CRYPTOGRAPHIC METHOD AND SYSTEM

Inventors: Liquin Chen, Bristol (GB); Keith Alexander Harrison, Woodcroft Chepstow (GB)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400 (US)

Publication Classification

Int. Cl. 7 ................................. H04L 9/00; H04L 9/32

U.S. Cl. .......................................... 713/161

ABSTRACT

First data to be sent by a first party to a second party is encrypted using public data of a trusted party and an encryption key string formed using at least a hash value generated by hashing at least one condition that typically serves as an identifier of an intended recipient of the first data. The encrypted first data is provided to a data recipient who requests a decryption key from the trusted party. The trusted party is responsible for verifying that the recipient meets the specified conditions before providing the decryption key. A valid decryption key is only provided if the correct conditions have been supplied to the trusted party.
Figure 1

(Prior Art)
<table>
<thead>
<tr>
<th>Identifier-Based Method</th>
<th>Quadratic Residuosity Based</th>
<th>RSA Based</th>
<th>EKS</th>
<th>EKS</th>
<th>EKS</th>
<th>EKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Type/Description</td>
<td>( p: G_1 \times G_1 \rightarrow G_2 )</td>
<td>( p ) is in ( G_1 )</td>
<td>Modulus ( n )</td>
<td>Modulus ( N )</td>
<td>Modulus ( n )</td>
<td>Modulus ( n )</td>
</tr>
<tr>
<td>Equation</td>
<td>( c = #(EKS) \mod n )</td>
<td>( K = #(EKS) )</td>
<td>( Q_{DP} = H_s(\text{Map-to-point hash}) )</td>
<td>( H_{F}(0,1) )</td>
<td>( m^e \mod n )</td>
<td>( m^e \mod n )</td>
</tr>
<tr>
<td>For each bit</td>
<td>( s_1 = (t_1 + K/K) \mod n )</td>
<td>( s_1 = (t_1 - K/K) \mod n )</td>
<td>For each bit</td>
<td>( s_1 ), ( s_2 )</td>
<td>For each bit</td>
<td>( V, U = rP )</td>
</tr>
</tbody>
</table>

(Prior art)

Figure 2

Encryption Process Output

EKS Conversion

Message Encryption

Encryption Parameters

TA Public Data

Encryption Key String (EKS)
BEGIN

PARSE $K_{ENC}$

CHECK PROVIDER 20

Fail

Pass

CHECK Condition1

Fail

Pass

COMPUTE $K_{20_{private}}$

DECRYPT

$E(K_{20_{public}}, N; (H(msg) :: nonce :: Condition 2))$

CHECK Condition2

Fail

Pass

COMPUTE

$H(K_{20_{private}} :: H(msg) :: nonce :: Condition 1 :: Condition 2)$

COMPARE Received & Computed Hashes

Fail

Pass

GENERATE $K_{DEC}$

RETURN $K_{DEC}, H(msg)$

RETURN Negative Response

END
CRYPTOGRAPHIC METHOD AND SYSTEM

FIELD OF THE INVENTION

[0001] The present invention relates to cryptographic methods and systems and, in particular, to data transfer methods and systems that use Identifier-Based Encryption.

BACKGROUND OF THE INVENTION

[0002] Identifier-Based Encryption (IBE) is an emerging cryptographic schema. In this schema (see FIG. 1 of the accompanying drawings), a data provider 10 encrypts payload data 13 using both an encryption key string 14, and public data 15 provided by a trusted authority 12. This public data 15 is derived by the trusted authority 12 using private data 17 and a one-way function 18. The data provider 10 then provides the encrypted payload data <13> to a recipient 11 who decrypts it, or has it decrypted, using a decryption key computed by the trusted authority 12 based on the encryption key string and its own private data.

[0003] A feature of identifier-based encryption is that because the decryption key is generated from the encryption key string, its generation can be postponed until needed for decryption.

[0004] Another feature of identifier-based encryption is that the encryption key string is cryptographically unconstrained and can be any kind of string, that is, any ordered series of bits whether derived from a character string, a serialized image bit map, a digitized sound signal, or any other data source. The string may be made up of more than one component and may be formed by data already subject to upstream processing. In order to avoid cryptographic attacks based on judicious selection of a key string to reveal information about the encryption process, as part of the encryption process the encryption key string is passed through a one-way function (typically some sort of hash function) thereby making it impossible to choose a cryptographically-prejudicial encryption key string. In applications where defence against such attacks is not important, it would be possible to omit this processing of the string.

[0005] Frequently, the encryption key string serves to “identify” the intended message recipient and this has given rise to the use of the label “identifier-based” or “identity-based” generally for cryptographic methods of the type under discussion. However, depending on the application to which such a cryptographic method is put, the string may serve a different purpose to that of identifying the intended recipient and, indeed, may be an arbitrary string having no other purpose than to form the basis of the cryptographic processes. Accordingly, the use of the term “identifier-based” or “IBE” herein in relation to cryptographic methods and systems is to be understood simply as implying that the methods and systems are based on the use of a cryptographically unconstrained string whether or not the string serves to identify the intended recipient. Generally, in the present specification, the term “encryption key string” or “EKS” is used rather than “identity string” or “identifier string”; the term “encryption key string” is also used in the shortened form “encryption key” for reasons of brevity.

[0006] A number of IBE algorithms are known and FIG. 2 indicates, for three such algorithms, the following features, namely:

[0007] the form of the encryption parameters used, that is, the encryption key string and the public data of the trusted authority (TA);

[0008] the conversion process applied to the encryption key string to prevent attacks based on judicious selection of this string;

[0009] the primary encryption computation effected;

[0010] the form of the encrypted output.

[0011] The three prior art IBE algorithms to which FIG. 2 relates are:


[0013] Bilinear Mappings p using, for example, a Tate pairing t or Weil pairing ĉ. Thus, for the Weil pairing:

[0014] ĉ: G1×G1→G2

[0015] where G1 and G2 denote two algebraic groups of prime order q and G1 is a subgroup of a multiplicative group of a finite field. The Tate pairing can be similarly expressed though it is possible for it to be of asymmetric form:

[0016] t: G1×Gq→Gq

[0017] where Gq is a further algebraic group the elements of which are not restricted to being of order q. Generally, the elements of the groups G1 and G2 are points on an elliptic curve though this is not necessarily the case. A description of this form of IBE method, using Weil pairings is given in the paper: D. Boneh, M. Franklin—“Identity-based Encryption from the Weil Pairing” in Advances in Cryptology—CRYPTO 2001, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[0018] RSA-Based methods The RSA public key cryptographic method is well known and in its basic form is a two-party method in which a first party generates a public/private key pair and a second party uses the first party’s public key to encrypt messages for sending to the first party, the latter then using its private key to decrypt the messages. A variant of the basic RSA method, known as “mediated RSA”, requires the involvement of a security mediator in order for a message recipient to be able to decrypt an encrypted message. An IBE method based on mediated RSA is described in the paper “Identity-based encryption using mediated RSA”, D. Boneh, X. Ding and G. Tsudik, 3rd Workshop on Information Security Application, Jeju Island, Korea, Aug., 2002.

[0019] A more detailed description of the QR method is given below with reference to the entities depicted in FIG. 1 and using the same notation as given for this method in FIG. 2. In the QR method, the trust authority’s public data 15 comprises a value N that is a product of two random prime numbers p and q, where the values of p and q are the private data 17 of the trust authority 12. The values of p and
q should ideally be in the range of 2^{511} and 2^{512} and should both satisfy the equation: p, q=\text{mod} 4. However, p and q must not have the same value. Also provided is a hash function \# which when applied to a string returns a value in the range 0 to N-1.

[0020] Each bit of the user’s payload data 13 is then encrypted as follows:

[0021] The data provider 10 generates random numbers \( t_x \) (where \( t_x \) is an integer in the range \([0; 2^x] \)) until a value of \( t_x \) is found that satisfies the equation \( \text{jacobi}(t_x, N)=m' \), where \( m' \) has a value of \( -1 \) or \( 1 \) depending on whether the corresponding bit of the user’s data is 0 or 1 respectively. (As is well known, the jacobi function is such that where \( x^2=\# \mod N \), the jacobi \((x, N)=1 \) if \( x \) does not exist, and \( -1 \) if \( x \) does exist). The data provider 10 then computes the value:

\[ s_x=m(t_x+Kt_x \mod N) \]

where: \( s_x \) corresponds to the encrypted value of the bit \( m \) concerned, and \( K=\# \text{(encryption key string)} \).

[0022] Since \( K \) may be non-square, the data provider additionally generates additional random numbers \( t \) (integers in the range \([0, 2^x] \)) until one is found that satisfies the equation \( \text{jacobi}(t_x, N)=m' \). The data provider 10 then computes the value:

\[ s_x=m(t_x+Kt_x \mod N) \]

[0023] as the encrypted value of the bit \( m \) concerned.

[0024] The encrypted values \( s_x \) and \( s \) for each bit \( m' \) of the user’s data are then made available to the intended recipient 11, for example, via e-mail or by being placed in an electronic public area; the identity of the trust authority 12 and the encryption key string 14 will generally also be made available in the same way.

[0025] The encryption key string 14 is passed to the trust authority 12 by any suitable means; for example, the recipient 11 may pass it to the trust authority or some other route is used—indeed, the trust authority may have initially provided the encryption key string. The trust authority 12 determines the associated private key \( B \) by solving the equation:

\[ B=K \mod N \text{ ("positive" solution)} \]

[0027] If a value of \( B \) does not exist, then there is a value of \( B \) that is satisfied by the equation:

\[ B=\bar{K} \mod N \text{ ("negative" solution)} \]

[0028] As \( N \) is a product of two prime numbers \( p \), \( q \) it would be extremely difficult for any one to calculate the decryption key \( B \) with only knowledge of the encryption key string and \( N \). However, as the trust authority 12 has knowledge of \( p \) and \( q \) (i.e., two prime numbers) it is relatively straightforward for the trust authority 12 to calculate \( B \).

[0029] Any change to the encryption key string 14 will result in a decryption key 16 that will not decrypt the payload data 13 correctly. Therefore, the intended recipient 11 cannot alter the encryption key string before supplying it to the trust authority 12.

[0030] The trust authority 12 sends the decryption key to the data recipient 11 along with an indication of whether this is the “positive” or “negative” solution for \( B \).

[0031] If the “positive” solution for the decryption key has been provided, the recipient 11 can now recover each bit \( m' \) of the payload data 13 using:

\[ m'=\text{jacobi}(s_x+2B, N) \]

[0032] If the “negative” solution for the decryption key \( B \) has been provided, the recipient 11 recovers each bit \( m' \) using:

\[ m'=\text{jacobi}(s_x+2B, N) \]

[0033] Returning now to a general consideration of IBE encryption, one application is to enable the data provider 10 to provide encrypted payload data over an unprotected communications path for receipt and decryption by a recipient 11 where certain conditions are met, namely condition 1 and condition 2. To ensure that the conditions are met before a recipient can read the payload data 13, the conditions are placed in the IBE encryption key string 14 and sent along with the encrypted payload data. Upon receipt, the data receiver 11 passes the encryption key string to the trusted authority 12 with a request for the corresponding IBE decryption key 16. The trusted authority 12 only provides the decryption key (over a secure channel) if satisfied that the conditions 1 and 2 included in the encryption key are met. Typically, the conditions serve to identify the intended recipient in some manner and can therefore be considered as the recipient’s identifiers by the requesting data receiver 11; however, other conditions are also possible such as a time or date condition.

[0034] The foregoing example exhibits a number of potential drawbacks. More particularly, the conditions are transmitted in clear which may be undesirable particularly where the conditions are identifiers of the intended data receiver. In certain circumstances, it is better for the conditions not to be included in the encryption key but to be provide by the data receiver to the trusted authority. However, this runs the risk of the data receiver modifying the conditions.

[0035] It is an object of the present invention to provide improved cryptographic methods and apparatuses.

SUMMARY OF THE INVENTION

[0036] According to one aspect of the present invention, there is provided a data-transfer method comprising:

[0037] (a) encrypting first data at a first party using as encryption parameters both public data of a trusted party and an encryption key string comprising a hash value generated by hashing second data comprising at least one condition;

[0038] (b) providing the encrypted first data to a second party, and the encryption key string and said at least one condition to a trusted party;

[0039] (c) at the trusted party effecting both a first check that the hash value in the encryption key string matches the result of a hash based on the said at least one condition provided in step (b), and a second check that the or each said at least one condition is met; and

[0040] (d) providing a decryption key to the second party only if said checks are satisfactory, this decryption key being generated at the trusted party using the encryption key string and private data related to said public data.
The present invention also encompasses a system for carrying out the data transfer method of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Embellishments of the invention will now be described, by way of non-limiting example, with reference to the accompanying diagrammatic drawings, in which:

FIG. 1 is a diagram illustrating the operation of a prior art encryption schema known as Identifier-Based Encryption;

FIG. 2 is a diagram illustrating how certain IBE operations are implemented by three different prior art IBE methods;

FIG. 3 is a diagram of a system embodying the present invention; and

FIG. 4 is a flow chart of a process carried out by a trusted authority of the FIG. 3 system.

BEST MODE OF CARRYING OUT THE INVENTION

FIG. 3 illustrates a system embodying the present invention, the system comprising a first computing entity 20 associated with a data provider party; a second computing entity 30 associated with a data receiver party; and a third computing entity 40 associated with a trusted authority. The computing entities 20, 30 and 40 are typically based around general-purpose processors executing stored programs but may include dedicated cryptographic hardware modules. The computing entities 20, 30 and 40 inter-communicate as needed (see arrows 50–53) via, for example, the internet or other network, though it is also possible that at least some of the entities actually reside on the same computing platform.

In the following, references to the data provider, data receiver and the trusted authority are generally used interchangeably with references to their respective computing entities 20, 30 and 40.

In functional terms, the data-provider entity 20 comprises a communications module 24 for communicating with the entities 30 and 40, a control module 21 for controlling the general operation of the entity 20, and a cryptographic module 22 for executing certain cryptographic functions comprising a hash function and an IBE encryption function.

The data-receiver entity 30 comprises a communications module 34 for communicating with the entities 20 and 40, a control module 31 for controlling the general operation of the entity 30, and a cryptographic module 32 for executing certain cryptographic functions comprising a hash function (the same as that used by the entity 20) and an IBE decryption function.

The trusted authority entity 40 comprises a communications module 48 for communicating with the entities 20 and 30, a control module 41 for controlling the general operation of the entity 40, a cryptographic module 42 for executing certain cryptographic functions, a condition checking module 43, and a user registration module 44. The cryptographic module 42 is arranged to implement both a hash function (the same as that used by the entity 20) and an IBE decryption function; in addition, the module 42 includes a unit 45 for generating an IBE decryption key using both a supplied encryption key string and private data securely held in local store 46.

The system employs Identifier-Based Encryption with the computing entities 20, 30 and 40 having, in respect of IBE encryption/decryption, the roles of the data provider 10, data recipient 11 and trusted authority 12 of the FIG. 1 IBE arrangement. The IBE algorithm used is, for example, the QR algorithm described above with respect to FIG. 1 with the private data held in store 46 being random prime numbers p,q and the corresponding public data being number N.

Consider the situation where the data provider 20 wishes to encrypt message data (“msg”) for sending over an unprotected communications path for receipt and decryption by a recipient that meets certain conditions, namely Condition 1 and Condition 2. These Conditions 1 and 2 are unknown to the data receiver 30 and the data provider 20 wishes to keep Condition 2 confidential from the data receiver 30.

It is assumed that the data provider 20 has previously registered with the trusted authority 40 and obtained (see arrow 50) a public/private key pair K_{public}/K_{private} where K_{public} is simply a public identifier of provider 20 (such as a name) and K_{private} is the IBE decryption key formed by the trusted authority using the K_{public} as an IBE encryption key and its private data p,q. The user registration module 44 is responsible at the time of registration for ensuring that the public key K_{public} is a correct identifier of the data provider 20; the module 44 is also arranged to keep a record of currently valid registered users.

To encrypt the message data msg, the data provider 20 first forms an IBE encryption key string K_{ENC} comprising:

K_{public}, \( H(k_{msg}) \cdot nonce \cdot \text{Condition 1} \cdot \text{Condition 2} \)

where:

\( \cdot \) means concatenation,

\( H(x) \) means the hash of data x using any suitable hash function such as SHA1,

\( E(k,\text{msg}) \) means the IBE encryption of data y using encryption key string k and the public data n of a trusted authority, and

\( \text{a nonce} \) is a one-time use random number selected by the data provider 20 and provided for freshness.

The process of forming the encryption key string K_{ENC} is carried out by the cryptographic module 22 under the direction of the control module 21.

As can be seen, whilst Condition 1 is visible in the encryption key string K_{ENC}, the Condition 2 only appears in encrypted form. Furthermore, the encryption key string K_{ENC} includes a hash of the message data msg, and a hashed
quantity that includes both the data-provider’s private key $K_{20private}$ and the message data hash; as will be seen herinafter, this enables the trusted authority 40 to check the origin of the encryption key string $K_{ENC}$ and the integrity of the message hash.

[0067] After the key $K_{ENC}$ has been generated, the control module 21 causes the cryptographic module 22 to use the key and the trusted authority’s public data N to encrypt the message data msg. The encrypted data and the encryption key string $K_{ENC}$ are then made available by the communications module 24 to the data receiver 30 (see arrow 51).

[0068] On receiving the encrypted message data and the encryption key string $K_{ENC}$, the control module 31 of the data receiver 30 may, if it understands the structure of the encryption key string, examine the identity $K_{20Public}$ of the data provider 20 and the unencrypted Condition 1. If the data receiver determines that it wants to read the message data and that it meets Condition 1, or if the data receiver decides to proceed without checking Condition 1 (for example, because it does not know the structure of the encryption key string), the control module 31 causes the encryption key string $K_{ENC}$ to be sent (arrow 52) to the trusted authority 40 with a request for the corresponding decryption key $K_{DEC}$.

[0069] On receipt of the request from the data receiver 30 for the decryption key $K_{DEC}$, the control module 41 of the trusted authority 40 oversees the processing represented by the flow chart of FIG. 4. More particularly, the control module 41 first parses (step 61) the encryption key string $K_{ENC}$ provided with the request into its four constituent components (the four concatenated components listed above)—this typically being done on the basis of predetermined separators inserted between the concatenated components.

[0070] Next, the control module 41 passes the component formed by the data provider’s public key $K_{20Public}$ to the module 44 in order to determine whether the data provider 20 is still a valid registered user of the services of the trusted authority 40 (step 62). If this check fails, a negative response is returned to the requesting data receiver 30 (step 72) and processing terminates; otherwise processing proceeds. In fact, the trusted authority 40 may decide to skip this check and simply proceed directly to the following processing steps.

[0071] The next processing step (step 63) involves the control module 41 passing the Condition 1 component extracted from the encryption key string $K_{ENC}$ to the condition checking module 43 for it to determine whether the data receiver 30 satisfies this condition. Condition checking may involve the consultation of internal and/or external databases and/or the interrogation of the data receiver 30 (for which purpose the latter may be implemented on a trusted computing platform). If this check fails, a negative response is returned to the requesting data receiver 30 (step 72) and processing terminates; otherwise processing proceeds.

[0072] The following step (step 64) involves the control module 41 obtaining the data provider’s private key $K_{20Private}$ and retrieving against the data provider’s public key $K_{20Public}$ (as extracted from the encryption key string $K_{ENC}$), it is simpler to have the key generation unit 45 regenerate the $K_{20Private}$ using the data provider’s public key $K_{20Public}$ and the private data p,q held in storage 46.

[0073] Once the private key $K_{20Private}$ has been obtained, it is used (step 65) to decrypt the encrypted component of the encryption key string $K_{ENC}$ in order to reveal:

[0074] $H(msg)::$ nonce:: Condition 2

[0075] this expression thereafter being separated into its three constituent elements.

[0076] Next, the control module 41 passes the now-decrypted Condition 2 to the condition checking module 43 for it to determine whether the data receiver 30 satisfies this condition (step 66). If this check fails, a negative response is returned to the requesting data receiver 30 (step 72) and processing terminates; otherwise proceeding proceeds.

[0077] Following the successful check of Condition 2, the control module 41 causes the hash:

[0078] $H(K_{20Private}, H(msg))::$ nonce:: Condition 1:: Condition 2

[0079] to be recomputed (step 67) using the key $K_{20Private}$ obtained in step 64, the values of $H(msg)$, the nonce and Condition 2 obtained by the decryption in step 65 of the encrypted data contained in $K_{ENC}$, and the Condition 1 obtained in step 61 from parsing $K_{ENC}$. This re-computed hash value is then compared (step 68) with the corresponding component contained in the encryption $K_{ENC}$. If these hash values are different then clearly something is wrong and a negative response is returned to the requesting data receiver 30 (step 72) and processing terminates.

[0080] However, if the hash values match, the trusted authority 40 accepts that the data provider is the entity associated with the private key $K_{20Private}$ and thus with the public key $K_{20Public}$; the trusted authority also accepts that the message hash $H(msg)$ is reliable. The control module 41 now causes the key generation unit 45 to compute (step 69) the decryption key $K_{DEC}$ using the encryption key string $K_{ENC}$ and the private data p,q. Finally, the trusted authority 40 returns (step 70) the decryption key $K_{DEC}$ together with $H(msg)$ to the data receiver 30 (see arrow 53, FIG. 3).

[0081] It will be appreciated that the ordering of the checking steps 62, 63, 66 and 68 relative to each other and to the other steps of the FIG. 4 is not critical (subject to the items concerned having become available) save that the steps 62, 63, 66 and 68 need to be carried out before the decryption key $K_{DEC}$ is sent to the data receiver in step 70.

[0082] On receiving the decryption key $K_{DEC}$ the data receiver 30 uses it to decrypt the encrypted message data after which it computes the hash of the message and compares it with that received from the trusted authority 40 as a final check on the message integrity. The data receiver 30 now has the integrity-checked decrypted message and can be sure that the trusted authority 40 is happy that the data provider 20 is as identified by the public key $K_{20Public}$ included in clear in the encryption key string $K_{ENC}$.

[0083] In the foregoing process, the only additional burden placed on the data receiver 30 is the message integrity check involving forming a hash of the message and comparing it with the message hash supplied by the trusted authority 40; otherwise, the functioning of the data receiver 30 is exactly as for any basic IBE system with the data
receiver 30 passing the encryption key string $K_{ENC}$ to the trusted authority 40 and receiving back the decryption key $K_{DEK}$. If the data receiver is prepared to pass the encrypted message to the trusted authority, then even the message integrity check can be carried out by the trusted authority.

[0084] The process described above with respect to FIGS. 2 and 3 not only provides the advantages of data-prover authentication carried out by the trusted authority 40, and a check on the integrity of the message hash, but also enables the data provider 20 to include a hidden condition (Condition 2 in the example) in the encryption key string $K_{ENC}$ only visible to the trusted authority 40 and not to the data receiver 30. Whilst the data provider 20 must, of course, know how to form the multi-component encryption key string $K_{ENC}$, the data receiver 30 need know nothing of the structure of this key, it being relieved of this burden by the trusted authority 40. Placing all the provider authentication and message integrity components into the encryption key string $K_{ENC}$ intimately ties these components to the encrypted message data.

[0085] Many variants are possible to the above-described embodiment. For example, instead of the QR IBE method being used for the encrypting and decrypting the message data $msg$, other, analogous, cryptographic methods can be used such as IBE methods based on Weil or Tate pairings. Furthermore, the data $K_{ENC}$ may be subject to further predetermined processing (such as a further hash) before being used in the operative encryption process and in this case the trusted authority will need to use the same processed value of $K_{ENC}$ when generating $K_{DEK}$ (it will, however, be appreciated that the trusted authority will need to receive $K_{ENC}$ unprocessed in order for it to be able to access the individual components of $K_{ENC}$). These generalizations also apply to the variants discussed below.

[0086] In the above-described embodiment, the data provider's public key $K_{public}^{data}$ is used in clear in the encryption key string $K_{ENC}$ to identify the data provider and to encrypt the encoded component of the encryption key string $K_{ENC}$ whilst the corresponding private key $K_{private}^{data}$ is used as an authenticator of the identity of the data provider by its inclusion in the hashed component of the encryption key string $K_{ENC}$. Although in the above embodiment this public/private key pair $K_{public}^{data}/K_{private}^{data}$ is an IBE encryption/decryption key pair, this need not be the case and the public/private key pair could, for example, be an RSA public/private key pair. In this case, the private key used to authenticate the data provider 20 cannot be computed in step 64 and must be accessed by look up in a database kept by the user registration module 44 relating private key to the data-provider identifier, such as the public key, included in clear in the encryption key string $K_{ENC}$. A potential drawback of using an RSA public/private key pair is that if the public key is used as the in-clear data-prover identifier included in the encryption key string $K_{ENC}$, the real-world identity of the data provider may not be apparent to the data receiver and will generally need translation. In fact, the in-clear data-provider identifier included in the encryption key string $K_{ENC}$ need not be the public key of the public/private key pair (whether IBE or RSA based) but can be any valid identity for the data provider that is known and accepted by the trusted authority as corresponding to the private key it holds for the data provider 20.

[0087] It is also possible to use a symmetric key known only to the data provider and the trusted authority to form the encrypted component of the encryption key string $K_{ENC}$ and for inclusion in the hashed component in place of $K_{private}^{data}$. In this case, the in-clear identifier of the data provider that is included in the encryption key string $K_{ENC}$ would not, of course, be this key but would be an identifier known by the trusted authority as associated with the data provider and thus with the symmetric key concerned.

[0088] It may be noted that the key used for encrypting the encrypted component of the encryption key string $K_{ENC}$ need not be cryptographically related to the key used in the hashed component of $K_{ENC}$—all that is required is that the key used for encrypting the encrypted component of the encryption key string $K_{ENC}$ is confidential to the data provider 20 and the trusted authority 40 and is known by the latter to belong to the same party as the key used in the hashed component of the key $K_{ENC}$.

[0089] It may also be noted that where there are only a few users registered with the trusted authority, it would be possible to omit the first component $K_{public}^{data}$ (or other in-clear identifier of the data provider 20) and simply arrange for the trusted authority 40 to try out the keys/key pairs of all registered users to see if the message came from a registered user. If a key/key pair was found that both sensibly decrypted the encrypted component of $K_{ENC}$ and gave rise to a computed hash matching the hashed component of $K_{ENC}$, the identity of the data provider can be considered as established and can be passed to the data receiver if the latter needed to know this identity.

[0090] As already indicated, the form of encryption key string $K_{ENC}$ described above with reference to FIGS. 2 and 3 serves at least three purposes besides encryption of the message data $msg$: more particularly, it provides for passing a condition in confidence to the trusted authority, for authentication of the data provider to the trusted authority, and for the transmission and checking of message integrity data (the message hash). However, where only one or two of these functions is required, the form of the encryption key string $K_{ENC}$ can be simplified.

[0091] Considering first the authentication of the data provider, this is achieved by including in the encryption key string $K_{ENC}$ a hash of a shared secret known to the data provider 20 and the trusted authority 40; in the illustrated embodiment this shared secret is the private key $K_{private}^{data}$ but, as discussed, could be an RSA private key of the data provider or a symmetric key, or indeed any shared secret. The presence in the encryption key string of the Conditions 1 and 2 is not relevant to this data-provider authentication function so that the example encryption key string $K_{ENC}$ given above can be reduced to:

[0092] $K_{public}^{data}$

[0093] $H(K_{private}^{data} || H(msg) || nonce)$

[0094] $E(K_{public}^{data} || N; (H(msg) || nonce))$

[0095] As already noted, where there are only a few users registered with the trusted authority, the first component $K_{public}^{data}$ can be omitted. Furthermore, the nonce could be omitted from both the encrypted and hashed components of $K_{ENC}$ provided freshness was not required. However, retention of the message hash $H(msg)$ in both the hashed com-
ponent and the encrypted (or, alternatively, in the in-clear) component of $K_{ENC}$ is necessary where it is desired to retain a link between the originator identity established for the encryption key $K_{ENC}$ and a message encrypted with this key—removal of the message hash $H$(msg) would enable the encryption key string $K_{ENC}$ to be used by a third party for encrypting a message which might then appear to come from the data provider 20 in view of the latter’s identity being embedded in the encryption key string $K_{ENC}$. In fact, it is possible to envisage circumstances where the original of the encryption key string $K_{ENC}$ is of a significance independent of the origin of a message encrypted with that key. For example, where the encryption key string $K_{ENC}$ includes one or more conditions in clear and/or in the encrypted component, then these may represent standard terms and conditions of a party which wishes to establish this fact independently of any message encrypted with the encryption key string; in this case, the conditions (or a hash of the conditions) would need to be included in the hashed component of the encryption key string to enable a check on their integrity. Where only non-confidential conditions were involved, such as Condition 1, then the encryption key string $K_{ENC}$ would be of the form:

$$[0096] K_{20_{public}}$$

$$[0097] ::\text{Condition 1}$$

$$[0098] ::H(K_{20_{private}}; \text{ nonce}; \text{ Condition 1})$$

$$[0099] ::E(K_{20_{public}}, N; \text{ nonce})$$

[0100] If the nonce is not required, then the encrypted component can be omitted. The condition or conditions included in such an encryption key can be replaced, or supplemented, by other data not intended to be an identifier of the data receiver 30 such as data about the data provider 20. This other data can, like the conditions, be included in clear and/or in the encrypted component and should also be included, directly or after hashing, in the hashed component if it is to be linked to the originator identity established for the encryption key string $K_{ENC}$.

[0101] Rather than using an un-keyed hash function such as SHA1, it is possible to use a keyed hash such as HMAC with the private key $K_{20_{private}}$ (or other shared secret) being the hash key used for hashing the other element or concatenated elements of the hashed component. In this case, the trusted authority would use the same keyed hash function in seeking to compute a hash value matching that in the encryption key string $K_{ENC}$.

[0102] If the identity of the data provider is not an issue, then the encryption key string $K_{ENC}$ of the illustrated embodiment reduces to:

$$[0103] K_{20_{public}}$$

$$[0104] ::\text{Condition 1}$$

$$[0105] H(H$(msg)$; \text{ nonce}; \text{ Condition 1}; \text{ Condition 2})$$

$$[0106] ::E(K_{20_{public}}, N; (H$(msg)$; \text{ nonce}; \text{ Condition 2}))$$

[0107] leaving only the elements forming the identifiers of the data receiver (Conditions 1 and 2) and those concerned with the message integrity (the in-clear component $K_{20_{public}}$ is retained to facilitate the trusted authority obtaining the key $K_{20_{private}}$ for decrypting the encrypted component, though as already discussed, in appropriate circumstances the component $K_{20_{public}}$ can be omitted). If only message integrity is of interest (for example, if there are no conditions), then the hashed component of this reduced form of the encryption key string $K_{ENC}$ can be removed leaving:

$$[0108] K_{20_{public}}$$

$$[0109] ::E(K_{20_{public}}, N; (H$(msg)$; \text{ nonce}))$$

[0110] In fact, since $K_{20_{public}}$ is public, encrypting the message hash does not serve much purpose as anyone wishing to provide a substitute message for that originally sent can also change the message hash and encrypt it accordingly. However, if the message hash, with or without the addition of a nonce, is encrypted using a private key (whether of a public/private key pair or a secret symmetric key) the message hash is protected from change and serves its purpose of providing a message integrity check for the original message. Rather than using the private key to encrypt the message hash, it can be used to form a keyed hash, such as HMAC, of the message. The trusted authority can be arranged to determine the correct private key to use for checking either by trial and error through a limited set of such keys, or by the inclusion in the encryption key string $K_{ENC}$ of a suitable indicator in clear.

[0111] Whether the message hash is included in a protected form or another form (such as in clear or encrypted with a public key) in the encryption key string $K_{ENC}$, its inclusion permits detection of non-malicious changes in the encrypted message such as may result from problems in the communications path. At its simplest, inclusion of the message hash, in clear or in a derived form, into the encryption key string $K_{ENC}$ provides a link between the encryption key string and the message giving rise to the included hash value. Whilst this does have utility without the addition of further data into the encryption key string, it is primarily of interest for associating such further data included in the key $K_{ENC}$ with the message, this further data being, for example, identity information of the data provider and/or data receiver as has already described.

[0112] As regards the inclusion in the encryption key string $K_{ENC}$ of conditions serving to identify the data receiver, it will be appreciated that the number of in-clear and encrypted conditions can be varied from that described above for the illustrated embodiment. Thus, there may be none, one or more in-clear conditions and none, one or more encrypted conditions, in any combination. Furthermore, where the conditions are already known to the data receiver 30, the conditions need not be included as such in the encryption key string $K_{ENC}$ but they should still included in the hashed component to enable the trusted authority to confirm that the conditions passed to it by the data receiver correspond to those intended by the data provider and included in the hashed component. In this case, for the illustrated embodiment, the encryption key string $K_{ENC}$ reduces to:

$$[0113] K_{20_{public}}$$

$$[0114] ::H(K_{20_{private}}; H$(msg)$; \text{ nonce}; \text{ Condition 1})$$

$$[0115] ::E(K_{20_{public}}, N; (H$(msg)$; \text{ nonce}))$$

[0116] there being no Condition 2 as this example only involves conditions known to the data receiver. If the
data-provider identity information and message hash data are not required and the nonce is omitted, the encryption key string \( K_{\text{ENC}} \) further reduces to:

\[ \text{[0117]} \quad H(\text{Condition 1}) \]

\[ \text{[0118]} \quad \text{This is of value because it ensures that the data receiver can only read the encrypted message data supplied by the data provider if it presents, and satisfies, the correct condition 1 to the trusted authority. The data receiver cannot alter the hash value to match a different condition as this would result in a decryption key \( K_{\text{DEC}} \) that would not serve to decrypt the received encrypted message data.} \]

\[ \text{[0119]} \quad \text{It may be noted that it is possible to achieve a similar result to that of the foregoing paragraph without using an IBE schema for the encryption and decryption keys \( K_{\text{ENC}}, K_{\text{DEC}} \). Consider a situation where the trusted authority has a secret key \( K_t \) which it uses to generate a secret key \( K_p \) for the data provider 20 (the subscript "p" here standing for the data Provider):} \]

\[ \text{[0120]} \quad K_p = \text{HMAC}(K_t, \text{identifier of data provider}) \]

\[ \text{[0121]} \quad \text{This enables the data provider 20 to generate a symmetric key \( K_{\text{PR}} \):} \]

\[ \text{[0122]} \quad K_{\text{PR}} = \text{HMAC}(K_p, \text{identifier of data receiver}) \]

\[ \text{[0123]} \quad \text{where the identifier of the data receiver is the Condition 1. The key \( K_{\text{PR}} \) is then used with a symmetric encryption algorithm to encrypt the message data which the data provider then sends, along with its identifier, to the data receiver. In order for the data receiver to obtain the key \( K_{\text{PR}} \) for decrypting the message data, it must provide its identifier (Condition 1) and that of the data provider to the trusted authority who can now compute the key \( K_{\text{PR}} \) as it already knows \( K_p \) or can re-compute it; assuming that the data receiver meets the Condition 1, the trusted authority then returns the key \( K_{\text{PR}} \) to the data receiver to enable the latter to decrypt the encrypted message data. If the data receiver supplies a modified Condition 1, the resultant key will not decrypt the encrypted message data. By also sending a hash of the key \( K_{\text{PR}} \), the data provider can provide assurance to the trusted authority that \( K_{\text{PR}} \) has been created by the data provider.} \]

\[ \text{[0124]} \quad \text{With regard to the above-described reduced forms of the encryption key string \( K_{\text{ENC}} \) it will be understood by persons skilled in the art that the FIG. 4 process carried out by the trusted authority is appropriately modified to omit any unnecessary computation or checks and to effect any changes needed to take account of the changed form of the encryption key string \( K_{\text{ENC}} \).} \]

\[ \text{[0125]} \quad \text{It will also be understood by persons skilled in the art that where elements are concatenated before being operated upon by a hashing or encryption function, the order of concatenation may be varied from that described above provided that the ordering is used consistently (for example, the trusted authority 40, when computing the hash value in step 67 of FIG. 4, must use the same ordering of the concatenated elements as used by the data provider 20 when generating the encryption key string \( K_{\text{ENC}} \). Indeed, elements can be combined in ways other than by concatenation. Thus, the concatenation operations performed by the data provider 20 that must be reversed by the trusted authority 40 can be replaced by any reversible combination function, whilst the concatenation operation performed by the data provider 20 in combining the elements for the hashed component:} \]

\[ \text{[0126]} \quad H(K_{\text{PR}}; H(\text{msg}); \text{nonce}; \text{Condition 1}; \text{Condition 2}) \]

\[ \text{[0127]} \quad \text{(or any simplified version discussed above) can be replaced by any deterministic combination function (the trusted authority needing only to be able to repeat the combination, not reverse it). Of course, the trusted authority and data provider must know to use the same combination functions.} \]

1. A data-transfer method comprising:

(a) encrypting first data at a first party using an encryption parameters both public data of a trusted party and an encryption key string comprising a hash value generated by hashing second data comprising at least one condition;

(b) providing the encrypted first data to a second party, and the encryption key string and said at least one condition to a trusted party;

(c) at the trusted party effecting both a first check that the hash value in the encryption key string matches the result of a hash based on the said at least one condition provided in step (b), and a second check that the or each said at least one condition is met; and

(d) providing a decryption key to the second party only if said checks are satisfactory, this decryption key being generated at the trusted party using the encryption key string and private data related to said public data.

2. A method according to claim 1, wherein in step (b) the encryption key string is provided to the trusted party via the second party.

3. A method according to claim 1, wherein in step (b) said at least one condition is provided to the trusted party by the second party.

4. A method according to claim 1, wherein in step (a) said second data comprises at least one further element in addition to said at least one condition; the first check carried out in step (c) checking that the hash value in the encryption key string matches the result of a hash of a combination of said at least one further element and the said at least one condition provided in step (b).

5. A method according to claim 1, wherein the encryption process effected using said encryption key and the public data of the trusted party is an identifier-based cryptographic process utilising quadratic residuosity.

6. A method according to claim 1, wherein the encryption process effected using said encryption key and the public data of the trusted party is an identifier-based cryptographic process utilising Weil or Tate pairings.

7. A data-transfer system for the encrypted transfer of first data, the system comprising:

- data provider equipment comprising:
  - a hashing arrangement for generating a hash value by hashing second data comprising at least one condition;
  - a keystring-forming arrangement for forming an encryption key string using at least said hash value;
an encryption arrangement for encrypting the first data using as encryption parameters both public data of a trusted party and said encryption key string; and
an output arrangement for outputting at least the encrypted first data and the encryption key string;
data receiver equipment for receiving the encrypted first data output by the data provider equipment; and
trusted party equipment associated with said trusted party and comprising:
an input arrangement for receiving the encryption key string and said at least one condition;
a first checking arrangement for carrying out a first check that the hash value in the encryption key string matches the result of a hash based on the said at least one condition;
a second checking arrangement for carrying out a second check that the or each said at least one condition is met; and
a decryption-key providing arrangement for providing a decryption key to the data receiver equipment only if said checks are satisfactory, the decryption-key providing arrangement being arranged to generate this decryption key using the encryption key string and private data related to said public data.

8. A system according to claim 7, wherein the data receiver equipment is arranged to receive the encryption key string from the data provider equipment and forward it to the trusted party equipment.

9. A system according to claim 7, wherein the data receiver equipment is arranged to provide said at least one condition to the trusted party equipment.

10. A system according to claim 7, wherein said second data comprises at least one further element in addition to said at least one condition; the first checking arrangement being arranged to check that the hash value in the encryption key string matches the result of a hash of a combination of said at least one further element and the said at least one condition.

11. A system according to claim 7, wherein the encryption arrangement is arranged to encrypt the first data, using said encryption key string and the public data of the trusted party, in accordance with an identifier-based cryptographic process utilising quadratic residuosity.

12. A system according to claim 7, wherein the encryption arrangement is arranged to encrypt the first data, using said encryption key string and the public data of the trusted party, in accordance with an identifier-based cryptographic process utilising Weil or Tate pairings.

* * * * *