OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

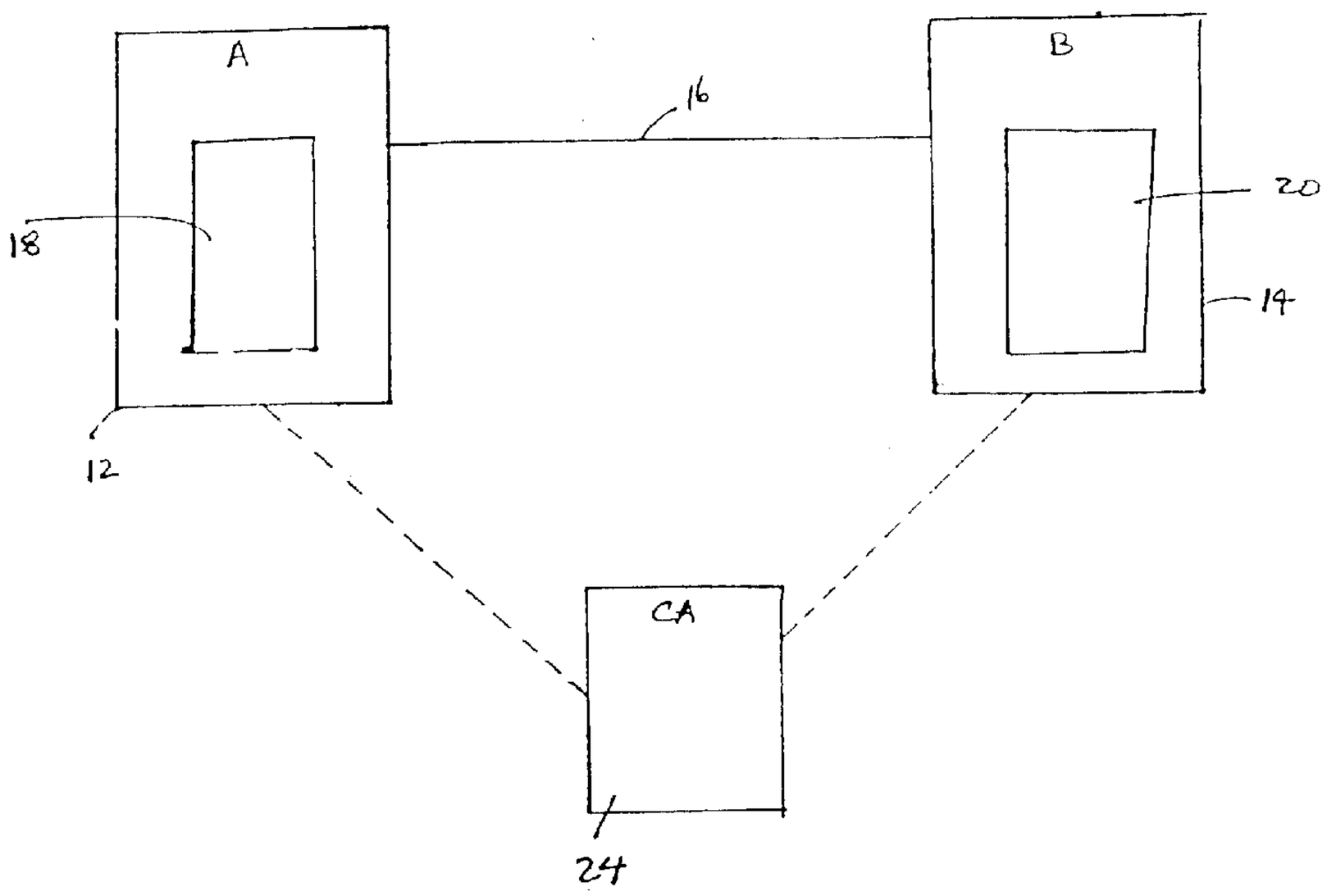
(12) (19) (CA) **Demande-Application**

(21) (A1) **2,306,468**
(86) 1998/10/16
(87) 1999/05/14

(72) VANSTONE, SCOTT A., CA
(72) JOHNSON, DONALD B., US
(71) CERTICOM CORP., CA

(51) Int.Cl.[6] H04L 9/32
(30) 1997/10/31 (08-962441) US

(54) **VERIFICATION DE SIGNATURE POUR SYSTEMES ELGAMAL**
(54) **SIGNATURE VERIFICATION FOR ELGAMAL SCHEMES**



(57) L'invention concerne un protocole de vérification de signature pour systèmes de signature de type ElGamal. Ce système de vérification de signature numérique permet au signataire du message de vérifier la signature numérique sans utiliser la clé publique. En règle générale, le système informatique du signataire est muni d'une clé secrète d et d'une clé publique y provenant d'un élément g et de la clé secrète d. Le procédé consiste

(57) A signature verification protocol is provided for ElGamal-like signature schemes. The digital signature verification scheme allows the signor of the message to verify the digital signature without using the public key. Generally the signors computer system has a private key d and a public key y derived from an element g and the private key d. The method comprises the steps of in the computer system signing a message m by generating a

OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(21) (A1) **2,306,468**
(86) 1998/10/16
(87) 1999/05/14

à signer un message m dans le système informatique par génération d'un premier élément de signature combinant l'élément g et le paramètre de signature k suivant une première fonction mathématique, et par génération d'un deuxième élément de signature par combinaison mathématique du premier élément de signature et de la clé secrète d, du message m et du paramètre de signature k. Le signataire vérifie la signature, d'une part en récupérant une valeur k à partir des éléments de signature sans utiliser la clé publique y, d'autre part en utilisant la valeur k' récupérée dans la première fonction mathématique pour générer une valeur r' permettant de vérifier que les paramètres de signature k et k' sont équivalents, et donc de vérifier la signature. La vérification de la signature est applicable aux signatures de type ElGamal et fonctionne dans n'importe quel groupe et, en particulier, dans des groupes de courbe elliptique. La méthode de vérification de la signature convient particulièrement pour des dispositifs ayant une puissance de calcul limitée, tels que les cartes dites "intelligentes", ou dans les cas où le signataire doit effectuer un grand nombre vérification.

first signature component by combining the element g, the signature parameter k according to a first mathematical function and generating a second signature component by mathematically combining the first signature component with the private key d, the message m and the signature parameter k, and the signor verifying the signature by recovering a value k from the signature components without using the public key y and utilizing the recovered value k' in the first mathematical function to derive a value r' in order to verify the signature parameter k and k' are equivalent, thereby verifying the signature. This signature verification applies to all ElGamal-type signatures and works in any group and in particular elliptic curve groups. The signature verification method is of particular use in devices having limited computational power such as 'smart cards' or where a large number of verifications are to be performed by the signor.

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04L 9/32

A1

(11) International Publication Number: WO 99/23781

(43) International Publication Date: 14 May 1999 (14.05.99)

(21) International Application Number: PCT/CA98/01018

(22) International Filing Date: 2 November 1998 (02.11.98)

(30) Priority Data:
08/962,441 31 October 1997 (31.10.97) US

(71) Applicant *(for all designated States except US)*: CERTICOM CORP. [CA/CA]; Suite 103, 200 Matheson Boulevard West, Mississauga, Ontario L5R 3L7 (CA).

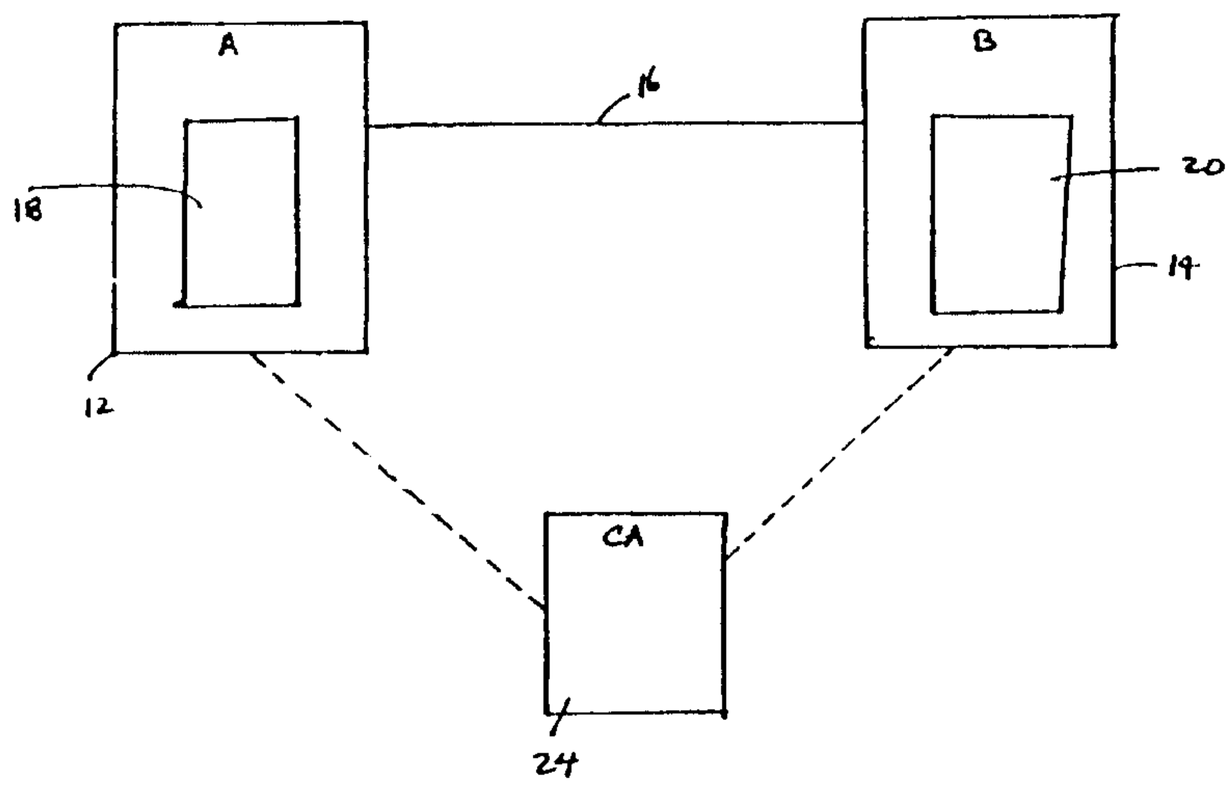
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: JOHNSON, Donald, B. [US/US]; 7684 Knightshays Drive, Manassas, VA 20111 (US). VANSTONE, Scott, A. [CA/CA]; 539 Sandbrook Court, Waterloo, Ontario N2T 2H4 (CA).

(74) Agents: CHARI, Santosh, K. et al.; Orange Chari Pillay, Suite 3600, P.O. Box 190, Toronto Dominion Bank Tower, Toronto-Dominion Centre, Toronto, Ontario M5K 1H6 (CA).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: SIGNATURE VERIFICATION FOR ELGAMAL SCHEMES



(57) Abstract

A signature verification protocol is provided for ElGamal-like signature schemes. The digital signature verification scheme allows the signor of the message to verify the digital signature without using the public key. Generally the signors computer system has a private key d and a public key y derived from an element g and the private key d. The method comprises the steps of in the computer system signing a message m by generating a first signature component by combining the element g, the signature parameter k according to a first mathematical function and generating a second signature component by mathematically combining the first signature component with the private key d, the message m and the signature parameter k, and the signor verifying the signature by recovering a value k from the signature components without using the public key y and utilizing the recovered value k' in the first mathematical function to derive a value r' in order to verify the signature parameter k and k' are equivalent, thereby verifying the signature. This signature verification applies to all ElGamal-type signatures and works in any group and in particular elliptic curve groups. The signature verification method is of particular use in devices having limited computational power such as 'smart cards' or where a large number of verifications are to be performed by the signor.

Signature Verification For ElGamal Schemes

This invention relates to a method of accelerating digital signature verification operations performed in a finite field and in particular to a method for use with processors having limited computing power.

Background of the Invention

One of the functions performed by a cryptosystem is the computation of digital signatures that are used to confirm that a particular party has originated a message and that the contents have not been altered during transmission. A widely used set of signature protocols utilizes the ElGamal public key signature scheme that signs a message with the sender's private key. The recipient may then recover the message with the sender's public key. The ElGamal scheme gets its security from calculating discrete logarithms in a finite field. Furthermore, the ElGamal-type signatures work in any group and in particular elliptic curve groups. For example given the elliptic curve group $E(F_q)$ then for $P \in E(F_q)$ and $Q = aP$ the discrete logarithm problem reduces to finding the integer a. Thus these cryptosystems can be computationally intensive.

Various protocols exist for implementing such a scheme. For example, a digital signature algorithm DSA is a variant of the ElGamal scheme. In these schemes a pair of correspondent entities A and B each create a public key and a corresponding private key. The entity A signs a message *m* of arbitrary length. The entity B can verify this signature by using A's public key. In each case however, both the sender, entity A, and the recipient, entity B, are required to perform a computationally intensive operations to generate and verify the signature respectively. Where either party has adequate computing power this does not present a particular problem but where one or both the parties have limited computing power, such as in a "Smart card " application, the computations may introduce delays in the signature and verification process.

There are also circumstances where the signor is required to verify its own signature. For example in a public key cryptographic system, the distribution of keys is easier than that of a symmetric key system. However, the integrity of public keys is critical. Thus the entities in such a system may use a trusted third party to certify the public key of each entity. This third party may be a certifying authority (CA), that has a

private signing algorithm S_T and a verification algorithm V_T assumed to be known by all entities. In its simplest form the CA provides a certificate binding the identity of an entity to its public key. This may consist of signing a message consisting of an identifier and the entity's authenticated public key. From time to time however the CA may wish to authenticate or verify its own certificates. Thus in these instances it would be convenient to implement an improved signature verification algorithm to speed up this verification process.

Summary of the Invention

It is therefore an object of the present invention to provide a method of fast signature verification.

This invention seeks to provide a digital signature verification method, which may be implemented relatively efficiently by a signor on a processor with limited processing capability, such as a smart card or where frequent verifications are performed such as a certification authority.

In accordance with this invention there is provided a method of verifying a digital signature generated by a signor in a computer system, the signor having a private key d and a public key y, derived from an element g and the private key d the method comprising the steps of:

a) in the computer system signing a message *m* by;

b) generating a first signature component by combining at least the element *g* and the signature parameter *k* according to a first mathematical function;

c) generating a second signature component by mathematically combining the first signature component with the private key d, the message *m* and the signature parameter *k*; and

the signor verifying the signature by:

d) recovering a value *k'* from the signature without using the public key y, and ;

e) utilizing the recovered value *k'* in the first mathematical function to derive a value *r'* to verify the signature parameter *k* and *k'* are equivalent.

Brief Description of the Drawings

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 communication system; and

Figure 2 is a flow chart showing a signature algorithm according to the present invention.

Detailed Description of a Preferred Embodiment

For the sake of convenience in the following discussion we use the multiplicative notation, although ElGamal-type signatures work in any group and in particular in elliptic curve groups.

Referring therefore to Figure 1, a data communication system 10 includes a pair of correspondents, designated as a sender A(12), and a recipient B(14), who are connected by a communication channel 16. Each of the correspondents A and B (12,14) includes an encryption unit 18,20 respectively that may process digital information and prepare it for transmission through the channel 16 as will be described below.

In accordance with a general embodiment, the sender A assembles a data string, which includes amongst others the public key y of the sender, a message m, the sender's short-term public key k and signature S of the sender A. When assembled the data string is sent over the channel 16 to the intended recipient B, who then verifies the signature using A's public key. This public key information may be obtained from a certification authority (CA) 24 or sometimes is set with the message. The CA generally has a public file of the entity's public key and identification.

For key generation in the ElGamal signature scheme, each correspondent A and B creates a public key and corresponding private key. In order to set up the scheme, the entities A and B select primes p and q such that q divides p-1. A g is selected such that it is an element of order q in F_p and the group used is $\{g^0, g^1, g^2, \ldots g^{q-1}\}$.

The digital signature algorithm (DSA) which is a special case of the ElGamal scheme, key generation is performed by selecting a random integer d in the interval [1, q-1] and computing $y=g^d \bmod p$. In the DSA the public key information is (p, q, g, y) and the private key is d, while in the general ElGamal scheme the public key information is (p, g, y) and the private key is d.

We consider firstly a signature scheme such as the DSA in which the signature components r and s are given by:

$$r = (g^k \bmod p) \bmod q \text{ ; and}$$

$$s = k^{-1}(h(m) + dr) \bmod q$$

where typically:

d is a random integer, the signors private key and is typically 160-bits;

p is typically a 1024-bit prime;

q is a 160-bit prime where q divides p-1;

g is the generator such that $y = g^d \bmod p$;

$h(m)$ is typically a SHA-1 hash of the message m;

k is a randomly chosen 160-bit value for each signature; and

the signature for m is the pair (r, s).

Normally to verify A's signature (r, s) on the message m, the recipient B should obtain A's authentic public key (p, q, g, y), and verify that $0< r < q$ and $0 < s < q$. Next the values

$w = s^{-1} \bmod q$ and $h(m)$ are computed. This is followed by computing $u_1 = w\ h(m) \bmod q$ and $u_2 = r\ w \bmod q$ and $v = (g^{u1}\ y^{u2} \bmod p) \bmod q$. The signature is accepted if and only if $v = r$. It may be seen therefore that in some cases if the owner of the signature wants to verify its own signature at a later stage it may be time consuming to retrieve the public key information and perform the steps above, particularly since the signor is verifying its own signature.

Thus, in order to implement fast signature verification using the private key d, it may be seen that the verifier, in this case the original signor, has knowledge of p, q, g, y, $h(m)$, r and s. Thus the verifier need only recover the (secret) per signature value k used and verify this value of k thus obtained in order to verify the signature. The verifier thus calculates $z = (h(m) + dr) \bmod q$. The value z^{-1} is calculated by inverting z mod q. Next calculate $k'^{-1} = s(z^{-1}) \bmod q$ and calculate k' by inverting $k'^{-1} \bmod q$. The verifier then evaluates $r = g^{k'} \bmod p \bmod q$ this verifies $k = k'$. Thus it may be seen that this verification step uses d not y and many of the calculations above can be sped up using pre-computed tables.

Next we consider an alternate ElGamal signature method shown in figure 2 as having signature components (s, e) where:

$r = g^k \bmod p$;

$e = h(m\|r)$ where || indicates concantenation; and

$s = (de + k) \bmod p$

The signature components are s and e where p is a large public prime, g is a public generator, m is a message, h is a hash function, d is a private key, $y = g^d \bmod p$ is a public key and k is a secret random integer.

In fast signature verification using the private key d we once again assume knowledge of p, g, y, h, m, r, e and d. Thus the verifier need only recover the k value used and verify k in order to verify the signature. Thus the verifier calculates $k' = (s - de) \bmod p$, $r' = g^{k'} \bmod p$ and e' = h(m||r'). If e = e' this verifies k = k'.

Thus it may be seen that an advantage of the present invention is where a signor signs data which for example may reside on the signors computer. This can be later verified without use of the correponding public key, instead the signor can use its private key to verify the data. This is also very useful for some applications with limited computational power such as smartcards.

In a data communication system that includes a certifying authority, the certifying authority (CA) or key distribution centre would sign data frequently before it is installed into the various communications systems and then could verify the signatures later. Thus the CA does not require the public key information to verify the signatures but simply uses the private key to verify, as all the other parameters are stored within the secure boundary of the signor.

A further application is in the verification of software such in pay-per-use software applications.

While the invention has been described in connection with specific embodiments thereof and in specific uses, various modifications thereof will occur to those skilled in the art without departing from the spirit of the invention as set forth in the appended claims. For example, in the above description of preferred embodiments, use is made of multiplicative notation however the method of the subject invention may be equally well described utilizing additive notation. It is well known for example that the elliptic curve

algorithm equivalent of the DSA, i.e. ECDSA is the elliptic curve analog of a discrete logorithm algorithm that is usually described in a setting of F_p^*, the multiplicative group of the integers modulo a prime. There is correspondence between the elements and operations of the group F_p^* and the elliptic curve group $E(F_q)$. Furthermore, this signature technique is equally well applicable to functions performed in a field defined over F_{2^n}.

The present invention is thus generally concerned with an encryption method and system and particularly an elliptic curve encryption method and system in which finite field elements is multiplied in a processor efficient manner. The encryption system can comprise any suitable processor unit such as a suitably programmed general-purpose computer.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method of verifying a digital signature generated by a signor in a computer system, said signor having a private key d and a public key y, derived from an element g and said private key d said method comprising the steps of:

a) in said computer system signing a message *m* by;

b) generating a first signature component by combining at least said element *g* and said signature parameter *k* according to a first mathematical function;

c) generating a second signature component by mathematically combining said first signature component with said private key *d*, said message *m* and said signature parameter *k*; and

said signor verifying said signature by:

d) recovering a value *k'* from said signature without using said public key *y*, and;

e) utilizing said recovered value *k'* in said first mathematical function to derive a value *r'* to verify said signature parameter *k* and *k'* are equivalent.

2. A method as defined in claim 1, wherein g is an element of order q in a field F_p^*.

3. A method as defined in claim 1, wherein g is a point of prime order n in $E(F_q)$, such that E is an elliptic curve defined over the field F_q.

4. A method as defined in claim 1, wherein said element g is a point on an elliptic curve over a finite field F_{q^m}.

5. A method as defined in claim 1, said signature parameter k being a randomly selected integer in the interval [1, q-1], and said first signature component having a form defined by $r=g^k \bmod p \bmod q$, wherein *p* and *q* are primes such that *q* divides *p-1*.

6. A method as defined in claim 5, including calculating a value $e = h(m)$ wherein h is a hash function, and wherein said second signature component $s = k^{-1}(e + dr) \bmod q$.

7. A method as defined in claim 6, said step of recovering said value k' including:

(a) calculating a value $z = (h(m) + dr) \bmod q$;

(b) calculating z^{-1} inverting z mod q;

(c) calculating $k'^{-1} = s(z^{-1}) \bmod q$; and

(d) calculating k' by inverting $k^{-1} \bmod q$.

8. A method as defined in claim 7, said step of verifying k including the steps of calculating $r' = g^{k'} \bmod p \bmod q$ and comparing r' to r in order to verify $k = k'$.

9. A method as defined in claim 9, including utilizing precomputed tables in said calculations.

10. A method as defined in claim 3, said signature parameter k being a statistically unique and unpredictable integer k selected in an interval [2, n-2] and said first signature component having a form defined by $r = x$, mod n wherein n is an n co-ordinate of a private key.

11. A method as defined in claim 10, including calculating a value $e = h(m)$ wherein h is a hash function and said second signature component is given by $s = k^{-1} (e + dr) \bmod n$.

12. A method as defined in claim 11, said recovering said value k' includes:

(a) calculating a value $z = (h(m) + dr) \bmod n$;

(b) calculating z^{-1} by inverting z mod n;

(c) calculating $k'^{-1} = s(z^{-1}) \bmod n$, and

(d) calculating k' by inverting $k^{-1} \bmod n$.

13. A method as defined in claim 12, said step of verifying k including the steps of calculating $r' = g^{k'} \bmod n$ and comparing r' to r in order to verify $k = k'$.

14. A method as defined in claim 2, said signature parameter k being a randomly selected integer in an interval [1, p-1], and said first signature component having a form

defined by $e = h(m||r)$ wherein $r = g^k \bmod p$, h is a hash function and || denotes concantenation.

15. A method as defined in claim 14, said second signature component being defined by $s = (de + k) \bmod p$.

16. A method as defined in claim 15, said step of recovering said value k' includes:

(a) calculating a value $k' = (s\text{-}de) \bmod p$;

(b) calculating a value $r' = g^k \bmod p$;

(c) calculating a value $e' = h(m||r')$; and

(d) comparing said value e' to e in order to verify $k' = k$.

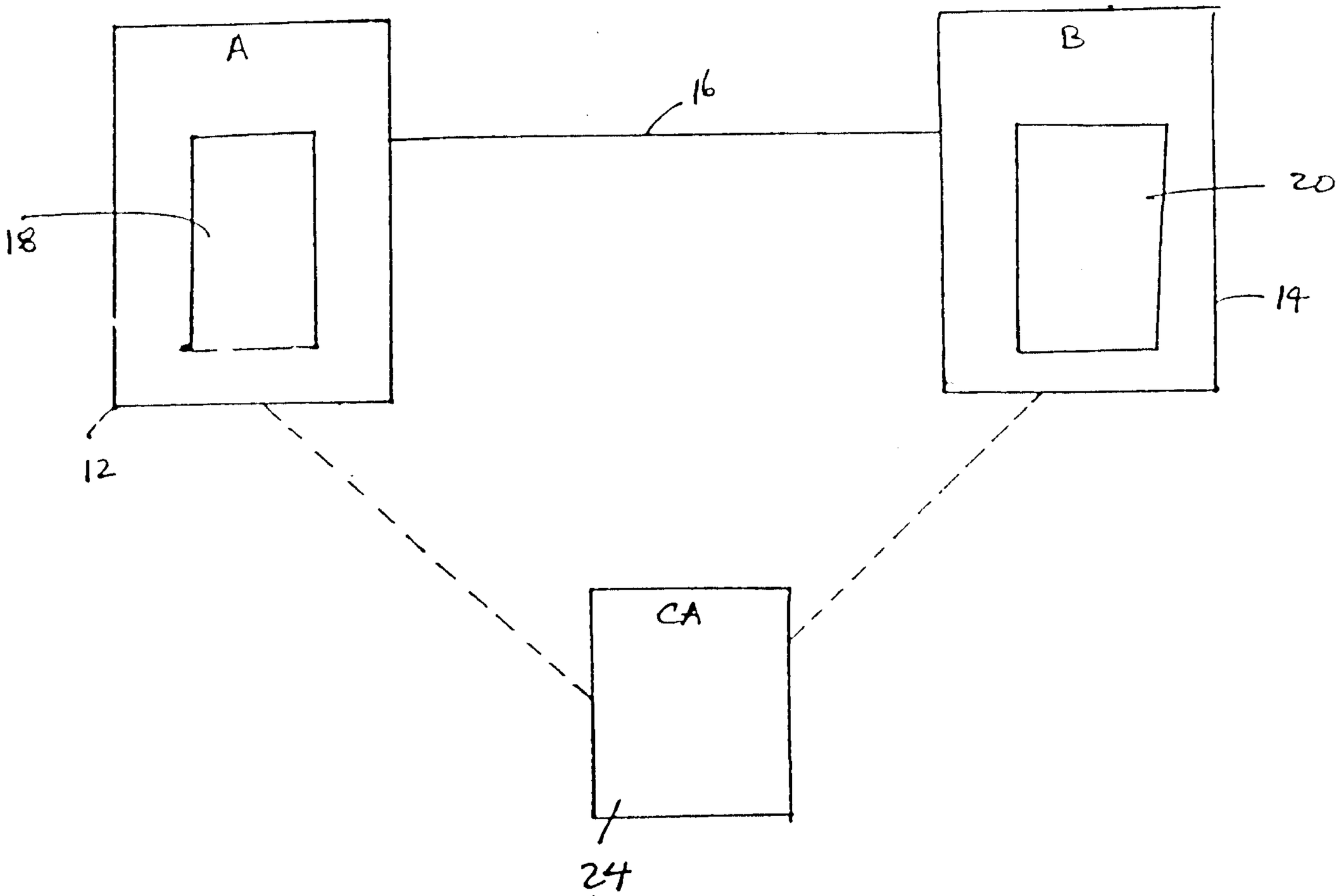


FIGURE 1

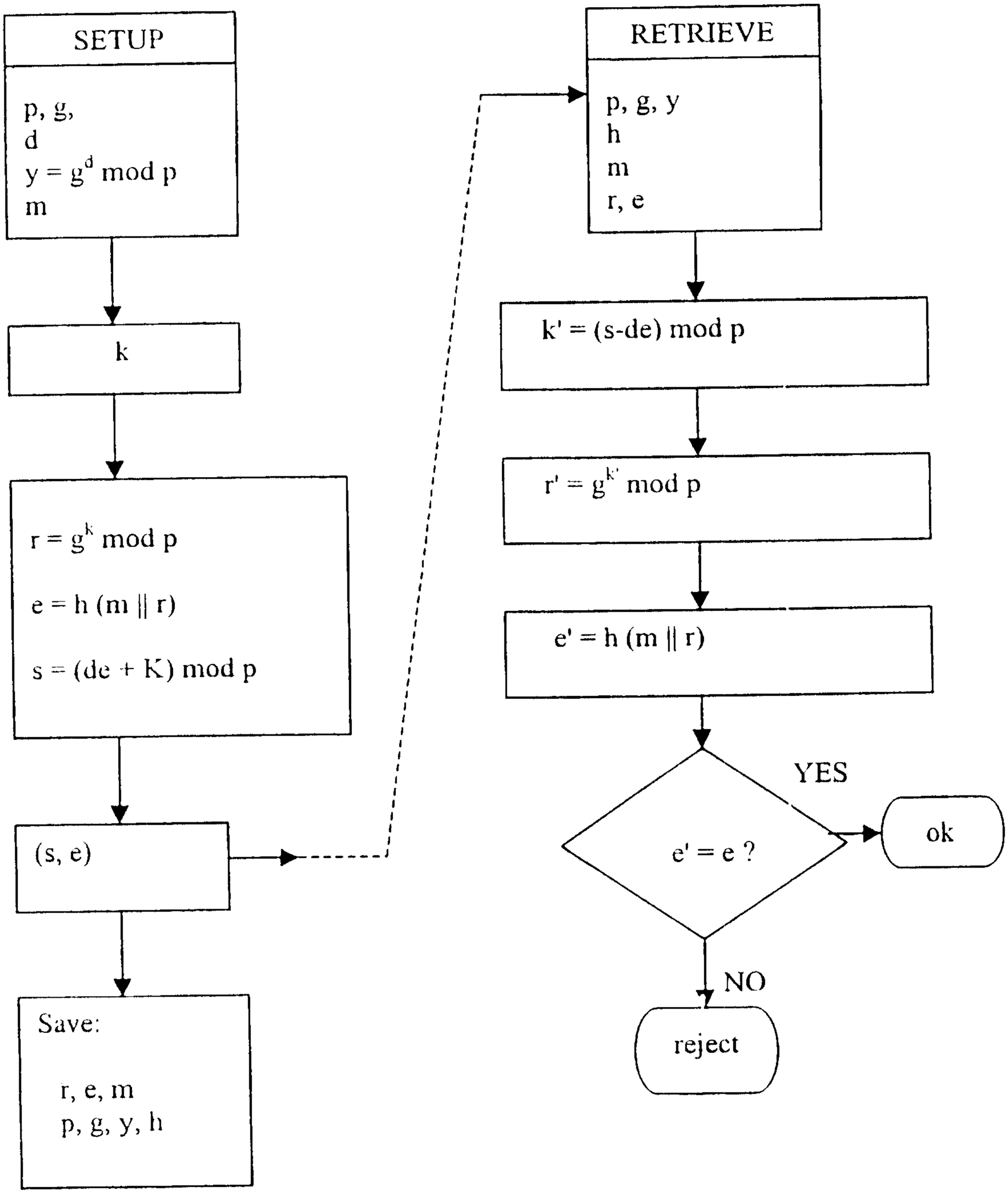

SIGN
VERIFY
SETUP
p, g,
d
y = g^d mod p
m
k
r = g^k mod p
e = h (m || r)
s = (de + K) mod p
(s, e)
Save:
r, e, m
p, g, y, h
RETRIEVE
p, g, y
h
m
r, e
k' = (s-de) mod p
r' = g^k' mod p
e' = h (m || r)
e' = e ?
YES
ok
NO
reject

Figure 2