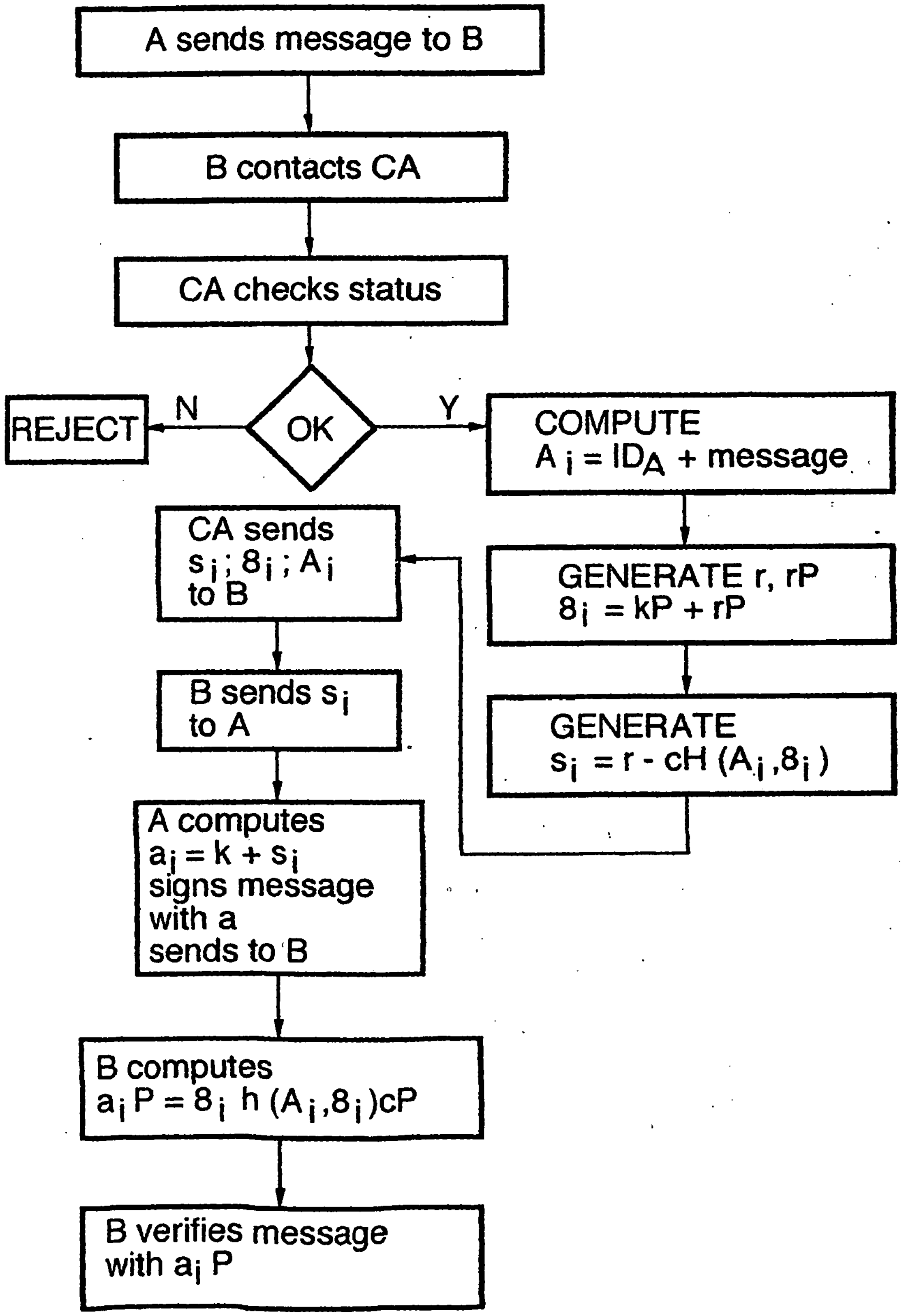
A sends message to B
B contacts CA
CA checks status
OK
N
REJECT
Y
COMPUTE
A i = IDA + message
GENERATE r, rP
8i = kP + rP
GENERATE
si = r - cH (Ai ,8i)
CA sends
si ; 8i ; Ai
to B
B sends si
to A
A computes
ai = k + si
signs message
with a
sends to B
B computes
ai P = 8i h (Ai ,8i)cP
B verifies message
with ai P

FIG.2

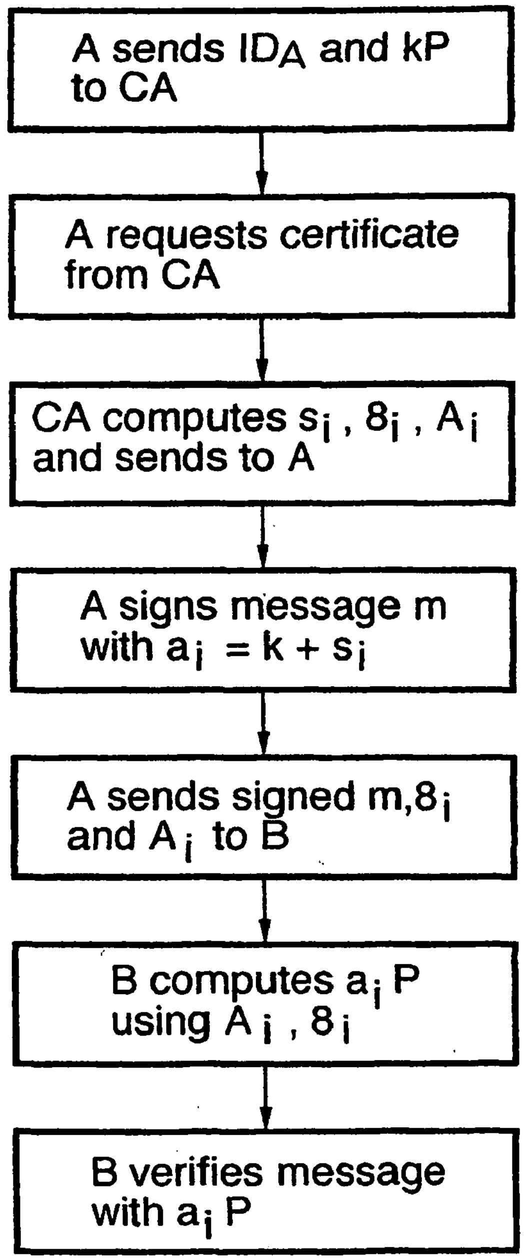
A sends ID_A and kP to CA
A requests certificate from CA
CA computes s_i, 8_i, A_i and sends to A
A signs message m with $a_i = k + s_i$
A sends signed m, 8_i and A_i to B
B computes $a_i P$ using A_i, 8_i
B verifies message with $a_i P$

FIG.3

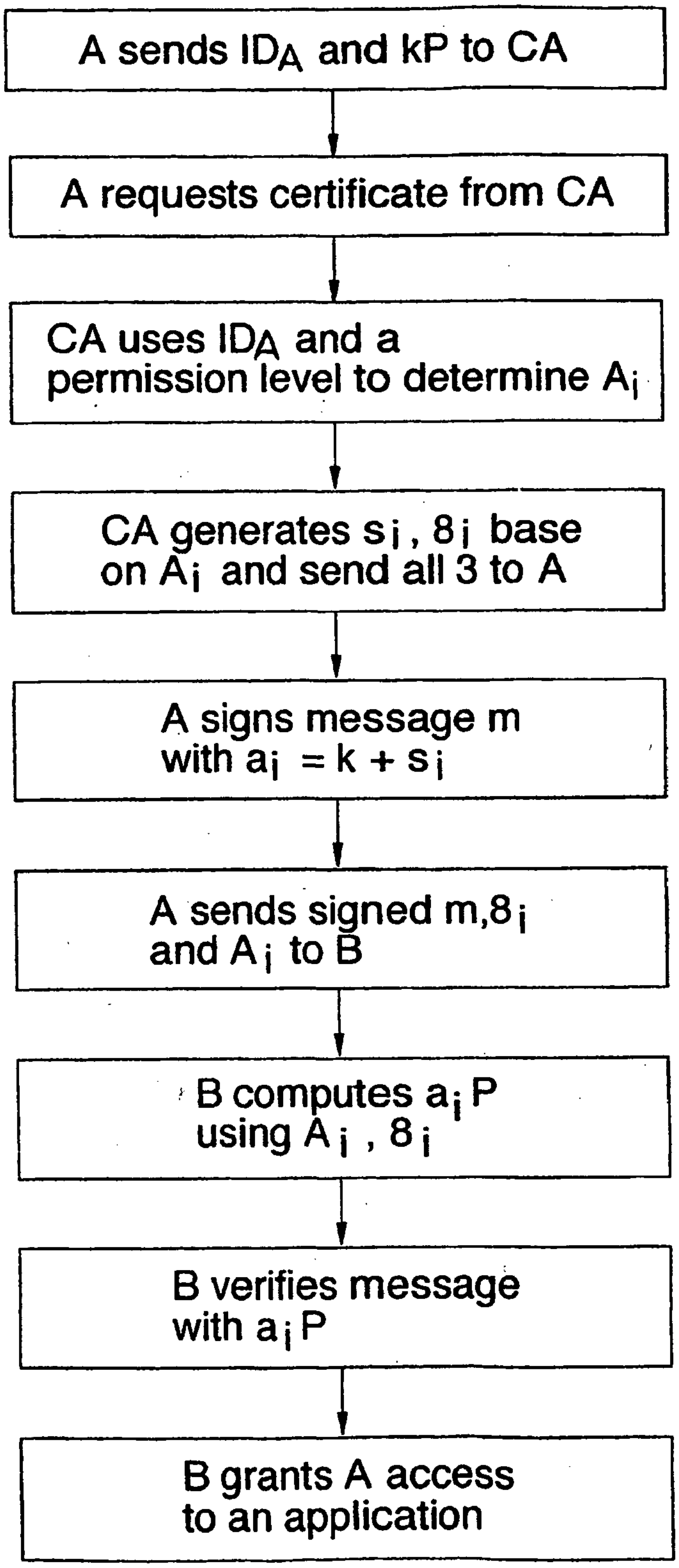
A sends ID$_A$ and kP to CA
A requests certificate from CA
CA uses ID$_A$ and a permission level to determine A$_i$
CA generates s$_i$, 8$_i$ base on A$_i$ and send all 3 to A
A signs message m with a$_i$ = k + s$_i$
A sends signed m,8$_i$ and A$_i$ to B
B computes a$_i$P using A$_i$, 8$_i$
B verifies message with a$_i$P
B grants A access to an application

FIG.4

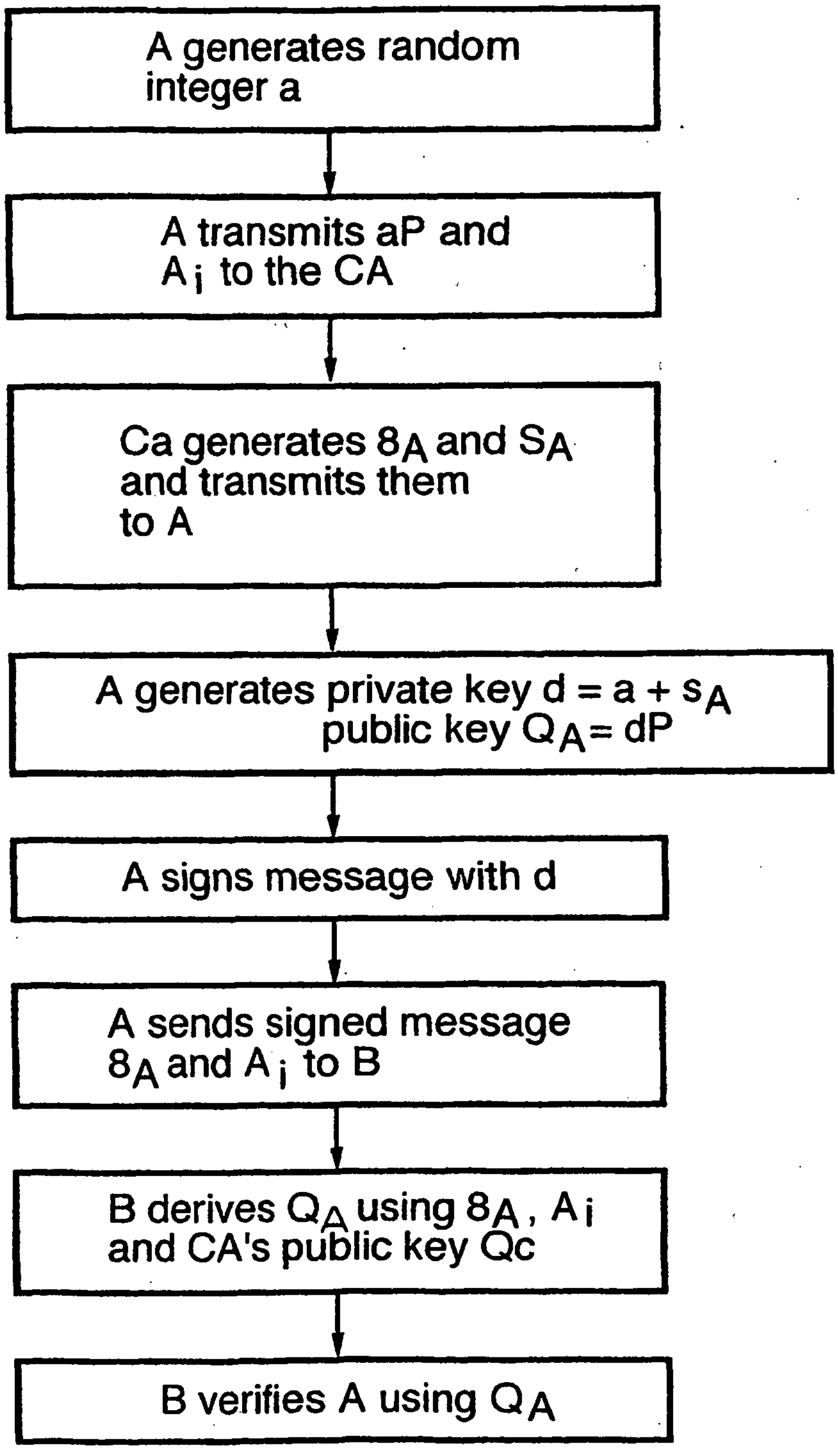
A generates random integer a
A transmits aP and A_i to the CA
Ca generates 8_A and S_A and transmits them to A
A generates private key $d = a + s_A$ public key $Q_A = dP$
A signs message with d
A sends signed message 8_A and A_i to B
B derives Q_A using 8_A, A_i and CA's public key Qc
B verifies A using Q_A

FIG.5

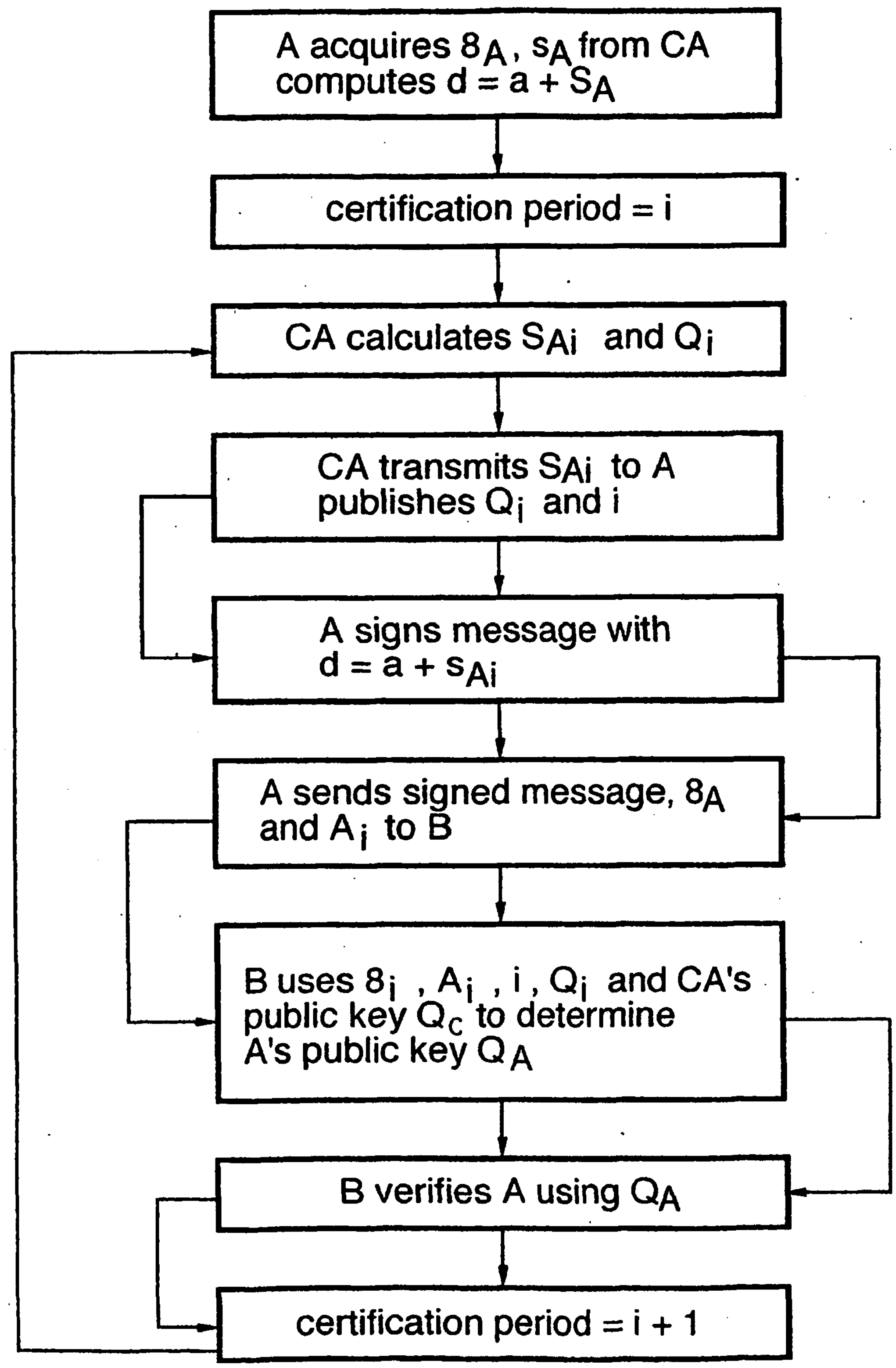
A acquires 8A, sA from CA
computes d = a + SA
certification period = i
CA calculates SAi and Qi
CA transmits SAi to A
publishes Qi and i
A signs message with
d = a + sAi
A sends signed message, 8A
and Ai to B
B uses 8i , Ai , i, Qi and CA's
public key Qc to determine
A's public key QA
B verifies A using QA
certification period = i + 1

FIG.6

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 9949612 A, QU **[0002]**
- CA 2232936 **[0014]**
- US 08449357 B **[0056]**

Non-patent literature cited in the description

- Intranet Security Framework based on Short-Lived Certificates. **Hsu et al.** Proceedings Sixth IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. IEEE Comput. SOC, 20 June 1997, 228-233 **[0003]**