Title: METHOD AND SYSTEM FOR A CERTIFICATE-LESS AUTHENTICATION ENCRYPTION (CLAE)

Abstract: A method of verifying a plurality of public parameters from a trusted center in an identity-based encryption system prior to encrypting a plaintext message by a sender having a sender identity string is described. The method includes identifying the trusted center by a TC identity string, the trusted center having an identity-based public encryption key of the trusted center based on the TC identity string; determining if the sender has a sender private key and the public parameters for the trusted center including the public encryption key of the trusted center and a bilinear map; and verifying the public parameters using the TC identity string prior to encrypting the plaintext message into a ciphertext by comparing values of the bilinear map calculated with variables comprising the sender private key and the public encryption key. The ciphertext may include an authentication component for authenticating the sender once the ciphertext is received and decrypted by the recipient using the identity string of the sender and the private key of the recipient.
Method and System for a Certificate-less Authentication Encryption (CLAE)

FIELD OF THE INVENTION

[1] This invention relates to a certificate-less authenticated encryption scheme and, in particular, an encryption scheme using identity strings to provide identity-based encryption.

BACKGROUND OF THE INVENTION

[2] Cryptographic encryption algorithms add confidentiality to sensitive data that is transmitted over an insecure channel. The data is protected, as the encryption algorithm transforms the data from plaintext into ciphertext prior to transmission. The recipient of the encrypted data is only able to decrypt the ciphertext and retrieve the plaintext from the received transmission if the recipient is able to reverse the encryption algorithm. If the encryption and decryption algorithms share the same key, the cryptosystem is known as "symmetric" and the algorithms are called symmetric-key algorithms. If the key in the encryption algorithm is different than the key in the decryption algorithm, the cryptosystem is known as "asymmetric" and the algorithms are called asymmetric-key algorithms.

[3] In asymmetric-key algorithms, the key used for encryption (i.e. the "public key") is publicly known, as everyone should be able to use it to encrypt sensitive data. However, the key used in the decryption (i.e. the "private key") is only known to the intended receiver of the encrypted data and is protected such that the intended receiver is the only
entity able to decrypt the encrypted message. An asymmetric cryptosystem is commonly referred to as a Public Key Cryptosystem (PKC).

[4] In a PKC, the public key and the private key are independent of each other such that knowledge of the public key does not reveal or lead to the private key. In other words, the public key can be made public such that anyone can encrypt data for a specific recipient, but only the specific recipient has knowledge of the private key and is able to utilize the private key to decrypt and retrieve the data. Since the public keys in the PKC are publicly known, they are considered insensitive and can be transmitted over any insecure, public channel. However, the main challenge with the PKC is to trust whether an available public key is actually associated with the intended recipient. In other words, if a different public key (i.e. a wrong or modified public key) is used by mistake or by fraud, the overall security achieved by utilizing encryption is compromised. The security of the encryption in a Public Key Cryptosystem therefore relies on correctly distributing the public keys that belong to or are associated with the intended recipients of the encrypted message. Accordingly, it is necessary to verify the public keys before encrypting sensitive data with a public key in a PKC.

[5] Since large systems are dynamic and new members join or leave the system at all times, public keys are constantly issued and/or revoked. At the time of registration (setup), a new member is assigned a new set of public/private keys and all the other existing members are notified of the new public key before they can securely communicate with the new member, using the new public key generated.

[6] In the PKC, there are two mechanisms for generating and distributing the public keys throughout the system. In the first mechanism, the public keys are generated by a
trusted center, which would then distribute them remotely over a secure channel to the
users in the system. The second mechanism is for a sender to generate the public key
locally for every recipient. In this way, the trusted center is not required to first generate a
private key for every recipient and then distribute these generated public keys remotely
over a secure channel to every sender.

[7] Generating public keys locally is superior to relying on a trusted center to provide
the public keys. When the public encryption keys are generated locally, the latency of
encryption is reduced in that it is no longer necessary to retrieve a certificate from a
remote server.

[8] Traditionally in the PKC, public keys are generated by a trusted center (certificate
authority) guaranteeing that a public key belongs to a certain recipient. The certificate
authority is a trustworthy entity that distributes the certificates throughout the PKC. In a
typical PKC, the trusted center is operable to produce an X.509 certificate that includes the
public key for a recipient as well as other ancillary data. The trusted center then digitally
signs the provided certificate, in order for the sender to verify the authenticity of the
provided certificate and the corresponding public key. Nevertheless, distributing and
managing the public key certificates in a large system is a challenging task, as the
certificates have to be protected from tampering over insecure channels during
transmission or when received at the sender's local machine.

[9] An alternate approach to public key encryption is to self-generate the public
parameters that would be used to encrypt sensitive data using the recipient's known
identity, such as a phone number, email address or username. Boneh and Franklin have
introduced an Identity-Based Encryption (IBE ®) scheme in which the identity of the
recipient is used in the encryption, such as described in Dan Boneh and Matthew Franklin, "Identity-Based Encryption from the Weil Pairing" SIAM Journal of Computing, 32(3):586-615, 2003 and U.S. Patent No. 7,13,594, the contents of which are hereby incorporated by reference in their entirety. In their setup, every user is given a private key, but the encryption key is constructed using the identity of the recipient and the trusted center's public key. Their system removes the need to contact the trusted center (certificate authority) to retrieve the public key of a recipient. However in their system, the public key of the trusted center ($P_{pub}$) has to be strictly protected. If a different public key is used in the encryption by mistake or fraud, the security of the encryption is entirely compromised.

[10] It should be noted that the entire security of their scheme relies on the security of the public key of the trusted center, which is publicly known and therefore widely available. If an adversary can change the public parameter(s) of the trusted center either by accessing the local storage of trusted center's public key or by sending a different public key via a man-in-the-middle attack, the security of the encryption system is compromised.

SUMMARY OF THE INVENTION

[11] The present invention is directed to provide an improved certificate-less authenticated encryption (CLAE) method and an authentication system using identity-based encryption.

[12] It is an object of the present invention to configure a PKC system which has eliminated the need for distributing and managing public keys throughout the system. Instead, the public keys are generated and verified locally. Once the system is initialized, any entity in the system can self-generate the public key of any other entities and encrypt
sensitive data by using the recipient's identity, such as a phone number, email address or username. Only the true recipient is then able to decrypt and retrieve the sensitive data using a private key known only to the recipient and obtained from a trusted center.

[13] One of the many security challenges in the PKC systems is protecting the public key certificates from tampering and securely distributing them throughout the system. Boneh and Franklin, as discussed above, proposed an IBE ® scheme in which the public identity of users is used to generate the encryption keys. However, the same problems occur with the public key of the trusted center (i.e. the "key generator", according to Boneh and Franklin). If \( P_{pub} \) and \( P \) are replaced fraudulently, the fraudster can easily access the encrypted messages. This attack is possible since \( P_{pub} \) and \( P \) in the Boneh and Franklin setup are publicly known and widely available. Therefore, the public keys are not protected at all or they are less protected throughout the system than the private keys or the secret master key. Furthermore, the public parameters are broadcast throughout the system via a public, insecure channel. Therefore, an adversary may try to change the values of \( P_{pub} \) and \( P \) in the encryption algorithm.

[14] As described in the prior art, if a fraudster substitutes \( P_{pub} \) with any other point, such as \( xP \) where \( x \) is known to the fraudster, the adversary can easily find \( g_{ij}^{'} \) (i.e. the "session key", according to Boneh and Franklin) and reverse the encryption of \( M \). This is further shown as follows: as described by Boneh and Franklin, discussed above, we have

\[
g_{ij} = e(Q_{ij}, P_{pub})
\]

which under the fraudster's modification would become

\[
g_{ij}^{'} = e(Q_{ij}, xP).
\]

The corresponding message key would be \( g_{ij}^{''} \), where it can be easily found from \( V = rP \) in the Voltage Security IBE ®. This is performed in IBE ® by computing:
\[ e(Q_p, Y) = e(Q_p, rPY = e(Q_p, xpY = g^y \) \]

[15] In contrast, the CLAE scheme according to the present invention allows the sender to locally verify the public keys (i.e. \( P_{pub} \)) of the server before encrypting the message. In other words, the sender is operable to verify the trusted center (TC) before encrypting a message, thereby ensuring that the public parameters have not been modified. The point of trust in the CLAE scheme of the present invention is established from the public identity of the server (e.g. "abc.com") and, unlike the prior art, it is not a fixed parameter that can be altered.

[16] In one aspect of the present invention, a new CLAE framework has been designed that uses the identity of the recipient to remove the need for public key certificates. Instead of using a predetermined parameter to generate the public/private encryption key, the user incorporates the identity of the trusted center as well as the identity of the recipient. In this manner, greater flexibility is provided in generating the encryption keys, as the user can arbitrarily choose any trusted center using its own identity and can be assured that its selection will be enforced on the recipient. For instance, the user might want to send an encrypted email from an "abc.com" account to someone with an "xyz.com" account. In this case, the user can choose either "abc.com" or "xyz.com" as the trusted center simply by using the trusted center's identity in the encryption process. The recipient is then forced to verify itself to the trusted center chosen by the sender. Such a system may also allow the sender of an encrypted message to verify one or more of the public parameters to ensure that they have not been tampered with.

[17] In one aspect, the present invention resides in a method of sending an encrypted message by a sender having a sender identity string \((Id_{sender})\) to a recipient over a network
using identity-based encryption, the method may include identifying a trusted center (TC) by a TC identity string (Idrc), the trusted center having an identity-based public encryption key of the trusted center (gpub) based on the TC identity string (Id'.c). Furthermore, the method may include determining if the sender has a sender private key (Prv-send) and a plurality of public parameters (PK) for the chosen trusted center (TC) and the public parameters (PK) including the identity-based public encryption key of the trusted center (gpub) and a bilinear map (e). Furthermore, the method may include verifying the public parameters (PK) of the trusted center (TC) using the TC identity string (Idrc) prior to encrypting a plaintext message (hi). Furthermore, the method may include identifying the recipient by a recipient identity string (Id_recipient), the recipient having an identity-based public encryption key of the recipient (g_recipient) on the trusted center (TC) based on the recipient identity string (Id_recipient). Furthermore, the method may include generating the identity-based public encryption key of the recipient (g_recipient) using the public parameters (PK) and the recipient identity string (Id_recipient). Furthermore, the method may include encrypting the plaintext message (M) as ciphertext (C) using the public parameters (PK), a random symmetric key (σ) and the identity-based public encryption key of the recipient (g_recipient), the ciphertext (C) including the encrypted message. Finally, the method may include transmitting the ciphertext (C) to the recipient over the network.

[18] In another aspect, the present invention resides in a method for using certificate-less authenticated encryption (CLAE) using identity-based encryption in a network system between a sender having a sender identity string (Id_send) and a recipient having a recipient identity string (Id_recipient), the method may include at the sender: identifying a trusted center (TC) by a TC identity string (Idrc), the trusted center having an identity-based public encryption key of the trusted center (gpub) based on the TC identity string (Idrc):
determining if the sender has a sender private key \((Prv_{\text{sender}})\) and a plurality of public parameters \((PK)\) for the trusted center \((TC)\), the public parameters \((PK)\) including the identity-based public encryption key of the trusted center \((g_{p,a})\) and a bilinear map \((e)\); verifying the public parameters \((PK)\) of the trusted center \((TC)\) using the TC identity string \((Id_{\text{r}})\) prior to encrypting a plaintext message \((M)\); identifying the recipient by a recipient identity string \((Id_{\text{r}})\), the recipient having an identity-based public encryption key of the recipient \((g_{e_{\text{piem}}}^{\text{r}})\) for the trusted center \((TC)\) based on the recipient identity string \((Id_{\text{r},p_{\text{piem}}}^{\text{r}})\); generating the identity-based public encryption key of the recipient \((g_{e_{\text{piem}}}^{\text{r}})\) using the public parameters \((PK)\) and the recipient identity string \((Id_{\text{r}})\); encrypting the plaintext message \((M)\) as ciphertext \((C)\) using the public parameters \((PK)\), a random symmetric key \((\sigma)\) and the identity-based public encryption key of the recipient \((g_{e_{\text{piem}}}^{\text{r}})\), the ciphertext \((C)\) including the encrypted message; and transmitting the ciphertext \((C)\) to the recipient over the network. Furthermore, the method may include at the recipient:

receiving the ciphertext \((C)\) from the sender over the network system; determining if the recipient has a recipient private key \((Prv_{\text{reci}})\) and the public parameters \((PK)\) for the trusted center \((TC)\); decrypting the ciphertext \((C)\) to obtain the plaintext message \((M)\) using the public parameters \((PK)\), and the recipient private key \((Prv_{\text{reci}})\).

[19] In another aspect, the present invention resides in a method of verifying a plurality of public parameters \((PK)\) from a trusted center \((TC)\) in an identity-based encryption system prior to encrypting a plaintext message \((M)\) by a sender having a sender identity string \((Id_{\text{s}}^{\text{s}})\). The method may include identifying the trusted center \((TC)\) by a TC identity string \((Id_{\text{r}})\), the trusted center having an identity-based public encryption key of the trusted center \((g_{p,a}^{\text{r}})\) based on the TC identity string \((Id_{\text{r}})\). Furthermore, the method may include determining if the sender has a sender private key \((Prv_{\text{s}})\) and the plurality
of public parameters ($PK$) for the trusted center ($TC$), the public parameters ($PK$)
including the identity-based public encryption key of the trusted center ($g_{p,ub}$) and a
bilinear map ($e$). Finally, the method may include verifying the public parameters ($PK$) of
the trusted center ($TC$) using the TC identity string ($Id_{tc}$) prior to encrypting the plaintext
message ($M$) by comparing values of the bilinear map ($e$) calculated with variables
comprising the sender private key ($Prv_{sender}$) and the identity-based public encryption key
of the trusted center ($g_{p,ub}$).

[20] In another aspect, the present invention resides in a system for sending an
encrypted message over a network using identity-based encryption. The system may
include: a trusted center ($TC$) having a TC identity string ($Id_{rc}$), a sender having a sender
identity string ($Id_{sender}$), and a recipient having a recipient identity string ($Id_{recipient}$). The
trusted center ($TC$) may include a first memory and one or more processors configured for:
generating a plurality of public parameters ($PK$) and a secret master key ($s$) from a security
parameter ($\lambda$), the public parameters ($PK$) including a bilinear map ($e$) and an identity-
based public encryption key of the trusted center ($g_{p,ub}$) based on the TC identity string
($Id_{rc}$); receiving a request from a requestor; if the request from the requestor contains an
identifier ($Id$) identifying the requestor, generating a private key ($Prv_{rq}$) based on the
identifier ($Id$) and the secret master key ($s$); and transmitting the private key ($Prv_{Id}$) to the
requestor over the network system; and if the request from the requestor includes a request
for the public parameters ($PK$), and transmitting the public parameters ($PK$) to the
requestor over the network system. The sender may include a second memory and one or
more processors configured for: identifying the trusted center ($TC$) by the TC identity
string ($Id_{rc}$), determining if the sender has a sender private key ($Prv_{sender}$) and the public
parameters ($PK$) for the trusted center ($TC$), verifying the public parameters ($PK$) of the
trusted center \((TQ)\) using the TC identity string \((Id_{tc})\) prior to encrypting a plaintext message \((M)\); identifying the recipient by the recipient identity string \((Id_{recipient})\), the recipient having an identity-based public encryption key of the recipient \((g_{recipient})\) for the trusted center \((TQ)\) based on the recipient identity string \((Id_{recipient})\); generating the identity-based public encryption key of the recipient \((g_{recipient})\) using the public parameters \((PK)\) and the recipient identity string \((Id_{recipient})\); encrypting the plaintext message \((M)\) as ciphertext \((Q)\) using the public parameters \((PK)\), a random symmetric key \((\sigma)\) and the identity-based public encryption key of the recipient \((g_{recipient})\), the ciphertext \((Q)\) including the encrypted message; and transmitting the ciphertext \((Q)\) to the recipient over the network. The recipient may include a third memory and one or more processors configured for: receiving the ciphertext \((Q)\) from the sender over the network system; determining if the recipient has a recipient private key \((Prv_{recipient})\) and the public parameters \((PK)\) for the trusted center \((TQ)\); decrypting the ciphertext \((Q)\) to obtain the plaintext message \((M)\) using the public parameters \((PK)\), and the recipient private key \((Prv_{recipient})\).

[21] In another aspect, the present invention resides in a computer program product comprising a computer readable memory storing computer executable instructions thereon that, when executed by a computer, perform the method of: identifying a trusted center \((TQ)\) by a TC identity string \((Id_{tc})\), the trusted center having an identity-based public encryption key of the trusted center \((g_{trust})\) based on the TC identity string \((Id_{tc})\); determining if a sender has a sender private key \((Prv_{sender})\) and a plurality of public parameters \((PK)\) for the trusted center \((TQ)\); the public parameters \((PK)\) including the identity-based public encryption key of the trusted center \((g_{trust})\) and a bilinear map \((e)\); verifying the public parameters \((PK)\) of the trusted center \((TQ)\) using the TC identity string...
prior to encrypting a plaintext message (M); identifying a recipient by a recipient identity string (Id_recepi); the recipient having an identity-based public encryption key of the recipient (g_recepi) for the trusted center (TC) based on the recipient identity string (Id_recepi); generating the identity-based public encryption key of the recipient (g_recepi) using the public parameters (PK) and the recipient identity string (Id_recepi); encrypting the plaintext message (M) as ciphertext (C) using the public parameters (PK), a random symmetric key (σ) and the identity-based public encryption key of the recipient (g_recepi), the ciphertext (C) including an encrypted message based on the plaintext message (M); and transmitting the ciphertext (C) to the recipient over a network.

Further and other features of the invention will be apparent to those skilled in the art from the following detailed description of the embodiments thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

Reference may now be had to the following detailed description taken together with the accompanying drawings in which:

FIG. 1 shows a network system in accordance with an embodiment of the present invention;

FIG. 2 shows a flowchart illustrating a method for sending an encrypted message in accordance with an embodiment of the present invention;

FIG. 3 shows a plurality of users in a network registering with the trusted center in accordance with an embodiment of the present invention;
FIG. 4 shows a flowchart for a user labeled "Identity String" registering with the trusted center in accordance with an embodiment of the present invention;

FIG. 5 shows a user labeled "User A" transmitting an encrypted message to a user labeled "User B" in a network with certificate-less authenticated encryption using identity-based encryption in accordance with an embodiment of the present invention;

FIG. 6 shows a flowchart for a user labeled "User A" encrypting a message to a user labeled "User B" in accordance with an embodiment of the present invention;

FIG. 7 shows a flowchart for a user labeled "User A" determining whether to get the public key for the trusted center in accordance with an embodiment of the present invention;

FIG. 8 shows a flowchart for a user labeled "User B" decrypting the message from a user labeled "User A" in accordance with an embodiment of the present invention; and

FIG. 9 shows a flowchart for a user labeled "User B" determining whether to get its private key from the trusted center in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[24] A network system 10 in accordance with an embodiment of the present invention is shown in FIG. 1. The network system 10 includes a trusted center 20, a user 30 labeled as "Sender" and a user 30 labeled as "Recipient" connected over a network 2, such as an intranet, the internet, and the like. While the two users 30 may be labeled differently, it should be understood that the labels are arbitrary and may change based on the direction
an encrypted message is being sent. A "Sender" is a user 30 operable to package a plaintext message as an encrypted message for transmission and a "Recipient" is a user 30 operable to receive the encrypted message from the "Sender". Upon a response to an encrypted message, the "Recipient" may become the "Sender", and vice versa.

[25] Each of the users 30 labeled "Sender" and "Recipient" are configured with a memory 32 and one or more processors 34. It should be understood that any additional hardware as known to those skilled in the art may be included, such as dedicated circuits, a field programmable gate array (FPGA), and the like. Each of the users 30 may exist on separate computers and/or mobile devices incorporating the necessary operating system, software and/or browsers as known to those skilled in the art.

[26] Similarly, the trusted center 20 may exist as a dedicated server or as part of a distributed network having memory 22 and one or more processors 24. The trusted center 20 may also include additional hardware and software components as known the art, such as firewalls and associated security mechanisms. The trusted center 20 is connected to the "Sender" and "Recipient" over network 2.

[27] In operation, the network system 10 in accordance with the present invention is operable to transmit information from the "Sender" to the "Recipient" using a certificate-less authenticated encryption (CLAE) scheme. Each of the users 30 is operable to communicate with the trusted center 20 to obtain a respective private key (Prv) and a plurality of public parameters (PK). The public parameters (PK) are specific to the trusted center, which includes an identity-based public encryption key of the trusted center (g_{p_e}b). Once these parameters (i.e. Prv and PK) have been obtained by the respective "Sender" and "Recipient", the sender and recipient are operable to communicate independent of the
trusted center 20 over a secure channel by encrypting the message using the recipient's public key, which is generated locally by the sender using the public parameters (PK) obtained from the trusted center (TQ) and a recipient identity string (Id\textsubscript{rec}pie\textsuperscript{i}) associated with the recipient. Furthermore, prior to encrypting a message, the sender is operable to verify the public parameters (PK) of the trusted center (TQ) using the trusted center identity string (Id\textsubscript{rc}), which is known to the sender, to ensure that the public parameters (PK) have not been modified.

[28] The CLAE scheme is based on identity-based encryption (IBE®) and, as shown in the flowchart 100 of a preferred embodiment as seen in FIG. 2, generally operates as follows:

a. In step 110, the user 30 (i.e. "Sender") identifies a trusted center (TC) by a TC identity string (Id\textsubscript{rc}, i.e. "xyz.com" or "name of TC").

b. In step 120, the "Sender" determines if it has a sender private key (Prv\textsubscript{sender}) and a plurality of public parameters (PK) associated with or generated by the trusted center (TO).

c. In step 130, the "Sender" verifies the public parameters (PK) of the trusted center (TC) prior to encrypting a plaintext message (M). The verification process relies on the properties of the public parameters (PK) and sender private key (Prv\textsubscript{sender}), both generated by the trusted center (TC), and the known TC identity string (Id\textsubscript{rc}) and the sender identity string (Id\textsubscript{sender}). In particular, the verification process relies on the mathematical properties of the bilinear map (e), which forms part of the public parameters, as further discussed below.
d. In step 140, the "Sender" identifies the user 30 (i.e. "Recipient") to receive the plaintext message \((M)\) by a recipient identity string \((Id_{recipen})\). The recipient identity string may be an email address, a phone number, a name, and the like.

e. In step 150, the "Sender" generates the identity-based public encryption key of the recipient \((g_{e,dp,ent})\) using the public parameters \((PK)\) and the recipient identity string \((Id_{recipiem})\). In this manner, communication with the trusted center \((TC)\) is not required and a certificate issued by the trusted center \((TC)\) is not necessary to authenticate the recipient's public key \((g_{e,cepem})\). Generating the recipient's public key \((g_{e,cepem})\) locally also allows for offline encryption.

f. In step 160, the "Sender" encrypts the plaintext message \((M)\) as ciphertext \((Q)\). The ciphertext \((C)\) includes the encrypted message, as well as ancillary information necessary for decrypting the message. Additional authentication information may also be appended to the ciphertext \((C)\).

g. In step 170, the "Sender" transmits the ciphertext \((C)\) to the "Recipient" over the network 2. As the message is encrypted, the message may be sent over an unsecured channel. A user will require the recipient private key \((Prv_{recipiem})\) and the public parameters for the trusted center \((PK)\) to read the plaintext message \((M)\).

[29] In this manner, the "Sender" is able to send a plaintext message \((M)\) as an encrypted message to the "Recipient" without accessing the trusted center \((TC)\), as long as the "Sender" has the required public parameters \((PK)\) and its own sender private key \((Prv_{sende})\). Furthermore, with the public parameters \((PK)\) and its own sender private key \((Prv_{sende})\), the "Sender" is able to verify that the public parameters \((PK)\) have not been
compromised. In this manner, the "Sender" is operable to ensure that only the "Recipient" having the recipient private key \( (Prv_{\text{recipient}}) \) will be able to decrypt the ciphertext \( (C) \).

[30] To implement the CLAE^ scheme, and the method of the preferred embodiment discussed above, four major algorithms are utilized. It should be understood that additional algorithms, application programming interfaces (APIs), methods, and/or functions will be known and/or implemented by those skilled in the art to provide common functions and operations necessary to implement the network system according to the present invention. The four major algorithms in accordance with a preferred embodiment include:

**Setup** \((\lambda)\): This algorithm is run by the trusted center 20 (i.e. system administrator) which provides the encryption/decryption services. It takes as inputs the security parameter \( (\lambda) \) and outputs the public parameters \( (PK) \) of the CLAE and the secret master key \( (s) \) of the administrator (e.g. with the identity string "admin") as the trusted center \( (TC) \).

**KeyGen** \((Id, s)\): This algorithm is also run by the trusted center 20 (i.e. administrator) which distributes the private keys \( (Prv) \) throughout the system. It takes as inputs an identity string \( (Id) \), which may be received from a user 30, and the secret master key \( (s) \) from the administrator. It outputs a private key \( (Prv_{Id}) \) that will be used for decryption in the Auth-Decrypt() algorithm, but which may also be incorporated into the Verifi-Encrypt() algorithm to allow for the authentication of a sent message. While the **KeyGen**() algorithm is run by the trusted center 20, it may be run on behalf of a user 30 which has provided its identity string \( (Id) \) to the trusted center \( (TC) \).
Verifi-Encrypt \((Id, PK, M)\): This algorithm is run independently by individual users who wish to encrypt sensitive data (i.e. the "Sender"). It takes as inputs the identity string of the recipient \((Id, \text{or more specifically, } Id_{\text{ recipient}})\), the set of public parameters \((PK)\) and the plaintext message \((M)\). The algorithm first verifies whether the public parameters \((PK)\) are genuine under the given administrative network for the specific trusted center and then outputs the ciphertext message \((C)\) where \(C = (V, U, W, T)\) and where \(V, U,\) and \(W\) are parameters that are necessary to decrypt and recover the plaintext message \((M)\), and \(Y\) is a parameter which may be used by the recipient to authenticate the sender.

Auth-Decrypt \((Prv/ PK, C)\): This algorithm is run independently by individual users who are recipients of the ciphertext \((C)\) and wish to decrypt and access the plaintext message \((M)\) from the ciphertext \((C)\). The algorithm takes as inputs the private key of the recipient \((Prv/\text{ or more specifically, } Prv_{\text{ recipient}})\), the public parameters \((PK)\) and the ciphertext message \((C)\). If the recipient has the proper private key \((Prv)\) under the same administrative network associated with identity string of the recipient \((Id)\), the algorithm outputs the plaintext message \((M)\).

[31] The internal structure of each of the four-mentioned algorithms in accordance with a preferred embodiment will be further discussed below, including the mathematical basis which may provide the functionality of the mentioned algorithms.

Bilinear Pairings

[32] The CLAE scheme is based on Bilinear pairings. A bilinear pairing is formally defined as follows:
Let $G$ and $G_\tau$ be groups of order $p$, and let $g$ be a generator of $G$. A bilinear map is an efficiently computable function from $G \times G$ onto $G_\tau$, such that it has the following properties:

1. Bilinearity: for all $g, g' \in G$,
\[
e : G \times G \rightarrow G_\tau,
\]
\[
e(gg', h) = e(g, h)e(g', h),
\]
\[
e(g, hh') = e(g, h)e(g, h')
\]

Note: that $e(\cdot, \cdot)$ is symmetric, that is $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab} \forall a, b$.

2. Non-degeneracy: If $e(g, h) = 1$ for all $h \in G$, then $g = I$ (identity).

Weil pairing and Tate pairing are two implementations of an efficient bilinear map over elliptic curve groups useful for cryptography, such as described in Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart, editors, Advances in Elliptic Curve Cryptography, Cambridge University Press, 2005, the contents of which are herein incorporated by reference in its entirety. Cryptographic bilinear maps must have certain complexity properties that are explained in the following section.

**Complexity Assumptions**

In general, cryptographic bilinear maps need to be one-way functions, i.e.

computing the bilinear pairing should be efficient, but the inverse has to be difficult. The Bilinear Diffie-Hellman (BDH) complexity assumption is related to the difficulty of solving Discrete Logarithm Problem (DLP) over large algebraic groups.
Definition 1 (BDH Assumption): Let $G$ and $G_T$ be groups of order $p$ with a bilinear map $e: G \times G \rightarrow G_T$, where $g$ is a generator for $G$ and $a, b, c \sim Z_p^*$. Given \( \{g^a, g^b, g^c\} \), computing $e(g^a, g^b) e^{abc} e G_T$, without knowing $a$, $b$ or $c$ is difficult.

Informally, the decision variant of the BDH assumption states that the probability of correctly distinguishing $e(g, g)^{abc} e G_T$ from a random choice $R e G_T$ is negligible $\text{□}$. A function $g : R \rightarrow R$ is negligible if for any $d > 0$ we have $g(\lambda) \ll 1 / \lambda^d$ for sufficiently large $\lambda$. Formal definitions proceeds below.

Decision BDH Assumption: Suppose that a challenger chooses $a, b, c, z \in Z_p^*$ at random. $g$ is generator for the group $G$ over which the bilinear mapping $e(\cdot, \cdot)$ is defined. Decision BDH assumption states that no polynomial-time adversary can distinguish the tuple $(A = g^a, B = g^b, C = g^c, Z = e(g, g)^{abc})$ from the tuple $(A - g^a, B = g^b, C = g^c, Z = e(g, g)^{abc})$ with a non-negligible probability.

This is a standard assumption in cryptography and provides a foundation for the proof of security for the main scheme.

Certificate-Less Authenticated Encryption

The details of the main algorithms in the CLAE scheme in a preferred embodiment of the present invention, using the bilinear mapping and assumptions provided above, are given as follows:

**Setup**($\lambda$): It takes as inputs the security parameter $\lambda$ and then generates the groups $G$ and $G_T$ of a prime order $p$ and a bilinear map $e$. The size of the groups is...
determined by $\lambda$. Let's denote the identity string of the trusted center by "admin" -
note that any other string representing the trusted center, e.g. "abc.com" or
"xyz.com" could be used. Select a symmetric key encryption function $\Box$ in which
the shared key is n-bit long. We denote the decryption function corresponding to $\Box$
by $D$, and it should be clear that by knowing $\Box$ it would be easy to find $D$. Choose
three cryptographic hash functions $H_1, H_2$ and $H_3$ such that $H_1 : \{0,1\}^* \rightarrow G$,
$H_2 : G_y \rightarrow \{0,1\}^n$, and $H_3 : \{0,1\}^n \times \{0,1\}^* \rightarrow Z_p^*$. Pick at random
$s \leftarrow_R Z_p^*$ and set $g_{\text{admin}} = H_1(\"admin\") \in G$. Compute $g_{\text{pub}} = g_{\text{admin}}^s$ as the trusted center's
public key. Let's denote the public parameters of the CLAE by public parameters
(PK) that includes a description of $\{G, G_y, e, g_{\text{pub}}, H_1, H_2, H_3, \Box \}$. Set $s$ as the
secret master key, which is only known to the trusted center. Output PK and $s$.

**KeyGen(Id, s):** It takes as inputs an identity string of a user (Id) and the secret
master key ($s$) of the administrator. It computes $g_{\text{Id}} = H_3(\text{Id}) \in G$ and sets the
private key of the user as $Prv_{\text{Id}} = g_{\text{Id}}^{1/s}$. It then outputs $Prv_{\text{Id}}$ along with the public
parameters (PK) and privately shares it with the user, for example, over a secure
channel. The private key of the user ($Prv_{\text{Id}}$) may also be shared with the user by
other secure mechanisms, for example, over short message service (SMS) of a
mobile device, in person, via a piece of paper which is later discarded, and the like.
Although the private key ($Prv_{\text{Id}}$) must be kept secure, the public parameters (PK)
are very resilient to attacks and can even be sent over an insecure channel and/or
be publicly broadcast.
Verifi-Encrypt(/rf, PK, M). It takes as inputs the identity string (Id) of the recipient (i.e. Id_{recipient}) and the set of public parameters (PK). The plaintext message M is encrypted as follows: The method first verifies the public parameters (PK) are genuine under the chosen trusted center by comparing \( e(Prv_{sender}, g_{pub}) \) with \( e(g_{sender}^r, g_{admin}^s) \). Recall that for correct values of \( g_{pub} = g_{admin}^s \) and

\[
Prv_{sender} = g_{sender}^r, \quad \text{we have:}
\]

\[
e(Prv_{sender}, g_{pub}) = e(g_{sender}^r, g_{admin}^s) = e(g_{sender}^r, g_{admin}^s)
\]

If a different trusted center is being used, e.g. "another-admin", the verification process proceeds the same by picking the public key of the trusted center, i.e. \( g_{pb, a} \) for "another-admin", and comparing that \( e(Prv_{sender}, g_{pub}) = e(g_{sender}, g_{another-admin}) \).

This verification process is an important step in making sure the correct public parameters (PK) are being used in the encryption. It should be noted that the only fixed parameters are the trusted center's public key \( (g_{pb, a}) \) and the private key of the sender \( (Prv_{sender}) \), which have been obtained from the trusted center (TC). The parameters \( g_{sender} \) and \( g_{admin} \) are generated locally every time the verification is performed using the public parameters (PK) and the identity strings of the sender and the trusted center \( (W^{\wedge} \text{and} \ Id_{TC}) \). If the verification passes, it picks a random symmetric key \( (\sigma) \) and outputs the encrypted ciphertext \( (Q, \text{where} \ C = (V, U, W, Y) \), and where \( V, U, W, \) and \( Y \) are computed as follows:

\[
V = g_{pub}^r, \quad \text{where} \quad r = H_a(a, M)
\]

\[
U = \sigma \oplus H_2(e(g_{sender}^r, g_{admin}^s))
\]
\[ w = \Box_g (M) \]

\[ Y = H_2 \left( e(\check{v}, g_{ld}) \right) \]

It should be noted that, for a sender, the identity string being used (Id) in the above formulas is the identity string of the recipient (Id_{recipient}).

**Auth-Decrypt (Prv_{Id}, PK, C):** It takes as inputs the private key of the recipient (Prv_{ld}, or more specifically, Prv_{recipient}), the public parameters (PK) and the encrypted ciphertext (C), where \( C = (V,U,W,Y) \). If the receiver has the proper private key (Prv_{ld}) under the same trusted domain associated with Id, the algorithm outputs the message M as follows: first calculate \( \sigma = U \circledast H_2 (e(Prv_{ld}, V)) \), where \( Prv_{ld} = g_L^{\frac{1}{\sigma}} \) and then output

\[ M = D_\sigma (W) \]

Note that Fis used to authenticate the sender as follows: The recipient uses M and \( \sigma \) to retrieve r by computing \( r = H_\lambda (\sigma, M) \). Then, the algorithm checks whether or not \( Y = H_2 \left( e(g_{ld}^{\sigma}, Prv_{ld}) \right) \). If yes, then the sender is authenticated and the encrypted ciphertext (C) originated from the claimed sender. Otherwise, the recipient rejects the sender.

Correctness: It is easy to check that decryption correctly recovers M. Note that

\[ v = g_{pub}^r, Prv_{ld} = g_{ld}^{\frac{1}{s}} \text{ and } g_{pub} = g_M^{\sigma} \]. Therefore, we have

\[ e(Prv_{ld}, V) = e(g_{ld}^{\frac{1}{s}}, g_{pub}^r) \]
\[ e\left(g^{\frac{1}{s}}_{id}, g_{ad}^{r}\right) \]

Therefore,
\[ U \oplus H_{2}(e(Pr_{vI}, V)) = \sigma \Theta H_{2}(e(Pr_{vI}, g_{ad}^{r})) \]

\[ H_{2}(e(Pr_{vW}, V)) = \sigma \]

which is then applied to \( D \) to decrypt \( W \) and retrieve \( M \) as required. It is also easy to check that \( s(g_{scvder}, Pr_{vI}^{i}) \) returns \( Y \) for correct values of \( r \) and \( s \) as the following:

\[ e(g_{scvder}, Pr_{vI}^{i}) = e(g_{scvder}, g_{id}^{\frac{1}{s}}) \]

\[ = e(g_{scvder}^{\frac{1}{s}}, g_{id}^{s}) \]

\[ = e(Pr_{vscvder}, g_{id}^{s}) \]

[41] As discussed above, the trusted center 20 is configured to initiate the setup of the certificate-less authenticated encryption (CLAE) system by running the \textbf{Setup (\( \lambda \))} algorithm and manage the distribution of private keys (\( Prv \)) and public parameters (\( PK \)) by responding to \textbf{KeyGen(7d, s)} function calls from different users 30 in the network system 10. The trusted center 20 may itself initiate the \textbf{Setup (\( \lambda \))} algorithm upon startup or when the trusted center 20 determines for itself that it is necessary to renew or reset the security of the network system 10 by generating a new secret master key (s) and new public parameters (\( PK \)). For example, the \textbf{Setup (\( \lambda \))} may be run according to a renewal schedule and the like.
[42] Using the mathematical foundation, described above in a preferred embodiment, users 30 in the network system 10 are able to encrypt and decrypt messages using the CLAE scheme.

[43] Turning now to FIG. 3, the different users 30 in the network system 10 are operable to register with the trusted center 20 and obtain the public parameters (PK) and their respective private keys (Prv). For example, the users 30 may call the KeyGen function 28 made available by the trusted center 20.

[44] The trusted center 20 may also provide a Setup function 26 for a user to call. When called, the Setup function 26 may initiate the Setup (λ) algorithm to renew and/or reset the security of the network system 10 by generating a new secret master key (s) and new public parameters (PK).

[45] Referring now to FIG. 4, a user 30 is operable to register itself with the trusted center 20 by first authenticating itself with the trusted center 20. Different means for authentication may be used to authenticate a user 30 with the trusted center 20, as known in the art. For example, password-based authentication, a challenge-response protocol such as the Kerberos protocol from the Massachusetts Institute of Technology, biometric authentication (i.e. fingerprint, retinal), and the like may be used. Once the user 30 has authenticated itself, the user 30 is operable to call the KeyGen function 28 provided by the trusted center 20 and submitting its identity string (Id) to the trusted center (TC). In FIG. 4, the user 30 is labeled as "Identity String" (i.e. "ID String"). The trusted center (TC) then calls the KeyGen (Id, s) algorithm using the identity string (Id) provided by the user 30 and its own secret master key (s). In return, the trusted center 20 receives the private key for the user 30 (Prv Id STRING) which the trusted center 20 then passes on to the user 30. The
trusted center 20 may also pass on to the user 30, as part of the KeyGen function 28, the public parameters \((PK)\) that are needed to locally generate the identity-based public encryption key of other users based on their identity strings \((Id)\).

[46] Once the user 30 has its respective private key \((Prv)\) and the public parameters \((PK)\), which includes the identity-based public encryption key of the trusted center \((g_p,b)\), a sender is able to send an encrypted message to a recipient without contacting the trusted center 20. No certificate authority is necessary. Instead, the user 30 is able to locally generate the recipient public key \((g_{rec\_recipient})\) using the recipient identity string \((Id_{rec\_recipient})\) known to the user 30 and the public parameters \((PK)\).

[47] Referring now to FIG. 5, a transmission from a user 30A labeled "User A" to a user 30B labeled "User B" is shown in a preferred embodiment. The transmission of an encrypted message may be completed without contacting the trusted center 20. Instead, the identity-based public encryption key of the recipient (i.e. \(g_{key\_B}\)) will be generated locally by the sender (i.e. "User A").

[48] In some embodiments, the CLAE scheme of the present invention may not be efficient for encrypting large files, such as video, audio and other media. In such a situation, it may be desirable to use a conventional symmetric encryption scheme such as the advanced encryption standard (AES), the triple data encryption algorithm (TDEA or 3DES), and the like to transmit the large file. The CLAE scheme of the present invention can then be used to transmit the conventional encryption key \((\Sigma)\) of the conventional symmetric encryption scheme as the plaintext message \((M)\), which is to be protected. In this manner, the file transferred may be symmetrically-encrypted using the conventional
symmetric encryption scheme, with the conventional encryption key ($\Sigma$) transmitted
alongside the encrypted data using the CLAE scheme of the present invention.

[49] Referring now to FIG. 6, a flowchart 600 is shown illustrating the method used by
a user 30A, labeled "User A", for encrypting a message for a user labeled "User B". In
step 610, the sender (i.e. user 30A or "User A") identifies or chooses a trusted center (TC)
to associate the encrypted message with. The sender identifies the trusted center (TC) by
its TC identity string ($Id_{TC}$), i.e. label "TC".

[50] Next, in step 612 the sender determines whether it has the plurality of public
parameters (PK) for the trusted center (TC), including the identity-based public encryption
key of the trusted center ($g_{pu}$) or whether the sender must get the TC's public key prior to
proceeding. Referring briefly to FIG. 7, the sender determines whether or not it needs to
get the TC's public key by comparing whether any of the $g_{pu}$ the sender may have stored
in memory is associated with the identity string of the trusted center ($Id_{rc}$)- If not, user
30A authenticates itself with the trusted center 20 and calls the KeyGen function to
receive its own private key ($Prv_{UserA}$) and the trusted center's public parameters (PK), as
previously described when registering a new user 30 in FIG. 4.

[51] Returning back to FIG. 6, once the sender believes it has the proper public key
$g_{pu}$, the sender in step 614 verifies the TC's public key $g_{pu}$ using the sender's private key
($Prv_{sender}$, $Prv_{UserA}$, or more specifically, $Prv_{UserA}$), as described above with respect to the Verifi-
Encrypt ($Id$, PK, M) algorithm. In particular, the sender calculates $e(Prv_{sender}, g_{pu})$ and
compares it with $e(g_{sender}, g_{admin})$ using the different parameters in the public parameters
(PK) associated with the trusted center 20. As noted above, the TC's public key $g_{pu}$ and
the sender private key ($Prv_{UserA}$, i.e. $Prv_{sender}$) are fixed values received and stored by the
sender from the trusted center (T C). However, the public keys \( g_{sender} \) and \( g_{admin} \) are always generated locally using the public parameter (PK) and the sender identity string (\( Id_{sender} \), i.e. "User A") and the trusted center identity string (\( Id_{rc} \), i.e. "admin"). The sender can be assured that the public parameters (PK) have not been tampered with if

\[
e(Pr_{sc} \cdot g_{pub}) = e(g_{sender} \cdot g_{admin})
\]

is satisfied, as discussed above.

[52] Once the sender has verified the public parameters (PK), the sender can begin by picking a conventional encryption key (27) that is used to encrypt the actual confidential data, such as large documents or video/audio files, using the conventional symmetric encryption scheme (i.e. AES, 3DES and the like). The plaintext message (\( M \)) is then set to include the conventional encryption key (27) of the conventional symmetric encryption scheme to be protected by the CLAE Verifi-Encrypt() algorithm, as shown in step 616. Next, as shown in step 618, the confidential data is symmetrically encrypted using the conventional symmetric encryption scheme and the conventional encryption key (27) is stored as (or as a part of) the plaintext message (\( M \)).

[53] Next, in step 620, the sender computes each component of the ciphertext (\( C \)), as described above with respect to the Verifi-Encrypt(\( Id, PK, M \)) algorithm to symmetrically-encrypt the plaintext message (\( M \)) using the symmetric key encryption function (\( □ \)), found within the public parameters (PK), and using a random symmetric key (\( σ \)), generated locally by the sender. In particular, each of V, U and W are computed for the particular trusted center (T C) using the recipient identity string (\( Id_{recipient} \), i.e. "User B") associated with the recipient, the random symmetric key (\( σ \)), which is randomly generated every time Verifi-Encrypt(\( /d, PK, M \)) is run, and the identity-based public encryption key.
of the trusted center \((g_{pA})\), as well as the other public parameters in the public parameters \((PK)\).

[54] In step 622, the encrypted message is signed by returning the ciphertext component \(Y\) for the recipient using the sender private key \(Prv_{send}\) or more specifically, \(Prv_{UserA}\). The recipient may be operable to use this ciphertext component \(Y\) to authenticate the received message.

[55] Finally, the ciphertext \((C)\) is packaged together in step 624 and is ready to be attached to the transmission to be sent to the recipient. The components of the ciphertext \((C)\) may be packaged as a file (i.e. "cip.key") or within other methods and/or structures known in the art. The symmetrically-encrypted data using the conventional symmetric encryption scheme along with the file, "cip.key", containing the conventional encryption key \((27)\) stored in the plaintext message \((M)\) is then ready for transmission to the recipient and may be sent over an unsecured channel as a properly encrypted message.

[56] Upon receipt of the ciphertext \((C)\), i.e. as stored in the file "cip.key", the recipient is operable to decrypt the ciphertext \((C)\) to retrieve the plaintext message \((M)\) that contains the conventional encryption key \((27)\) used to encrypt the actual data. Referring now to Fig. 8, a flowchart 800 is shown illustrating a user 30B, labeled "User B", decrypting a message from a user labeled "User A". In step 810, the recipient (i.e. user 30B or "User B") retrieves the ciphertext \((C)\) from the received transmission. For example, the ciphertext \((C)\) may be packaged as the file, "cip.key".

[57] Next, the recipient determines, at step 812, whether it needs to get the recipient private key \(Prv_{recipient}\) or more specifically, \(Prv_{UserB}\) from the trusted center \((TC)\). Referring briefly to FIG. 9, the recipient determines whether or not it needs to get the
recipient private key \((Prv_{User B})\) of the particular trusted center (TC) by comparing whether any of the \(Prv_{User B}\) the recipient may have stored in memory is associated with the identity string of the trusted center \((Idrc, \text{ i-e. } "admin")\). If not, user 30B authenticates itself with the trusted center 20 and calls the KeyGen() function to receive its own private key \((Prv_{User B})\) and the trusted center’s public key \(g_{p,r,b}\), as previously described when registering a new user 30 in FIG. 4. At this time, the recipient may also receive an updated set of public parameters \((PK)\) from the trusted center (TC).

[58] Returning to FIG. 8, the recipient is now able to decrypt the different components \(V, U \text{ and } W\) of the ciphertext \((C)\) using the recipient private key \((Prv_{User B})\) and the public parameters \((PK)\) in step 814. From these components, the recipient is able to retrieve the conventional encryption key \((27)\) stored in the plaintext message "M" and "r", as described in the Auth-Decrypt \((Prv_{Id}, PK, C)\) algorithm. Furthermore, in step 816, the recipient is operable to verify the sender of the message by inspecting the ciphertext \((C)\) component \(Y\).

As described in the Auth-Decrypt \((Prv_{Id}, PK, C)\) algorithm, the sender may be authenticated by calculating \(Y = H_2(e(Prv_{sender}, g_{napi, Id})^r)\). Therefore, because \(e(g_{sender}, Prv_{recipient}) \equiv e(Prv_{sender}, g_{napi, Id})\), the recipient is operable to verify the sender if \(Y = H_2(e(g_{sender}, Prv_{recipient})^r)\) is satisfied, where \(g_{sender}\) is generated locally from the sender identity string \((Id_{sender})\).

[59] Finally, once the sender has been verified, the recipient is operable to restore the plaintext message \((M)\) that contains the conventional encryption key \((27)\) used by the conventional symmetric encryption scheme to encrypt the confidential data. The plaintext message \((M)\) is restored using the random symmetric key \((\sigma)\) and the symmetric key encryption function \(\Box\) to easily determine \(D\), as described above in the Auth-Decrypt.
(Prvi_{i,d}, PK, C) algorithm. Using the conventional encryption key \( \Sigma \) stored in the plaintext message \( M \), the recipient is operable to decrypt the transmitted message using the conventional symmetric encryption scheme. The decryption process is now complete.

[60] The CLAE scheme as described above in a preferred embodiment of the present invention allows for the intended recipient to verify the sender's identity without storing or referring to any public key certificates. If the recipient successfully decrypts the ciphertext \( C \), the recipient can authenticate the sender based on the properties of the bilinear map \( (e) \) using the public parameters \( PK \) and the sender identity string \( (Id_{sender}) \). The authentication is an integral part of the CLAE scheme and it is more efficient to verify the sender in this manner than separately adding additional authentication components to the encryption process. The CLAE scheme according to the preferred embodiment of the present invention will ensure that not only is the sensitive data kept confidential, but also the sender of the confidential data is authentic. It should be noted that the authentication can be made optional by removing \( Y \) from the ciphertext \( (C) \) in an application where other authentication/digital signature schemes exist.

Dynamic Authority

[61] The encryption key in CLAE scheme according to the preferred embodiment of the present invention is derived from dynamic parameters that are calculated from the trusted center's identifier, e.g. domain name, phone number, etc.

[62] This new design yields greater flexibility in working with multiple authorities. The sender of encrypted data can enforce elaborate access conditions on the recipient before the recipient can decrypt and retrieve sensitive data. The sender can select not only who the recipient is but also how the recipient receives its private key \( (Prv_{recipient}) \). For example,
a descriptive string may be combined or appended with the TC identity string (Idrc) to increase the level of authentication required by the trusted center (TC) necessary for the recipient to obtain its private key (Prv_{rec,TC}) from the trusted center (TC). The recipient may be forced by the trusted center (TC) to further authenticate itself by satisfying this additional condition, provided by the descriptive string. The additional descriptive string may include the recipient's role, the recipient's age, the recipient's location, an expiry date and the like.

[63] As an example, the sender can choose "bob@abc.com" as the recipient's identity of the encrypted message and can set "abc.com — December-2013" as the trusted center's public identity string (TCy) with an expiry date of December 2013. The sender's description of the TC identity string (TCid) is then enforced in the ciphertext (C) by locally computing a new $g_{admin-new}$, where $g_{a,mm-new} = H ("abc.com — December-2013")$. If the sender possesses the public key of the trusted center (TC) $g_{pub} = g_{a,mm-new}$ corresponding to the new $g_{a,mm-new}$, it can proceed with Verifi-Encrypt(), as described above. Otherwise, the sender is forced to obtain a new $g_{a,b}$, as shown with respect to FIG. 7.

[64] Beyond this date, the encryption algorithm would receive a new public key from the trusted center and would use it to generate new encryption keys. The recipient would then be forced to renew its private key from trusted center. This is very useful in conditions where the identity of users in the system will remain the same for an extended period, but it is beneficial for security reasons to have the private keys update periodically and/or frequently. On the receiver's side, if the same secret master key (s) is used to compute the new identity-based public encryption key of the trusted center $g_{pub-new}$, no change has to be made to the private key of the recipient (Prv_{rec,TC}) or the decryption
algorithm. If, however, a different secret master key (s) is used, the recipient will be forced to receive a new private key corresponding to the secret master key of the trusted center (TC) with the new TC identity string (Idrc-new), as shown in FIG. 9.

[65] It should be noted that the same encryption algorithm can be used if the sender decides to use a different server (trusted authority) such as "xyz.com" instead of "abc.com". The only difference in the encryption algorithm would be in using the identity-based public encryption key of the new trusted center (g_ρφ) of the new trusted center and everything else remains the same.

Exchange of Trust

[66] As discussed above, the new CLAE scheme allows the sender to communicate privately with any recipient under various trusted authorities. For example, the sender can send an encrypted message from the "abc.com" domain (with secret master key s) to someone in the "xyz.com" domain (with secret master key s'). Two different cases can happen depending if s = s'.

a. If s = s', the sender uses the public key of the new trusted center (s') in the encrypt. Everything else in the encryption proceeds exactly the same as before with the new public key (s'). No further changes are required in the decryption algorithm and the intended recipients decrypts the ciphertext as before using the same private key (Prv_{Id}).

b. If s ≠ s', the sender uses the public key of the new trusted center (s') in the encrypt. Everything else in the encryption proceeds exactly the same as before with the new public key (s'). The intended recipient is now forced to
authenticate itself to the new trusted center, picked by the sender, in order to receive the corresponding new private key $Pr_{w}$. No further changes are required in the decryption algorithm and the intended recipients decrypts the ciphertext $(C)$ as before using the new private key $Pr_{w}$.

[67] In this manner, the sender has control of which trusted center $(TC)$ to associate with and can force the recipient to authenticate to a trusted center $(TC)$ of its choosing. Accordingly, the sender can rely on the additional security of choosing its preferred trusted center to generate and supply the private key $(Pr_{td})$ to the recipient. The onus can then be placed on the recipient to associate with reliable trusted centers and authenticate itself to the trusted center $(TC)$, selected by the sender, to receive the private key for the recipient $(Pr_{recipient})$.

[68] Although this disclosure has described and illustrated certain preferred embodiments of the invention, it is also to be understood that the invention is not restricted to these particular embodiments rather, the invention includes all embodiments which are functional, or mechanical equivalents of the specific embodiments and features that have been described and illustrated herein. The scope of the claims should not be limited to the preferred embodiments set forth in the examples, but should be given the broadest interpretation consistent with the description as a whole.

[69] It will be understood that, although various features of the invention have been described with respect to one or another of the embodiments of the invention, the various features and embodiments of the invention may be combined or used in conjunction with other features and embodiments of the invention as described and illustrated herein. Furthermore, while the various methods described herein may refer to a specific order and
number of steps, it should be understood that the order and/or number of method steps described herein should not be construed as limiting, as other orders and/or number of steps would be understood by persons skilled in the art.
CLAIMS:

The embodiments of the invention in which an exclusive property or privilege is claimed is defined as follows:

1. A method of sending an encrypted message by a sender having a sender identity string (Id_{sender}) to a recipient over a network using identity-based encryption, the method comprising:

   identifying a trusted center (TC) by a TC identity string (Id_{tc}), the trusted center having an identity-based public encryption key of the trusted center \((g_{pa},b)\) based on the TC identity string (Id_{tc});

   determining if the sender has a sender private key \((Prv_{sen},d_{sen})\) and a plurality of public parameters \((PK)\) for the trusted center (TC), the public parameters \((PK)\) including the identity-based public encryption key of the trusted center \((g_{pa},b)\) and a bilinear map \((<\cdot>)\);

   verifying the public parameters \((PK)\) of the trusted center (TC) using the TC identity string (Id_{tc}) prior to encrypting a plaintext message \((M)\);

   identifying the recipient by a recipient identity string (Id_{rc,ident}). the recipient having an identity-based public encryption key of the recipient \((g_{rc,recipient})\) for the trusted center (TC) based on the recipient identity string (Id_{rc,ident});

   generating the identity-based public encryption key of the recipient \((g_{rc,ident})\) using the public parameters \((PK)\) and the recipient identity string (Id_{rc,ident});

   encrypting the plaintext message \((M)\) as ciphertext \((C)\) using the public parameters \((PK)\), a random symmetric key \((\sigma)\) and the identity-based public encryption key of the recipient \((g_{rc,ident})\), the ciphertext \((C)\) including the encrypted message; and
transmitting the ciphertext (C) to the recipient over the network.

2. The method of claim 1, wherein verifying the public parameters (PK) comprises comparing values of the bilinear map (e) calculated with variables comprising the sender private key (Prv_{sende}) and the identity-based public encryption key of the trusted center (gpub).

3. The method of claim 2, wherein the public parameters (PK) are verified if

\[ e^{Prv_{sende}}(S_{pub}) = e^{g_{sende}}(ld_{sende}) \]

where \( g_{sende} \) is based on the sender identity string (ld_{sende}) and \( g_{ld} \) is based on the TC identity string (ld_{rc}).

4. The method of claim 3, wherein the sender private key (Prv_{sende}) and the identity-based public encryption key of the trusted center (g_{pub}) are provided by the trusted center (TC) and \( g_{sende} \) and \( g_{ld} \) are generated locally by the sender during the verifying step.

5. The method of claim 1, wherein the public parameters (PK) further include a first cryptographic hash function (H_1), a second cryptographic hash function (H_2), a third cryptographic hash function (H_3), a symmetric key encryption function (□), and G and G_r, where G and G_r are groups of order p and where the bilinear map (e) satisfies:

\[ e : G \times G \rightarrow G_1 \]

6. The method of claim 1, wherein the ciphertext (C) includes an authentication component (Y) for authenticating the sender once the ciphertext (C) is received by the recipient.

7. The method of claim 6, wherein the authentication component (Y) is based on the sender private key (Prv_{sende}) and the identity-based public encryption key of the recipient (g_{recipien}), and wherein the recipient is operable to verify the sender using the public
parameters (PK), the sender identity string (Id_{sender}), and a recipient private key (Prv_{recipient}) obtained from the trusted center (TC).

8. The method of claim 1, wherein the sender can specify an expiry date for the identity-based public encryption key of the trusted center (g_{pub}), whereby after the expiry date, the recipient is forced to authenticate with the trusted center (TC) to obtain a new recipient private key (Prv_{recipient,new}).

9. The method of claim 1, wherein the sender can specify a descriptive string to append to the TC identity string (Id_{rc}), whereby the descriptive string is used by the trusted center (TC) to require additional levels of authentication from the recipient in order for the trusted center (TC) to provide a recipient private key (Prv_{recipient}) to the recipient.

10. The method of claim 9, wherein the descriptive string is selected from the group consisting of: a role of the recipient, an age of the recipient, a location of the recipient and an expiry date.

11. The method of claim 1, wherein the plaintext message (M) is a conventional encryption key.

12. A method for using certificate-less encryption (CLAE) using identity-based encryption in a network system between a sender having a sender identity string (Id_{sender}) and a recipient having a recipient identity string (Id_{recipient}), the method comprising:

   at the sender:

   identifying a trusted center (TC) by a TC identity string (Id_{rc}), the trusted center having an identity-based public encryption key of the trusted center (g_{pub}) based on the TC identity string (Id_{rc});

determining if the sender has a sender private key (Prv_{sender}) and a plurality of public parameters (PK) for the trusted center (TC), the public parameters
(PK) including the identity-based public encryption key of the trusted center (gpub) and a bilinear map (e);

verifying the public parameters (PK) of the trusted center (TO) using the TC identity string (Id TC) prior to encrypting a plaintext message (M);

identifying the recipient by a recipient identity string (Id recipient), the recipient having an identity-based public encryption key of the recipient (grecipient) for the trusted center (TC) based on the recipient identity string (Idrecipient);

generating the identity-based public encryption key of the recipient (grecipient) using the public parameters (PK) and the recipient identity string (Idrecipient).

encrypting the plaintext message (M) as ciphertext (C) using the public parameters (PK), a random symmetric key (σ) and the identity-based public encryption key of the recipient (grecipient), the ciphertext (C) including the encrypted message; and

transmitting the ciphertext (C) to the recipient over the network; and

at the recipient:

receiving the ciphertext (C) from the sender over the network system;

determining if the recipient has a recipient private key (Prvrecipient) and the public parameters (PK) for the trusted center (TC);

decrypting the ciphertext (C) to obtain the plaintext message (M) using the public parameters (PK), and the recipient private key (Prvrecipient).
13. The method of claim 12, wherein the method further comprises, at the recipient, verifying the sender using the ciphertext (C), the public parameters (PK), the sender identity string (Id_{sender}), and the recipient private key (Prv_{recip}).

14. A method of verifying a plurality of public parameters (PK) from a trusted center (TC) in an identity-based encryption system prior to encrypting a plaintext message (M) by a sender having a sender identity string (Id_{sender}), the method comprising:

- identifying the trusted center (TC) by a TC identity string (Id_{tc}), the trusted center having an identity-based public encryption key of the trusted center (g_p) based on the TC identity string (Id_{tc});

- determining if the sender has a sender private key (Prv_{sender}) and the plurality of public parameters (PK) for the trusted center (TC), the public parameters (PK) including the identity-based public encryption key of the trusted center (g_p) and a bilinear map (e); and

- verifying the public parameters (PK) of the trusted center (TC) using the TC identity string (Id_{tc}) prior to encrypting the plaintext message (M) by comparing values of the bilinear map (e) calculated with variables comprising the sender private key (Prv_{sender}) and the identity-based public encryption key of the trusted center (g_p).

15. A system for sending an encrypted message over a network using identity-based encryption, the system comprising:

- a trusted center (TC) having a TC identity string (Id_{tc}), a sender having a sender identity string (Id_{sender}), and a recipient having a recipient identity string (Id_{recip});

wherein the trusted center (TC) has a first memory and one or more processors configured for:
generating a plurality of public parameters \((PK)\) and a secret master key \((s)\) from a security parameter \((\lambda)\), the public parameters \((PK)\) including a bilinear map \((e)\) and an identity-based public encryption key of the trusted center \((g_{\rho,ab})\) based on the TC identity string \((Idrc)\);

receiving a request from a requestor;

if the request from the requestor contains an identifier \((Id)\) identifying the requestor, generating a private key \((Prv_{Id})\) based on the identifier \((Id)\) and the secret master key \((s)\); and transmitting the private key \((Prv_{Id})\) to the requestor over the network system; and

if the request from the requestor includes a request for the public parameters \((PK)\), transmitting the public parameters \((PK)\) to the requestor over the network system;

wherein the sender has a second memory and one or more processors configured for:

identifying the trusted center \((TC)\) by the TC identity string \((Idrc)\),

determining if the sender has a sender private key \((Prv_{sender})\) and the public parameters \((PK)\) for the trusted center \((TC)\),

verifying the public parameters \((PK)\) of the trusted center \((TC)\) using the TC identity string \((Idrc)\) prior to encrypting a plaintext message \((M)\);

identifying the recipient by the recipient identity string \((Id_{recipient})\) \((Id_{recipient})\), the recipient having an identity-based public encryption key of the recipient \((g_{recipient,b})\) \((g_{recipient,b})\) the trusted center \((TC)\) based on the recipient identity string \((Id_{recipient})\)\).

40
generating the identity-based public encryption key of the recipient
\( (g_{rec} \cdot k_{ed}) \) using the public parameters \((PK)\) and the recipient identity string
\((Id_{recipient})\);

encrypting the plaintext message \((M)\) as ciphertext \((C)\) using the public parameters \((PK)\), a random number \((\sigma)\) and the identity-based public encryption key of the recipient \((g_{recipient})\), the ciphertext \((C)\) including the encrypted message; and

transmitting the ciphertext \((C)\) to the recipient over the network; and

wherein the recipient has a third memory and one or more processors configured for:

- receiving the ciphertext \((C)\) from the sender over the network system;
- determining if the recipient has a recipient private key \((Prv_{recipient})\) and the public parameters \((PK)\) for the trusted center \((TC)\); and
- decrypting the ciphertext \((C)\) to obtain the plaintext message \((M)\) using the public parameters \((PK)\), and the recipient private key \((Prv_{recipient})\).

16. The system of claim 15, wherein the plurality of public parameters \((PK)\) further include the identity-based public encryption key of the trusted center \((g_{public})\) based on the TC identity string \((Id_{TC})\), a first cryptographic hash function \((H_0)\), a second cryptographic hash function \((H_1)\), a third cryptographic hash function \((H_2)\), a symmetric key encryption function \((\oplus)\), and \(G\) and \(G_r\), where \(G\) and \(G_r\) are groups of order \(p\) and where the bilinear map \((e)\) satisfies:

\[ e : G \times G \rightarrow G_r \]
17. A computer program product comprising a computer readable memory storing computer executable instructions thereon that, when executed by a computer, perform the method steps of:

identifying a trusted center (TC) by a TC identity string (Id_{TC}), the trusted center having an identity-based public encryption key of the trusted center (g_{pub}) based on the TC identity string (Id_{TC});

determining if a sender has a sender private key (Prv_{s,der}) and a plurality of public parameters (PK) for the trusted center (TC), the public parameters (PK) including the identity-based public encryption key of the trusted center (g_{pub}) and a bilinear map (e);

verifying the public parameters (PK) of the trusted center (TC) using the TC identity string (Id_{TC}) prior to encrypting a plaintext message (M);

identifying a recipient by a recipient identity string (Id_{recipient}), the recipient having an identity-based public encryption key of the recipient (g_{pub}) for the trusted center (TC) based on the recipient identity string (Id_{recipient});

generating the identity-based public encryption key of the recipient (g_{pub}) using the public parameters (PK) and the recipient identity string (Id_{recipient});

encrypting the plaintext message (M) as ciphertext (C) using the public parameters (PK), a random number (\sigma) and the identity-based public encryption key of the recipient (g_{pub}), the ciphertext (C) including an encrypted message based on the plaintext message (M); and

transmitting the ciphertext (C) to the recipient over a network.
Identify a trusted center (TC) by a TC identity string (Id_{TC})

Determine if the sender has a sender private key (Prv_{sender}) and a plurality of public parameters (PK) for the trusted center (TC)

Verify the public parameters (PK) of the trusted center (TC) prior to encrypting a plaintext message (M)

Identify the recipient by a recipient identity string (Id_{recipient})

Generate the identity-based public encryption key of the recipient (g_{recipient})

Encrypt the plaintext message (M) as ciphertext (C)

Transmit the ciphertext (C) to the recipient over a network

FIG. 2
Identity String:

10

Register with TC

20

Setup

26

KeyGen

28

Continue

Call KeyGen for new PrvD String, and 'gpub'
User A:

Encrypt for "User B"

Choose TC with "TC"

Get TC public key?

Verify TC's 'g_{pub}' using 'Prv_{User A}'

Set a conventional encryption key 'Σ' as plaintext message 'M'

Symmetric-encrypt data with 'M' (i.e. 'Σ')

Compute 'V', 'U', 'W' for "TC" and "User B" using σ, 'M' and 'g_{pub}'

Sign by returning 'Y' for "User B" using 'Prv_{User A}'

Attach 'V', 'U', 'W', 'Y' as "cip.key"

FIG. 6
User B:

810
Retrieve "cip.key"

812
Get 'PrvUserB' from TC?

814
Decrypt 'V', 'U', 'W' to retrieve 'M' and 'r' using 'PrvUserB'.

816
Verify "User A" using 'PrvUserB' and 'r'.

818
Symmetric-decrypt data with 'M' (i.e. 'Σ').

FIG. 8
INTERNATIONAL SEARCH REPORT

International application No. PCT/CA2013/000103

A. CLASSIFICATION OF SUBJECT MATTER
IPC: H04L 9/32 (2006.01), H04L 12/58 (2006.01), H04L 9/30 (2006.01)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
IPC: H04L* (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 2010/0208995 A1 (BONEH et al.), 19 August 2010 (19-08-2010) *see para. [0039-0044] and Fig. 1</td>
<td>1-17</td>
</tr>
<tr>
<td>A</td>
<td>US 7,239,701 B1 (OGISHJ et al.), 3 July 2007 (03-07-2007) **see entire document</td>
<td>1-17</td>
</tr>
<tr>
<td>A</td>
<td>Tao Du; Shouning Qu; Guangyu Zheng; Jingwen Xu; Linjiang Dong. &quot;An Efficient Certificateless E3-based Signcryption Scheme Based on Pair Public Key,&quot; Networked Computing (INC), 2010 6th International Conference on, pp. 1-6, 11-13 May 2010 (13-05-2010) *see entire document</td>
<td>1-17</td>
</tr>
</tbody>
</table>

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

Date of the actual completion of the international search
2 April 2013 (02-04-2013)

Date of mailing of the international search report
16 April 2013 (16-04-2013)

Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 001-819-953-2476

Authorized officer
Jamie Hayami (819) 934-2670
## DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patent Document</td>
<td>Publication Date (19-08-2010)</td>
<td>Patent Family</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>US2010208895A1</td>
<td>19 August 2010</td>
<td>AT465571T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2002332671A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE60236094D1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP1425874A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP1425874A4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP1425874B1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP2224637A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP2224637A3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP2429116A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP2429116A3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP2005500740A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP2011151866A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP2012032843A</td>
</tr>
<tr>
<td>WO20040759A3</td>
<td>10 July 2003</td>
<td>WO20040759A3</td>
</tr>
</tbody>
</table>