SYSTEM AND METHOD FOR COMPUTING COMMON SESSION KEYS IN A FORWARD SECURE IDENTITY-BASED AUTHENTICATED KEY EXCHANGE SCHEME

Field of the Invention

This invention relates to a system and method for generating common session keys that each have strong forward security for encoding digital communications between devices. In particular, the system utilizes a forward secure identity-based authenticated key exchange scheme to allow two devices to verify the veracity of each device before these authenticated devices proceed to generate a common session key that is then utilized to encode digital communications between these two devices.

Summary of the Prior Art

Due to a convergence of multiple technologies, an ever increasing number of devices are now able to seamlessly communicate wirelessly with the Internet or wirelessly exchange communications between themselves. This convergence has resulted in the vision of the Internet of Things (IoT) gaining more traction through recent years. In the Internet of Things, millions of entities or devices (i.e. Things) will be connected to one another. These devices, which comprise devices or entities such as smart chips, smart plugs, smart watches, smart phones, smart vehicles, smart buildings, and etc., either communicate directly with one another or via the Internet.

As the Internet of Things spreads into more areas, these devices become more prone to cyber-attacks from hackers or unauthorized users as a compromised device would grant a malicious user access to the network to which the device belongs. Hence, it is of utmost importance that a security protocol be set in place to allow one entity to verify the authenticity of a peer entity, with which it communicates with, before data is exchanged between these two entities. In addition to the above, once the authenticity of the entities have been verified, a common secret key known to only these two entities, has to be established to facilitate the signing of data messages between these entities. This is to ensure that data communicated between these two entities will not be compromised even if the data were intercepted or redirected.

A common method of establishing a secret key for signing data communications between entities involves the pre-sharing of a common symmetric secret key between the relevant parties. For such a method, data messages that are to be transmitted between the entities will be signed using this pre-shared or pre-agreed-upon secret key. If the receiving

entity is able to decrypt the received message using this pre-shared secret key, this implies that the authenticity of the sender has been verified and the receiver may then proceed to process the decrypted message accordingly. Unfortunately, this method is not scalable and is quite inflexible as it requires the common secret to be pre-shared or communicated to all trusted entities or devices before the entities or devices may communicate with one another. In the IoT setting, the high mobility of devices is a norm and devices that are required to exchange data with one another may not have had the opportunity to establish a secret key beforehand.

Another approach that has been proposed utilizes public key infrastructure (PKI) based solutions whereby key-pairs allocated to each authorized entity are bound to its holders by means of a public key certificate. The key pair then utilizes a public key cryptosystem such as public-key encryption or digital signature methodologies to sign data messages or to verify the authenticity of a sender by validating the public key certificate of the sender. The setup and maintenance of such public key infrastructures are notoriously expensive and requires entities to constantly maintain contact with a PKI server to validate the respective public key certificates.

Yet another approach that has been proposed utilizes identity based cryptography methodologies to authenticate entities and to sign data messages. Such identity based cryptosystems are special public key cryptosystems, which are based on bilinear pairing and utilize an entity's identity, such as user name, email address, telephone number, IP address, etc. as the public key and a corresponding private key is then derived from the entity's identity by a Key Generation Centre (KGC) which contains a master secret key, which is utilized in the generation of private keys for entities.

It is quite challenging for identity based signature schemes having authenticated key exchange protocols to possess both strong forward security and escrow of session keys to an approved KGC. As a result, those skilled in the art have proposed various schemes to address this.

One of the approaches proposed by those skilled in the art involves a tailored construction, which does not directly make use of the signing or encryption functionality of identity-based cryptography to authenticate entities and to exchange keys. In this tailored construction, escrow of the common session keys is achieved because the Key Generation Centre (KGC) is able to compute the users' private keys that were generated by the Key Generation Centre. The downside to this approach is that only weak forward security is

achievable. Another method that has been proposed utilizes an explicit escrow approach whereby the escrow agent (i.e. the KGC) is configured to have an extra key pair for public key encryption such that the common session keys are encrypted under the escrow agent's public key and in transmission together with other key exchange messages between users. The downside to this approach is that it is difficult for any party (except the encryptor) to check whether the escrowed object is indeed an encryption of the secret session key to be established.

For the above reasons, those skilled in the art are constantly striving to come up with a system and method to generate common session keys for users of an identity based signature scheme whereby the generated common session keys have strong forward security and the session keys must be able to be computed by the KGC when required.

Summary of the Invention

The above and other problems are solved and an advance in the art is made by systems and methods provided by embodiments in accordance with the invention.

A first advantage of embodiments of systems and methods in accordance with the invention is that the common session keys generated in accordance with the invention are able to achieve strong forward security. Further, the invention allows the generated common session keys to be escrowed to an authorized Key Generation Centre.

A second advantage of embodiments of systems and methods in accordance with the invention is that the invention may be executed using general identity based authenticated key exchange protocols by utilizing the signature signing functionality of such identity based signature schemes for entity authentication and key exchange.

A third advantage of embodiments of systems and methods in accordance with the invention is that the escrow of the generated common session keys is not dependent on the escrow of the users' private keys to the KGC. In other words, the KGC can compute the common session key by gathering all communications that took place between users of key exchange, and the KGC does not necessarily know the users' private keys (in fact, in certain identity based signature schemes, a user's private key is jointly generated by the KGC and the user, so the KGC itself does not learn the user's private key).

A fourth advantage of embodiments of systems and method in accordance with the invention is that an implicit escrow of the common session keys takes place without the need

for an extra key pair to be maintained and stored at the KGC, and the escrow object that enables the KGC to compute the escrowed session key is verifiable to the KGC.

The above advantages are provided by embodiments of a method in accordance with the invention operating in the following manner.

According to a first aspect of the invention, a system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: a secure server configured to: generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*; generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device; communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; the first device configured to: compute a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*; communicate to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*; compute signature σ_j by signing the first element k_i and the second element k_j using a signing function of an Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using a verification function of the Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the first aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the signature σ_i to the second device, whereby the system further comprises: the second device being configured to: verify the signature σ_i using the verification function of the Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, compute a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*, and generate the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

With reference to the first aspect, in accordance with embodiments of the invention, the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = b \cdot P$; and the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* defined over the elliptic curve and cs_i is defined as $cs_i = e(b \cdot P, s \cdot P)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot P)^b$.

With reference to the first aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; compute a common-secret cs_{kgc} based on the first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot P)^s$.

With reference to the first aspect, in accordance with embodiments of the invention, the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and the common-secret cs_i and the common secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where G is cyclic group with prime order q defined over the elliptic curve, and cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

With reference to the first aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; and compute a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* and the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot Q)^s$.

According to a second aspect of the invention, a system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: a secure server configured to: generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*; generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device; communicate the master public

key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; the first device configured to: compute a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i; communicate to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i; compute a signature σ_j by signing a part of the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing a part of the second element k_j and a part of the private key sk_j as combined with the random value a using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; compute a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the second aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the part of the private key sk_j as combined with the random value a and the signature σ_i to the second device, whereby the system further comprises: the second device being configured to: verify the signature σ_i using the verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, compute a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, a part of the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( ); and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function.

With reference to the second aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where g is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and $H()$ is a

collision-resistant hash function; the first element k_i is defined as $k_i = (R_i, g^a)$; the second element k_j is defined as $k_j = ({R_i}^b, g^b)$; the two input function f( )comprises a concatenation function or an exclusive-OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j, R_j)}, {(g^b)^{s_i}}/{{R_i}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i, R_i)}, {(g^a)^{s_j}}/{{R_j}^a})$.

With reference to the second aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; and compute a common-secret cs_{kgc} by providing the master secret key *msk*, a part of the first element k_i, the identity of the second device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

According to a third aspect of the invention, a system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: a secure server configured to: generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*; generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device; communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; the first device configured to: compute a first element k_i based on a first random value *a* generated by the first device; communicate to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device; compute a second element k_j based on the element u_j and the second random value *b*; generate a signature σ_j by signing the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute a signature σ_i by signing a part of the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; compute an element u_i based on the part of the second element k_j and the first random value a; compute a common-secret cs_i based on

the element u_i and the master public key *mpk*; and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the third aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the signature σ_i to the second device, whereby the system further comprises: the second device being configured to: verify the signature σ_i using the verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, compute a common-secret cs_j based on the element u_j and the master public key *mpk;* and generate the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

With reference to the third aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = g^a$; the second element k_j is defined as $k_j = (U, g^b)$ where *U* is defined as $= g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$; the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

With reference to the third aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; compute a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein cs_{kgc} is defined as $cs_{kgc} = U^x$.

According to a fourth aspect of the invention, a system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: a secure server configured to: generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*; generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device; communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; the first device configured to: compute a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*; compute signature σ_i by signing the first element k_i using a signing function of an Identity

Based Signature Scheme and the private key sk_i; communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the Identity Based Signature Scheme and the identity of the first device id_i; compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*, when the signed first element k_i is verified; compute a signature σ_j by signing the first element k_i and the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using the verification function of the Identity Based Signature Scheme and the identity of the second device id_j and compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*, when the signature σ_j is verified; generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the fourth aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the verification data vd_i to the second device, whereby the system further comprises: the second device being configured to: compute a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*; generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determine if the verification data vd_i matches with the verification data vd_j; and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the fourth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = b \cdot P$; and the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map $e: G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* and where cs_i is defined as $cs_i = e(s \cdot P, b \cdot P)^a$ and cs_j is defined as $cs_j = e(s \cdot P, a \cdot P)^b$.

With reference to the fourth aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; compute a common-secret cs_{kgc} based on the first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.P)^s$.

With reference to the fourth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q*, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and the common-secret cs_i and the common secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

With reference to the fourth aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; compute a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.Q)^s$.

According to a fifth aspect of the invention, a system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: a secure server configured to: generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*; generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device; communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; the first device configured to: compute a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i; compute a signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i; communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i, when the signature σ_i is verified; compute a signature σ_j by signing a part of the first element k_i and the

second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature $\sigma_{j;}$ verify the signature σ_j using the verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j; compute a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ), when the signature σ_j is verified; generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the fifth aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the verification data vd_i and the part of the private key sk_j as combined with the random value a to the second device, whereby the system further comprises: the second device being configured to: compute a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( ); generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determine if the verification data vd_i matches with the verification data vd_j; and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the fifth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and *H*() is a collision-resistant hash function; the first element k_i is defined as $k_i = (g^a)$;the second element k_j is defined as $k_j = ({R_i}^b, g^b)$; the two input function f( )comprises a concatenation function or an exclusive-

OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j,R_j)}, {(g^b)^{s_i}}/{R_i{}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i,R_i)}, {(g^a)^{s_j}}/{R_j{}^a})$.

With reference to the fifth aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; and compute a common-secret cs_{kgc} by providing the master secret key *msk*, the first element k_i, the identity of the second device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

According to a sixth aspect of the invention, a system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: a secure server configured to: generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*; generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device; communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; the first device configured to: compute a first element k_i based on a first random value *a* generated by the first device; compute signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i; communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device, when the signature σ_i is verified; compute a second element k_j based on the element u_j and the second random value *b*; compute signature σ_j by signing the first element k_i and the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using the verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute an element u_i based on a part of the second element k_j and the first random value a, when the signature σ_j is verified; compute a common-secret cs_i based on

the element u_i and the master public key *mpk*; generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the sixth aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the verification data vd_i to the second device, whereby the system further comprises: the second device being configured to: compute a common-secret cs_j based on the element u_j and the master public key *mpk*; generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determine if the verification data vd_i matches with the verification data vd_j; and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the sixth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where g is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = g^a$; the second element k_j is defined as $k_j = (U, g^b)$ where U is defined as $= g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$; the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

With reference to the sixth aspect, in accordance with embodiments of the invention, the secure server is further configured to: receive all communications and transmissions exchanged between the first and second devices; compute a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = U^x$.

According to a seventh aspect of the invention, a method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises the steps of generating, by a secure server, a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key

sk_j to the second device; computing, by the first device, a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*; communicating, by the first device, to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*; compute signature σ_j by signing the first element k_i and the second element k_j using a signing function of an Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verifying, by the first device, the signature σ_j using a verification function of the Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; computing, by the first device, a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and generating, by the first device, the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the seventh aspect, in accordance with embodiments of the invention, the first device further communicates the signature σ_i to the second device, and the method further comprises: verifying, by the second device, the signature σ_i using the verification function of the Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, computing a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*, and generating the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

With reference to the seventh aspect, in accordance with embodiments of the invention, the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = b \cdot P$; and the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map $e: G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* defined over the elliptic curve and cs_i is defined as $cs_i = e(b \cdot P, s \cdot P)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot P)^b$.

With reference to the seventh aspect, in accordance with embodiments of the invention, the secure server further: receives all communications and transmissions exchanged between the first and second devices; and computes a common-secret cs_{kgc} based on the

first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot P)^s$.

With reference to the seventh aspect, in accordance with embodiments of the invention, the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and the common-secret cs_i and the common secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where G is cyclic group with prime order q defined over the elliptic curve, and cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

With reference to the seventh aspect, in accordance with embodiments of the invention, the secure server further receives all communications and transmissions exchanged between the first and second devices; and computes a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* and the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot Q)^s$.

According to an eighth aspect of the invention, a method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises the steps of generating, by a secure server, a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; computing, by the first device, a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i; communicating to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i; compute a signature σ_j by signing a part of the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verifying, by the first device, the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and computing signature σ_i by signing a part of the second element k_j and a part of the private key sk_j as combined

with the random value a using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; computing a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ); and generating the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the eighth aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the part of the private key sk_j as combined with the random value a and the signature σ_i to the second device, whereby the method further comprises: verifying, by the second device, the signature σ_i using the verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, computing a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, a part of the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( ); and generating the common session key SK by providing the common-secret cs_j to a Key Deriving Function.

With reference to the eighth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and *H*() is a collision-resistant hash function; the first element k_i is defined as $k_i = (R_i, g^a)$; the second element k_j is defined as $k_j = ({R_i}^b, g^b)$; the two input function f( )comprises a concatenation function or an exclusive-OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j,R_j)}, {(g^b)^{s_i}}/{{R_i}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i,R_i)}, {(g^a)^{s_j}}/{{R_j}^a})$.

With reference to the eighth aspect, in accordance with embodiments of the invention, the secure server further: receives all communications and transmissions exchanged between the first and second devices; and computes a common-secret cs_{kgc} by providing the master secret key *msk*, a part of the first element k_i, the identity of the second device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i

with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

According to a ninth aspect of the invention, a method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: generating, by a secure server, a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; computing, by the first device, a first element k_i based on a first random value *a* generated by the first device; communicating to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device; compute a second element k_j based on the element u_j and the second random value *b*; generate a signature σ_j by signing the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verifying, by the first device, the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute a signature σ_i by signing a part of the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; computing an element u_i based on the part of the second element k_j and the first random value a; computing a common-secret cs_i based on the element u_i and the master public key *mpk*; and generating the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the ninth aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the signature σ_i to the second device, the method further comprising: verifying, by the second device, the signature σ_i using the verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, computing a common-secret cs_j based on the element u_j and the master public key *mpk;* and generating the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

With reference to the ninth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = g^a$; the second element k_j is defined as $k_j = (U, g^b)$ where *U* is defined as $= g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$; the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

With reference to the ninth aspect, in accordance with embodiments of the invention, the secure server further: receives all communications and transmissions exchanged between the first and second devices; and computes a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein cs_{kgc} is defined as $cs_{kgc} = U^x$.

According to a tenth aspect of the invention, a method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprises: generating, by a second server, a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; computing, by the first device, a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*; computing signature σ_i by signing the first element k_i using a signing function of an Identity Based Signature Scheme and the private key sk_i; communicating to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the Identity Based Signature Scheme and the identity of the first device id_i; compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*, when the signed first element k_i is verified; compute a signature σ_j by signing the first element k_i and the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verifying, by the first device, the signature σ_j using the verification function of the Identity Based Signature Scheme and the identity of the second device id_j and compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*, when the

signature σ_j is verified; generating a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; computing a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generating the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the tenth aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the verification data vd_i to the second device, the method comprising: computing, by the second device, a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*; generating a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determining if the verification data vd_i matches with the verification data vd_j; and generating the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the tenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = b \cdot P$; and the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map $e: G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* and where cs_i is defined as $cs_i = e(s \cdot P, b \cdot P)^a$ and cs_j is defined as $cs_j = e(s \cdot P, a \cdot P)^b$.

With reference to the tenth aspect, in accordance with embodiments of the invention, the secure server further: receives all communications and transmissions exchanged between the first and second devices; and computes a common-secret cs_{kgc} based on the first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.P)^s$.

With reference to the tenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q*, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and the common-secret cs_i and the common

secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

With reference to the tenth aspect, in accordance with embodiments of the invention, the secure server further receives all communications and transmissions exchanged between the first and second devices; and computes a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.Q)^s$.

According to an eleventh aspect of the invention, a method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising: generating, by a secure server, a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; computing, by the first device, a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i; computing a signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i; communicating to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i, when the signature σ_i is verified; compute a signature σ_j by signing a part of the first element k_i and the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verifying, by the first device, the signature σ_j using the verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j; computing a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ), when the signature σ_j is verified; generating a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; computing a verification data vd_i by providing the verification key vk_i to a

Authentication Data Deriving function (AdDF); and generating the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the eleventh aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the verification data vd_i and the part of the private key sk_j as combined with the random value a to the second device, whereby the method further comprises: computing, by the second device, a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( ); generating a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determining if the verification data vd_i matches with the verification data vd_j; and generating the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the eleventh aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where g is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and $H()$ is a collision-resistant hash function; the first element k_i is defined as $k_i = (g^a)$; the second element k_j is defined as $k_j = ({R_i}^b, g^b)$; the two input function f( )comprises a concatenation function or an exclusive-OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j, R_j)}, {(g^b)^{s_i}}/{{R_i}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i, R_i)}, {(g^a)^{s_j}}/{{R_j}^a})$.

With reference to the eleventh aspect, in accordance with embodiments of the invention, the secure server further: receives all communications and transmissions exchanged between the first and second devices; and computes a common-secret cs_{kgc} by providing the master secret key *msk*, the first element k_i, the identity of the second device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

According to a twelfth aspect of the invention, a method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising: generating, by a secure server, a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, generating a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; computing, by the first device, a first element k_i based on a first random value *a* generated by the first device; computing signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i; communicating to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device, when the signature σ_i is verified; compute a second element k_j based on the element *u* and the second random value *b*; compute signature σ_j by signing the first element k_i and the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verifying, by the first device, the signature σ_j using the verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute an element u_i based on a part of the second element k_j and the first random value a, when the signature σ_j is verified; computing a common-secret cs_i based on the element u_i and the master public key *mpk*; generating a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; computing a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generating the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the twelfth aspect, in accordance with embodiments of the invention, the first device is further configured to communicate the verification data vd_i to the second device, whereby the method further comprises: computing, by the second device, a common-secret cs_j based on the element u_j and the master public key *mpk*; generating a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication

Data Deriving function (AdDF); determining if the verification data vd_i matches with the verification data vd_j; and generating the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the twelfth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = g^a$; the second element k_j is defined as $k_j = (U, g^b)$ where *U* is defined as $= g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$; the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

With reference to the twelfth aspect, in accordance with embodiments of the invention, the secure server further: receives all communications and transmissions exchanged between the first and second devices; computes a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = U^x$.

According to a thirteenth aspect of the invention, a first device *i* for generating a common session key SK for encoding digital communications between the first device *i* and a second device *j*, the first device comprising: a processor; and a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to: instruct a secure server to generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device, and to communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; compute a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*; communicate to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*; compute signature σ_j by signing the first element k_i and the second element k_j using a signing function of an Identity Based Signature Scheme and the private key sk_j; transmit to

the first device the second element k_j and the signature σ_j; verify the signature σ_j using a verification function of the Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the thirteenth aspect, in accordance with embodiments of the invention, the instructions for directing the processor to communicate to the second device further comprises: instructions for directing the processor to: communicate the signature σ_i to the second device, whereby upon receiving the communication, the second device is configured to: verify the signature σ_i using the verification function of the Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, and compute a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*, and generate the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

With reference to the thirteenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = b \cdot P$; and the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map $e: G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* defined over the elliptic curve and cs_i is defined as $cs_i = e(b \cdot P, s \cdot P)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot P)^b$.

With reference to the thirteenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and compute a common-secret cs_{kgc} based on the first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot P)^s$.

With reference to the thirteenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1

with prime order *q* defined over an elliptic curve, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and the common-secret cs_i and the common secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where G is cyclic group with prime order q defined over the elliptic curve, and cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

With reference to the thirteenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and compute a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* and the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot Q)^s$.

According to a fourteenth aspect of the invention, a first device *i* for generating a common session key SK for encoding digital communications between the first device *i* and a second device *j*, comprises a processor; and a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to: instruct a secure server to generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk,* a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device, and to communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; compute a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i; communicate to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i; compute a signature σ_j by signing a part of the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing a part of the second element k_j and a part of the private key sk_j as combined with the random value a using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i,

when the signature σ_j is verified; compute a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the fourteenth aspect, in accordance with embodiments of the invention, the instructions for directing the processor to communicate to the second device further comprises: instructions for directing the processor to: communicate the part of the private key sk_j as combined with the random value a and the signature σ_i to the second device, whereby upon receiving the communication, the second device is configured to: verify the signature σ_i using the verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, compute a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, a part of the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( ); and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function.

With reference to the fourteenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and *H*() is a collision-resistant hash function; the first element k_i is defined as $k_i = (R_i, g^a)$; the second element k_j is defined as $k_j = ({R_i}^b, g^b)$; the two input function f( )comprises a concatenation function or an exclusive-OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j,R_j)}, {(g^b)^{s_i}}/{{R_i}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i,R_i)}, {(g^a)^{s_j}}/{{R_j}^a})$.

With reference to the fourteenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and compute a common-secret cs_{kgc} by providing the master secret key *msk*, a part of the first element k_i, the identity of the second

device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

According to a fifteenth aspect of the invention, a first device *i* for generating a common session key SK for encoding digital communications between the first device *i* and a second device *j*, comprising: a processor; and a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to: instruct a secure server to generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device, and communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; compute a first element k_i based on a first random value *a* generated by the first device; communicate to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device; compute a second element k_j based on the element u_j and the second random value *b*; generate a signature σ_j by signing the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute a signature σ_i by signing a part of the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; compute an element u_i based on the part of the second element k_j and the first random value a; compute a common-secret cs_i based on the element u_i and the master public key *mpk*; and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the fifteenth aspect, in accordance with embodiments of the invention, the instructions for directing the processor to communicate to the second device further comprises: instructions for directing the processor to: communicate the signature σ_i to the second device, whereby upon receiving the communication, the second device is configured to: verify the signature σ_i using the verification function of the self-certified Identity Based

Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified, compute a common-secret cs_j based on the element u_j and the master public key *mpk;* and generate the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

With reference to the fifteenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = g^a$; the second element k_j is defined as $k_j = (U, g^b)$ where *U* is defined as $= g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$; the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

With reference to the fifteenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and to compute a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein cs_{kgc} is defined as $cs_{kgc} = U^x$.

According to a sixteenth aspect of the invention, a first device *i* for generating a common session key SK for encoding digital communications between the first device *i* and a second device *j*, comprises: a processor; and a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to: instruct a secure server to generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device and communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; compute a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*; compute signature σ_i by signing the first element k_i using a signing function of an Identity Based Signature Scheme and the private key sk_i; communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the Identity Based

Signature Scheme and the identity of the first device id_i; compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*, when the signed first element k_i is verified; compute a signature σ_j by signing the first element k_i and the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using the verification function of the Identity Based Signature Scheme and the identity of the second device id_j and compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*, when the signature σ_j is verified; generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the sixteenth aspect, in accordance with embodiments of the invention, the instructions for directing the processor to communicate to the second device further comprises: instructions for directing the processor to: communicate the verification data vd_i to the second device, whereby upon receiving the communication, the second device is configured to: compute a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*; generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determine if the verification data vd_i matches with the verification data vd_j; and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the sixteenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = b \cdot P$; and the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map $e: G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* and where cs_i is defined as $cs_i = e(s \cdot P, b \cdot P)^a$ and cs_j is defined as $cs_j = e(s \cdot P, a \cdot P)^b$.

With reference to the sixteenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and to compute a common-secret cs_{kgc} based on the

first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.P)^s$.

With reference to the sixteenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q*, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = a \cdot P$; the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and the common-secret cs_i and the common secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

With reference to the sixteenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and to compute a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.Q)^s$.

According to a seventeenth aspect of the invention, a first device *i* for generating a common session key SK for encoding digital communications between the first device *i* and a second device *j*, comprises a processor; and a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to: instruct a secure server to generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device, and communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; compute a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i; compute a signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i; communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i, when the signature σ_i is verified; compute a signature σ_j by signing a part

of the first element k_i and the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature $\sigma_{j;}$ verify the signature σ_j using the verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j; compute a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ), when the signature σ_j is verified; generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the seventeenth aspect, in accordance with embodiments of the invention, the instructions for directing the processor to communicate to the second device further comprises: instructions for directing the processor to: communicate the verification data vd_i and the part of the private key sk_j as combined with the random value a to the second device, whereby upon receiving the communication, the second device is configured to: compute a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( ); generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; computing a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determine if the verification data vd_i matches with the verification data vd_j; and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the seventeenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and *H*() is a collision-resistant hash function; the first element k_i is defined as $k_i = (g^a)$; the second element k_j is defined as $k_j = ({R_i}^b, g^b)$; the two input function f( )comprises a concatenation

function or an exclusive-OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j,R_j)}, {(g^b)^{s_i}}/{R_i{}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i,R_i)}, {(g^a)^{s_j}}/{R_j{}^a})$.

With reference to the seventeenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and to compute a common-secret cs_{kgc} by providing the master secret key *msk*, the first element k_i, the identity of the second device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

According to an eighteenth aspect of the invention, a first device *i* for generating a common session key SK for encoding digital communications between the first device *i* and a second device *j*, comprises: a processor; and a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to: instruct a secure server to generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device, and to communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device; compute a first element k_i based on a first random value *a* generated by the first device; compute signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i; communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to: verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device, when the signature σ_i is verified; compute a second element k_j based on the element *u* and the second random value *b*; compute signature σ_j by signing the first element k_i and the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j; transmit to the first device the second element k_j and the signature σ_j; verify the signature σ_j using the verification function of the self-certified Identity

Based Signature Scheme and the identity of the second device id_j, and compute an element u_i based on a part of the second element k_j and the first random value a, when the signature σ_j is verified; compute a common-secret cs_i based on the element u_i and the master public key *mpk*; generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function; compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

With reference to the eighteenth aspect, in accordance with embodiments of the invention, the instructions for directing the processor to communicate to the second device further comprises: instructions for directing the processor to: communicate the verification data vd_i to the second device, whereby upon receiving the communication, the second device is configured to: compute a common-secret cs_j based on the element u_j and the master public key *mpk*; generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function; compute a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF); determine if the verification data vd_i matches with the verification data vd_j; and generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

With reference to the eighteenth aspect, in accordance with embodiments of the invention, the master public key mpk comprises g^x where g is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*; the first element k_i is defined as $k_i = g^a$; the second element k_j is defined as $k_j = (U, g^b)$ where U is defined as $= g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$; the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

With reference to the eighteenth aspect, in accordance with embodiments of the invention, the instructing the secure server further comprises: instructions for directing the processor to: instruct the secure server to receive all communications and transmissions exchanged between the first and second devices; and to compute a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = U^x$.

Brief Description of the Drawings

The above advantages and features in accordance with this invention are described in the following detailed description and are shown in the following drawings:

Figure 1 illustrating a block diagram representative of an entity-pair authentication and a common session key generation system for the authenticated entity-pair in accordance with embodiments of the invention;

Figure 2 illustrating a block diagram representative of components in an electronic device or server for implementing embodiments in accordance with embodiments of the invention;

Figure 3 illustrating a timing diagram for the authentication of an entity-pair and for the generation of a common session key for the authenticated entity-pair in accordance with embodiments of the invention;

Figure 4 illustrating a flow diagram of a process for verifying the authenticity of a second entity and for generating a common session key in accordance with embodiments of the invention.

Detailed Description

This invention relates to a system and method for generating common session keys that have strong forward security for encoding digital communications between devices. In particular, the system utilizes a forward secure identity-based authenticated key exchange scheme to allow two devices to verify the veracity of each device before these authenticated devices proceed to generate a common session key that is then utilized to encode digital communications between these two devices.

Further, the invention allows the generated common session keys to be escrowed to an authorized Key Generation Centre (KGC) whereby the escrow of the generated common session keys is not dependent on the escrow of users' private keys to the KGC. This means that, the common session keys of the users may be computed by the KGC itself once the KGC has gathered all communications that took place between users of the scheme, without the KGC being able to compute/know the users' private keys.

Figure 1 illustrates a block diagram of an entity-pair authentication and a common session key generation system in accordance with embodiments of the invention. One

skilled in the art will recognize that the term entity and device may be used interchangeably throughout the description without departing from the invention.

The system illustrated in Figure 1 comprises devices or entities 105, 110, that are connected to secure server 120. Entities 105 and 110 each may comprise, but is not limited to, any device that is able to carry out wireless communicative functions such as a smart phone, a tablet computer, a mobile computer, a netbook, a wearable electronic device such as smart watch, smart plugs, or transceivers that may be found in smart devices or Internet of Things (IoT) enabled devices, and etc.

As for secure server 120, this server may comprise a secure cloud server or a remotely located secure server which is able to communicate wirelessly with entities 105 and 110 either through Internet 115 or directly with entities 105 and 110. If server 120 is configured to communicate with entities 105 and 110 through Internet 115, server 120 may do so via wired networks or wireless networks 125 such as, but are not limited to, cellular networks, satellite networks, telecommunication networks, or Wide Area Networks (WAN). Alternatively, if server 120 is configured to communicate directly with entities 105 and 110, this may be accomplished through wireless networks 130 such as, but not limited to, Wireless-Fidelity (Wi-Fi), Bluetooth, or Near Field Communication (NFC). It should be noted that entities 105 and 110 may utilize either one of wireless network 125 (via the Internet) or wireless network 130 (direct communication) to exchange data messages with one another.

Figure 2 illustrates a block diagram representative of components of an electronic device 200 that is provided within entities 105, 110 and server 120 for implementing embodiments in accordance with embodiments of the invention. One skilled in the art will recognize that the exact configuration of each electronic device provided within the entities or the server may be different and the exact configuration of electronic device 200 may vary and Figure 2 is provided by way of example only.

In embodiments of the invention, device 200 comprises controller 201 and user interface 202. User interface 202 is arranged to enable manual interactions between a user and electronic device 200 and for this purpose includes the input/output components required for the user to enter instructions to control electronic device 200. A person skilled in the art will recognize that components of user interface 202 may vary from embodiment to embodiment but will typically include one or more of display 240, keyboard 235 and track-pad 236.

Controller 201 is in data communication with user interface 202 via bus 215 and includes memory 220, Central Processing Unit (CPU) 205 mounted on a circuit board that processes instructions and data for performing the method of this embodiment, an operating system 206, an input/output (I/O) interface 230 for communicating with user interface 202 and a communications interface, in this embodiment in the form of a network card 250. Network card 250 may, for example, be utilized to send data from electronic device 200 via a wired or wireless network to other processing devices or to receive data via the wired or wireless network. Wireless networks that may be utilized by network card 250 include, but are not limited to, Wireless-Fidelity (Wi-Fi), Bluetooth, Near Field Communication (NFC), cellular networks, satellite networks, telecommunication networks, Wide Area Networks (WAN) and etc.

Memory 220 and operating system 206 are in data communication with CPU 205 via bus 210. The memory components include both volatile and non-volatile memory and more than one of each type of memory, including Random Access Memory (RAM) 220, Read Only Memory (ROM) 225 and a mass storage device 245, the last comprising one or more solid-state drives (SSDs). Memory 220 also includes secure storage 246 for securely storing secret keys, or private keys. It should be noted that the contents within secure storage 246 are only accessible by a super-user or administrator of device 200 and may not be accessed by any user of device 200. One skilled in the art will recognize that the memory components described above comprise non-transitory computer-readable media and shall be taken to comprise all computer-readable media except for a transitory, propagating signal. Typically, the instructions are stored as program code in the memory components but can also be hardwired. Memory 220 may include a kernel and/or programming modules such as a software application that may be stored in either volatile or non-volatile memory.

Herein the term “CPU” is used to refer generically to any device or component that can process such instructions and may include: a microprocessor, microcontroller, programmable logic device or other computational device. That is, CPU 205 may be provided by any suitable logic circuitry for receiving inputs, processing them in accordance with instructions stored in memory and generating outputs (for example to the memory components or on display 240). In this embodiment, CPU 205 may be a single core or multi-core processor with memory addressable space. In one example, CPU 205 may be multi-core, comprising—for example—an 8 core CPU.

Referring back to Figure 1, prior to adding devices 105 and 110 to the device-pair authentication and the common session key generation system in accordance with

embodiments of the invention, server 120, which is configured as a Key Generation Centre, will first initiate a setup procedure based on identity based signature schemes to generate a master secret key *msk* and a master public key *mpk*.

Server 120 will then select a cryptographic collision-resistant hash function H: {0,1}* → $\{0,1\}^{l}$, whereby ɻ is an appropriate integer known to a person skilled in the art. Server 120 may also select an Authentication Data Deriving Function and a Key Deriving Function that is to be adopted for use in the system. In embodiments of the invention, the Authentication Data Deriving Function (AdDF) may include any algorithm or scheme for verifying the authenticity of a message such as a scheme for generating a message authentication code (MAC), a scheme for generating a message integrity code or a keyed hash function while the Key Deriving Function (KDF) may include any scheme for deriving a secret key from a secret value such as a collision-resistant hash function.

When device 105 or 110 joins the system, a private key unique to each of these devices will be issued by secure server 120. These unique private keys once generated will then be communicated to each of these devices whereby the respective private keys will then be stored in the secure memory within each of devices 105 and 110.

In particular, when device 105 registers itself with server 120, device 105 will communicate its identity to server 120. The identity of device 105 may comprise its user name, email address, telephone number, IP address, MAC address, or any alphanumeric combination that may be utilized to uniquely identify entity 105. Server 120 then provides the identity id_{105} associated with device 105 to a selected identity based signature scheme to generate a secret private key sk_{105} for entity 105.

Similarly, when device 110 registers itself with server 120, device 110 will also transmit its identity to server 120. Server 120 then provides identity id_{110} associated with device 110 to the same selected identity based signature scheme to generate a secret private key sk_{110} for device 110.

The private keys sk_{105} and sk_{110} are then communicated to their respective devices. Once the private keys have been stored in the secure memory of the respective devices, the device pair, i.e. device 105 and 110, may then commence authentication procedures. Upon successfully authenticating each other, the device -pair may then proceed to generate a common session key for encoding or signing digital communications sent between each other. In the following description, for ease of reading, device 105 may also be identified as device “i" while device 110 may also be identified as device “j”.

Embodiments Based On Symmetric Bilinear Map

In a first embodiment of the invention, the selected identity based signature scheme may be based on a symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order q defined over an elliptic curve. For such identity based signature schemes, the master public key *mpk* may be defined as mpk = ($s.P$, P), where P is a generator of G_1 and $msk = s \in Z_q^*$ where Z_q^* are non-zero residuals of modular q. In this embodiment, device 105's private key is defined as sk_i while device 110's private key is defined as sk_j, as per the selected identity based signature scheme.

The generation of a common session key for devices 105 and 110 in accordance with this embodiment of the invention is illustrated in Figure 3. In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value a and the master public key *mpk*. The element k_i may be defined as $k_i = (a \cdot P)$ where P is a parameter obtained from the master public key *mpk*.

Device 105 will then transmit the element k_i and the identity of device 105 id_i to device 110 at step 310. Upon receiving the element k_i from device 110, at step 315, device 110 will then generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the master public key *mpk*. The element k_j may be defined as $k_j = (b \cdot P)$ where P is a parameter obtained from the master public key *mpk*. The element k_i and the element k_j are then signed using a signing function associated with the selected Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as $\sigma_j = Sign(sk_j, a \cdot P || b \cdot P)$, where || denotes concatenation.

At step 320, device 110 then transmits the signature σ_j, i.e. $\sigma_j = Sign (sk_j, a \cdot P || b \cdot P)$, and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the selected Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 will then proceed to compute its own

signature σ_i. In particular, device 105's signature σ_i may be computed as σ_i = Sign (sk_i, b·P). Device 105 then proceeds to calculate its common-secret cs_i as $e(b \cdot P, s \cdot P)^a$ where s·P is a parameter obtained from the master public key *mpk* and e is the symmetric bilinear map *e*: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order *q*. The calculated common-secret cs_i is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF(cs_i).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate its signature σ_i to device 110. Alternatively, device 105 may also communicate its signature σ_i to device 110 once it has computed its signature σ_i. This takes place at step 330.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the selected Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 335 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to calculate its common-secret cs_j as $e(a \cdot P, s \cdot P)^b$ where s·P is the parameter obtained from the master public key *mpk* and e is the symmetric bilinear map *e*: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order *q*. The calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF(cs_j).

In this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to utilize its own master secret key s to generate the common session key SK. In particular, server 120 may utilize the element k_i as shared by device 105 and the element k_j as shared by device 110 to generate a common secret cs_{kgc} as $cs_{kgc} = e(a \cdot P, b \cdot P)^s$ and a common session key SK as SK = KDF(cs_{kgc}).

In another embodiment of the first embodiment of the invention, the generation of a common session key for devices 105 and 110 in accordance with this embodiment of the may be as follows.

In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value "a" and the

master public key *mpk*. The element k_i may be defined as $k_i = (a \cdot P)$ where P is a parameter obtained from the master public key *mpk*. Device 105 will also at this step proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as $\sigma_i = \text{Sign}(sk_i, a \cdot P)$.

Device 105 will then transmit its signature σ_i the element k_i and the identity of device 105 id_i to device 110 at step 310.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 315 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the master public key *mpk*. The element k_j may be defined as $k_j = (b \cdot P)$ where P is a parameter obtained from the master public key *mpk*. The element k_i and the element kj are then signed using a signing function associated with the Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as $\sigma_j = \text{Sign}(sk_j, a \cdot P || b \cdot P)$.

At step 320, device 110 then transmits the signature σ_j, i.e. $\sigma_j = \text{Sign}(sk_j, a \cdot P \,||\, b \cdot P)$, and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 then proceeds to calculate its common-secret cs_i as $e(b \cdot P, s \cdot P)^a$ where $s \cdot P$ is a parameter obtained from the master public key *mpk* and e is the symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order q. The common-secret cs_i is then provided to a Key Deriving Function (KDF) to obtain verification key vk_i as $vk_i = \text{KDF}(cs_i)$. The obtained verification key vk_i is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_i as vd_i

= AdDF(vk_i). The calculated common-secret cs_i is also simultaneously provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_i).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate verification data vd_i to device 110. Alternatively, device 105 may also communicate verification data vd_i to device 110 once it has computed verification data vd_i. This takes place at step 330.

Upon receiving verification data vd_i at step 335, device 110 will then compute its common-secret cs_j as e $(a \cdot P, s \cdot P)^b$ where $s \cdot P$ is the parameter obtained from the master public key *mpk* and e is the symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order q. The common-secret cs_j is then provided to a Key Deriving Function (KDF) to obtain verification key vk_j as vk_j = KDF (cs_j). The obtained verification key vk_j is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_j as vd_j = AdDF(vk_j). Verification data vd_j is then compared with verification data vd_i and if a match is not found, device 110 will cancel the process. If a match is found, the calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_j).

Similarly, for this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to utilize its own master secret key s to generate the common session key SK. In particular, server 120 may utilize the element k_i as shared by device 105 and the element k_j as shared by device 110 to generate a common secret cs_{kgc} as cs_{kgc} = e $(a \cdot P, b \cdot P)^s$ and a common session key SK as SK = KDF(cs_{kgc}).

In yet another embodiment of the first embodiment of the invention, option fields op_f1, op_f2, op_f3, op_f4, op_f5, op_f6 or op_f7 may be added to the various functions such as the signing and verification functions, and the Key Deriving Function and may comprise identities of entities of the system where applicable or any application specific data as determined by the entities themselves. These option fields may be applied to both embodiments above.

If the option fields are adopted, the generation of a common session key for devices 105 and 110 in accordance with this embodiment of the invention may be as follows. In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value a and the master public

key *mpk*. The element k_i may be defined as $k_i = (a \cdot P)$ where P is a parameter obtained from the master public key *mpk*.

Device 105 will then transmit an option field op_f1, the element k_i and the identity of device 105 id_i to device 110 at step 310. Upon receiving the element k_i from device 110, at step 315, device 110 will then generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the master public key *mpk*. The element k_j may be defined as $k_j = (b \cdot P)$ where P is a parameter obtained from the master public key *mpk*. The element k_i and the element kj are then signed with option field op_f2 using a signing function associated with a Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as $\sigma_j = Sign(sk_j, a \cdot P || b \cdot P || op_f2)$.

At step 320, device 110 then transmits an option field op_f3, the signature σ_j, i.e. $\sigma_j = Sign (sk_j, a \cdot P || b \cdot P || op_f2)$, and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 will then proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as $\sigma_i = Sign (sk_i, b \cdot P || op_f4)$. Device 105 then proceeds to calculate its common-secret cs_i as $e(b \cdot P, s \cdot P)^a$ where $s \cdot P$ is a parameter obtained from the master public key *mpk* and e is the symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order q. The calculated common-secret cs_i is then provided to a Key Deriving Function (KDF) to generate a common session key SK as $SK = KDF(cs_i, opt_f6)$.

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate an option field op_f5 and its signature σ_i to device 110. Alternatively, device 105 may also communicate its signature σ_i to device 110 once it has computed its signature σ_i. This takes place at step 330.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 335 as Verify

(id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to calculate its common-secret cs_j as $e(a \cdot P, s \cdot P)^b$ where $s \cdot P$ is the parameter obtained from the master public key *mpk* and e is the symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order q. The calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as $SK = KDF(cs_j, opt_f6)$.

In still yet another embodiment of the first embodiment of the invention, the generation of a common session key for devices 105 and 110 using option fields in accordance with this embodiment may be as follows.

In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value "a" and the master public key *mpk*. The element k_i may be defined as $k_i = (a \cdot P)$ where P is a parameter obtained from the master public key *mpk*. Device 105 will also at this step proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as $\sigma_i = Sign(sk_i, a \cdot P || opt_f1)$.

Device 105 will then transmit an option field opt_f2, its signature σ_i, the element k_i and the identity of device 105 id_i to device 110 at step 310.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 315 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the master public key *mpk*. The element k_j may be defined as $k_j = (b \cdot P)$ where P is a parameter obtained from the master public key *mpk*. The element k_i and the element kj with option field opt_f3 are then signed using a signing function associated with the Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as $\sigma_j = Sign(sk_j, a \cdot P || b \cdot P || opt_f3)$.

At step 320, device 110 then transmits an option field opt_f4, the signature σ_j, i.e. σ_j = Sign (sk_j, a·P || b·P|| opt_f3), and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 then proceeds to calculate its common-secret cs_i as $e(b\cdot P, s\cdot P)^a$ where s·P is a parameter obtained from the master public key *mpk* and e is the symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order q. The common-secret cs_i is then provided to a Key Deriving Function (KDF) to obtain verification key vk_i as vk_i = KDF (cs_i, opt_f5). The obtained verification key vk_i is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_i as vd_i = AdDF(vk_i, opt_f6). The calculated common-secret cs_i is also simultaneously provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_i, opt_f7).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate the option field opt_f6 and verification data vd_i to device 110. Alternatively, device 105 may also communicate the option field opt_f6 and verification data vd_i to device 110 once it has computed verification data vd_i. This takes place at step 330.

Upon receiving verification data vd_i at step 335, device 110 will then compute its common-secret cs_j as $e(a\cdot P, s\cdot P)^b$ where s·P is the parameter obtained from the master public key *mpk* and e is the symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$, where G_1 and G_2 are cyclic groups with prime order q. The common-secret cs_j is then provided to a Key Deriving Function (KDF) to obtain verification key vk_j as vk_j = KDF (cs_j, opt_f5). The obtained verification key vk_j is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_j as vd_j = AdDF(vk_j, opt_f6). Verification data vd_j is then compared with verification data vd_i and if a match is not found, device 110 will cancel the process. If a match is found, the calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_j, opt_f7). The application of the option fields to the signing function, the verifying function, the Key

Deriving Function and/or the AdDF function may also be applied to the second, third and fourth embodiments as described in the following sections.

Embodiments Based On Asymmetric Bilinear Map

In a second embodiment of the invention, the selected identity based signature scheme may be based on an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G_t$, where G_1, G_2 and G_t are cyclic groups with prime order q; in addition, G_1, G_2 are defined over an elliptic curve. For such identity based signature schemes, the master public key *mpk* may be defined as mpk = ($s.Q$, Q, P), where P is a generator of G_1, Q is a generator of G_2 and $msk = s \in Z_q^*$ where Z_q^* are non-zero residuals of modular q. In this embodiment, device 105's private key is defined as sk_i while device 110's private key is defined as sk_j as per the selected identity based signature scheme.

The generation of a common session key for devices 105 and 110 in accordance with this embodiment of the invention is illustrated in Figure 3. In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value a and the master public key *mpk*. The element k_i may be defined as $k_i = (a \cdot P)$ where P is a parameter obtained from the master public key *mpk*.

Device 105 will then transmit the element k_i and the identity of device 105 id_i to device 110 at step 310. Upon receiving the element k_i from device 110, at step 315, device 110 will then generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the master public key *mpk*. The element k_j may be defined as $k_j = (b \cdot P, b \cdot Q)$ where P and Q are parameters obtained from the master public key *mpk*. The element k_i and the element kj are then signed using a signing function associated with the Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as σ_j = Sign(sk_j, $a \cdot P || b \cdot P || b \cdot Q$).

At step 320, device 110 then transmits the signature σ_j, i.e. σ_j = Sign (sk_j, $a \cdot P || b \cdot P || b \cdot Q$), and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step

325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 will then proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as $\sigma_i = Sign\ (sk_i, b \cdot P)$. Device 105 then proceeds to calculate its common-secret cs_i as $e(b \cdot P, s \cdot Q)^a$ where $s \cdot Q$ is a parameter obtained from the master public key *mpk* and e is the asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G_t$, where G_1, G_2 and G_t are cyclic groups with prime order q. The calculated common-secret cs_i is then provided to a Key Deriving Function (KDF) to generate a common session key SK as $SK = KDF(cs_i)$.

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate its signature σ_i to device 110. Alternatively, device 105 may also communicate its signature σ_i to device 110 once it has computed its signature σ_i. This takes place at step 330.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 335 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to calculate its common-secret cs_j as $e\ (a \cdot P, s \cdot Q)^b$ where $s \cdot Q$ is the parameter obtained from the master public key *mpk* and e is the asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G_t$, where G_1, G_2 and G_t are cyclic groups with prime order q. The calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as $SK = KDF(cs_j)$.

In this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to utilize its own master secret key s to generate the common session key SK. In particular, server 120 may utilize the element k_i as shared by device 105 and a part of the element k_j as shared by device 110 to generate a common secret cs_{kgc} as $cs_{kgc} = e\ (a \cdot P\ ,\ b \cdot Q)^s$ and a common session key SK as $SK = KDF(cs_{kgc})$.

In another embodiment of the second embodiment of the invention, the generation of a common session key for devices 105 and 110 in accordance with this embodiment of the may be as follows.

In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value "a" and the master public key *mpk*. The element k_i may be defined as $k_i = (a \cdot P)$ where P is a parameter obtained from the master public key *mpk*. Device 105 will also at this step proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as $\sigma_i = \text{Sign}(sk_i, a \cdot P)$.

Device 105 will then transmit its signature σ_i the element k_i and the identity of device 105 id_i to device 110 at step 310.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 315 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the master public key *mpk*. The element k_j may be defined as $k_j = (b \cdot P, b \cdot Q)$ where P and Q are parameters obtained from the master public key *mpk*. The element k_i and the element k_j are then signed using a signing function associated with the Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as $\sigma_j = \text{Sign}(sk_j, a \cdot P || b \cdot P || b \cdot Q)$.

At step 320, device 110 then transmits the signature σ_j, i.e. $\sigma_j = \text{Sign}(sk_j, a \cdot P || b \cdot P || b \cdot Q)$, and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is

successful in verifying the signature $\sigma_{j,}$ device 105 then proceeds to calculate its common-secret cs_i as e $(b \cdot P, s \cdot Q)^a$ where $s \cdot Q$ is a parameter obtained from the master public key *mpk* and e is the asymmetric bilinear map *e*: $G_1 \times G_2 \rightarrow G_t$, where $G_{1,}$ G_2 and G_t are cyclic groups with prime order *q*. The common-secret cs_i is then provided to a Key Deriving Function (KDF) to obtain verification key vk_i as vk_i = KDF (cs_i). The obtained verification key vk_i is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_i as vd_i = AdDF(vk_i). The calculated common-secret cs_i is also simultaneously provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_i).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate verification data vd_i to device 110. Alternatively, device 105 may also communicate verification data vd_i to device 110 once it has computed verification data vd_i. This takes place at step 330.

Upon receiving verification data vd_i at step 335, device 110 will then compute its common-secret cs_j as e $(a \cdot P, s \cdot Q)^b$ where $s \cdot Q$ is the parameter obtained from the master public key *mpk* and e is the symmetric bilinear map *e*: $G_1 \times G_2 \rightarrow G_t$, where $G_{1,}$ G_2 and G_t are cyclic groups with prime order *q*. The common-secret cs_j is then provided to a Key Deriving Function (KDF) to obtain verification key vk_j as vk_j = KDF (cs_j). The obtained verification key vk_j is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_j as vd_j = AdDF(vk_j). Verification data vd_j is then compared with verification data vd_i and if a match is not found, device 110 will cancel the process. If a match is found, the calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_j).

Similarly, for this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to utilize its own master secret key s to generate the common session key SK. In particular, server 120 may utilize the element k_i as shared by device 105 and a part of the element k_j as shared by device 110 to generate a common secret cs_{kgc} as cs_{kgc} = e $(a \cdot P, b \cdot Q)^s$ and a common session key SK = KDF(cs_{kgc}).

In yet another embodiment of the second embodiment of the invention, option fields op_f1, op_f2, op_f3, op_f4, op_f5, op_f6 or op_f7 may be added to the various functions such as the signing and verification functions, and the Key Deriving Function and may comprise identities of entities of the system where applicable or any application specific data

as determined by the entities themselves. These option fields may be applied to both embodiments above.

If the option fields are adopted, the generation of a common session key for devices 105 and 110 in accordance with this embodiment of the invention may be as follows. In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value a and the master public key *mpk*. The element k_i may be defined as $k_i = (a \cdot P)$ where P is a parameter obtained from the master public key *mpk*.

Device 105 will then transmit an option field op_f1, the element k_i and the identity of device 105 id_i to device 110 at step 310. Upon receiving the element k_i from device 110, at step 315, device 110 will then generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the master public key *mpk*. The element k_j may be defined as $k_j = (b \cdot P, b \cdot Q)$ where P and Q are parameters obtained from the master public key *mpk*. The element k_i and the element k_j with option field op_f2 are then signed using a signing function associated with the Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as $\sigma_j = \text{Sign}(sk_j, a \cdot P || b \cdot P || b \cdot Q || \text{opt_f2})$.

At step 320, device 110 then transmits an option field op_f3, the signature σ_j, i.e. $\sigma_j = \text{Sign}(sk_j, a \cdot P || b \cdot P || b \cdot Q || \text{op_f2})$, and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 will then proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as $\sigma_i = \text{Sign}(sk_i, b \cdot P || \text{opt_f4})$. Device 105 then proceeds to calculate its common-secret cs_i as $e(b \cdot P, s \cdot Q)^a$ where $s \cdot Q$ is a parameter obtained from the master public key *mpk* and e is the asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G_t$, where G_1, G_2 and G_t are cyclic groups with prime order q. The calculated common-secret cs_i is then provided to a Key Deriving Function (KDF) to generate a common session key SK as $SK = \text{KDF}(cs_i, \text{opt_f6})$.

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate an option field op_f5 and its signature σ_i to device 110. Alternatively, device 105 may also communicate the option field op_f5 and its signature σ_i to device 110 once it has computed its signature σ_i. This takes place at step 330.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 335 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to calculate its common-secret cs_j as $e(a \cdot P, s \cdot Q)^b$ where $s \cdot Q$ is the parameter obtained from the master public key *mpk* and e is the asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G_t$, where G_1, G_2 and G_t are cyclic groups with prime order q. The calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF(cs_j, opt_f6).

Those skilled in the art may construct alternatives to the above embodiments based on asymmetric bilinear map by observing that k_i for device 105 may be defined as $k_i = (a \cdot P, a.Q)$ and k_j for device 110 may be defined as $k_j = (b.P)$. In the alternatives, if server 120 were configured to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to utilize its own master secret key s to generate a common secret cs_{kgc} as $cs_{kgc} = e(b.P, a.Q)^s$ and in turn the common session key SK.

Embodiments Based on Self-Certified IBS With Weak Forward Security

In a third embodiment of the invention, the selected identity based signature (IBS) scheme may be based on self-certified identity based signature schemes, which are a special class of IBS schemes. These self-certified IBS schemes are based on DLP (Discrete Logarithm Problem) or ECDLP (Elliptic Curve Discrete Logarithm Problem), and the signatures generated by a particular private key always contains a fixed component specific to that private signing key (thus it is called key-specific data, KSD), where the KSD assists in the verification of the signatures and is often attached to the signatures as part thereof. Typical examples of self-certified IBS schemes include ISO/IEC 29192-4 and IETF RFC 6507. Self-certified IBS can be implemented over either conventional finite field, or elliptic curves (defined upon a finite field). In the subsequent description, although multiplicative notations are utilized for finite field arithmetic operations; it should be straightforward for a

person skilled in the art to realize that the arithmetic operations should be described using additive notations when implemented over elliptic curves.

The structure of the private keys of the specific self-certified IBS scheme defined in ISO/IEC 19192-4 will be utilized in this embodiment for illustration purposes, but the idea contained in this embodiment is directly applicable to the scheme in IETF RFC 6507 and other self-certified IBS schemes as well. In the IBS scheme selected for this embodiment, the master key pair held by KGC is (*mpk*, *msk*) = ($y = g^x$, x), where g is a generator of a cyclic multiplicative group G with prime order q, and $x \in Z_q^*$ where Z_q^* are non-zero residuals of modular q. The private key for device 105 is $sk_i = (R_i = g^{r_i}, s_i = r_i + xh(id_i, R_i))$, and the private key for device 110 is $sk_j = (R_j = g^{r_j}, s_j = r_j + xh(id_j, R_j))$, where r_i *and* $r_j \in_R Z_q^*$ and $h(.)$ is a cryptographic hash function.

The generation of a common session key for devices 105 and 110 in accordance with this embodiment of the invention is illustrated in Figure 3. In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value "a" and the private key sk_i. The element k_i may be defined as $k_i = (g^a, R_i)$ where R_i is a parameter obtained from the private key sk_i and g is a generator of a cyclic multiplicative group G with prime order q.

Device 105 will then transmit the element k_i and the identity of device 105 id_i to device 110 at step 310.

Upon receiving the element k_i from device 110, at step 315, device 110 will then generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 110 will then generate an element k_j based on the random value b, and the first element k_i. The element k_j may be defined as $k_j = (g^b, R_i^b)$ where R_i is a parameter obtained from the element k_i and g is a generator of a cyclic multiplicative group G with prime order q. A part of the element k_i and the element k_j are then signed using a signing function or algorithm associated with the self-certified Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as $\sigma_j = \text{Sign}(sk_j, g^a \| g^b \| R_i^b)$.

At step 320, device 110 then transmits the signature σ_j, i.e. $\sigma_j = \text{Sign}(sk_j, g^a \| g^b \| R_i^b)$, and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the self-certified Identity

Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 will then proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as σ_i = Sign (sk_i, g^b || R_j^a).

Device 105 then proceeds to calculate its common-secret cs_i as $cs_i = f(y^{a.h(id_j,R_j)}, {(g^b)^{s_i}}/{R_i{}^b})$ where *f(.)* is a two-input function such as a concatenation function || or an exclusive-OR function $\oplus$, and h(.) is a cryptographic hash function. The calculated common-secret cs_i is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF(cs_i).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate its signature σ_i and a part of the private key sk_j as combined with the random value a, i.e. R_j^a, to device 110. Alternatively, device 105 may also communicate its signature σ_i and a part of the private key sk_j as combined with the random value a, i.e. R_j^a, to device 110 once it has computed its signature σ_i. This takes place at step 330.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the self-certified Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 335 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process.

Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to calculate its common-secret cs_j as $cs_j = f({(g^a)^{s_j}}/{R_j{}^a}, y^{b.h(id_i,R_i)})$ where *f(.)* is a two-input function such as a concatenation function || or an exclusive-OR function $\oplus$, and h(.) is a cryptographic hash function. The calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF(cs_j).

In this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to compute $cs_{kgc} = f(y^{a.h(id_j,R_j)}, y^{b.h(id_i,R_i)})$ and in turn SK = KDF(cs_{kgc}).

In another embodiment of the third embodiment of the invention, the generation of a common session key for devices 105 and 110 in accordance with this embodiment of the may be as follows.

In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value "a". The element k_i may be defined as $k_i = (g^a)$ where *g* is a generator of a cyclic group *G* with prime order *q.* Device 105 will also at this step proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as σ_i = Sign (sk_i, g^a).

Device 105 will then transmit its signature σ_i the element k_i and the identity of device 105 id_i to device 110 at step 310.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the self-certified Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 315 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q.

Device 110 will then generate an element k_j based on the random value b, and the first element k_i. The element k_j may be defined as $k_j = (g^b , R_i^b)$ where R_i is a parameter obtained from the signature σ_i and *g* is a generator of a cyclic multiplicative group *G* with prime order *q.* A part of the element k_i and the element k_j are then signed using a self-certified Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as σ_j = Sign (sk_j, g^a || g^b || R_i^b).

At step 320, device 110 then transmits the signature σ_j, i.e σ_j = Sign (sk_j, g^a || g^b || R_i^b), and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the self-certified Identity

Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process.

Conversely, if the verification function is successful in verifying the signature σ_j, proceeds to calculate its common-secret cs_i as $cs_i = f(y^{a.h(id_j,R_j)}, {(g^b)^{s_i}}/{R_i{}^b})$ where *f(.)* is a two-input function such as a concatenation function || or an exclusive-OR function $\oplus$, and *h(.)* is a cryptographic hash function. The common-secret cs_i is then provided to a Key Deriving Function (KDF) to obtain verification key vk_i as vk_i = KDF (cs_i). The obtained verification key vk_i is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_i as vd_i = AdDF(vk_i, R_j^a). The calculated common-secret cs_i is also simultaneously provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_i).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate verification data vd_i and R_j^a to device 110. Alternatively, device 105 may also communicate verification data vd_i and R_j^a to device 110 once it has computed verification data vd_i. This takes place at step 330.

Upon receiving verification data vd_i at step 335, device 110 will then proceed to calculate its common-secret cs_j as $cs_j = f(y^{b.h(id_i,R_i)}, {(g^a)^{s_j}}/{R_j{}^a})$ where *f(.)* is a two-input function such as a concatenation function || or an exclusive-OR function $\oplus$, and *h(.)* is a cryptographic hash function. The common-secret cs_j is then provided to a Key Deriving Function (KDF) to obtain verification key vk_j as vk_j = KDF (cs_j). The obtained verification key vk_j is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_j as vd_j = AdDF(vk_j, R_j^a). Verification data vd_j is then compared with verification data vd_i and if a match is not found, device 110 will cancel the process. If a match is found, the calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_j).

Similarly, for this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to compute $cs_{kgc} = f(y^{a.h(id_j,R_j)}, y^{b.h(id_i,R_i)})$.

Embodiments Based on Self-Certified IBS With Strong Forward Security

In a fourth embodiment of the invention, the selected identity based signature (IBS) scheme may be based on existing self-certified identity based signature schemes, which are a special class of IBS schemes. These self-certified IBS schemes are based on DLP (Discrete Logarithm Problem) or ECDLP (Elliptic Curve Discrete Logarithm Problem), and the signatures generated by a particular private key always contains a fixed component specific to that private signing key (thus it is called key-specific data, KSD), where the KSD assists in the verification of the signatures and is often attached to the signatures as part thereof. Typical examples of self-certified IBS schemes include ISO/IEC 29192-4 and IETF RFC 6507. Self-certified IBS can be implemented over either conventional finite field, or elliptic curves (defined upon a finite field). In the subsequent description of this embodiment, although multiplicative notations are utilized for finite field arithmetic operations; it should be straightforward for a person skilled in the art to realize that the arithmetic operations should be described using additive notations when implemented over elliptic curves.

The structure of the private keys of the specific self-certified IBS scheme defined in ISO/IEC 19192-4 will be utilized in this embodiment for illustration purposes, but the idea contained in this embodiment is directly applicable to the scheme in IETF RFC 6507 and other self-certified IBS schemes as well. In the IBS scheme selected for this embodiment, the master key pair held by KGC is (*mpk*, *msk*) = ($y = g^x$, x), where g is a generator of a cyclic multiplicative group G with prime order q, and $x \in Z_q^*$ where Z_q^* are non-zero residuals of modular q. The private key for device 105 is sk_i and the private key for device 110 is sk_j.

The generation of a common session key for devices 105 and 110 in accordance with this embodiment of the invention is illustrated in Figure 3. In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in Z_q^*$ where Z_q^* are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value "a" and the private key sk_i. The element k_i may be defined as $k_i = g^a$ where g is a generator of a cyclic multiplicative group G with prime order q.

Device 105 will then transmit the element k_i and the identity of device 105 id_i to device 110 at step 310.

Upon receiving the element k_i from device 110, at step 315, device 110 will then generate a random number $b \in Z_q^*$ where Z_q^* are non-zero residuals of modular q. Device 110 will then generate element u_j as $u_j = g^{ab}$. Element u_j is then used to compute a

parameter U as U = g^{u_j}. Device 110 then computes an element k_j based on the parameter U and the random number b. In particular, the element k_j may be defined as $k_j = (U, g^b)$ where g is a generator of a cyclic group *G* with prime order *q*. The element k_i and the element k_j are then signed using a signing function associated with a self-certified Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as σ_j = Sign (sk_j, g^a || g^b || U).

At step 320, device 110 then transmits the signature σ_j, i.e. σ_j = Sign (sk_j, g^a || g^b || U), and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the self-certified Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_j, device 105 will then proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as σ_i = Sign (sk_i, g^b).

Device 105 will then generate element u_i as $u_i = g^{ab}$ and device 105 then proceeds to calculate its common-secret cs_i as $cs_i = y^{u_i}$ where *y* is the master public key *mpk.* The calculated common-secret cs_i is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF(cs_i).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate its signature σ_i to device 110. Alternatively, device 105 may also communicate its signature σ_i to device 110 once it has computed its signature σ_i. This takes place at step 330.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the self-certified Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 335 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process.

Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to calculate its common-secret cs_j as $cs_j = y^{u_j}$ where *y* is the

master public key *mpk.* The calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF(cs_j).

In this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to compute $cs_{kgc} = U^x$ and in turn SK = KDF(cs_{kgc}).

In another embodiment of the fourth embodiment of the invention, the generation of a common session key for devices 105 and 110 in accordance with this embodiment of the may be as follows.

In particular, at step 305, device 105 will initiate the common session key generation process by first generating a random number $a \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q. Device 105 will then compute an element k_i using the random value "a" and the private key sk_i. The element k_i may be defined as $k_i = g^a$ where *g* is a generator of a cyclic multiplicative group *G* with prime order *q.* Device 105 will also at this step proceed to compute its own signature σ_i. In particular, device 105's signature σ_i may be computed as σ_i = Sign (sk_i, g^a).

Device 105 will then transmit its signature σ_i the element k_i and the identity of device 105 id_i to device 110 at step 310.

Upon receiving σ_i from device 105, device 110 then proceeds to verify the received signature σ_i using the same verification function associated with the self-certified Identity Based Signature Scheme, Verify( ), and the identity of device 105 id_i. This takes place at step 315 as Verify (id_i, σ_i). If the verification function returns a negative result, meaning if the signature σ_i may not be verified using the identity of the device 105 id_i, device 110 will abort the common session key generation process. Conversely, if the verification function is successful in verifying the signature σ_i, device 110 will then proceed to generate a random number $b \in \mathbb{Z}_q^*$ where $\mathbb{Z}_q^*$ are non-zero residuals of modular q.

Device 110 will then generate element u_j as $u_j = g^{ab}$. Element u_j is then used to compute a parameter U as $U = g^{u_j}$. Device 110 then forms an element k_j based on the parameter U and the random number b. In particular, the element k_j may be defined as k_j = (U, g^b) where *g* is a generator of a cyclic multiplicative group *G* with prime order *q.* The element k_i and the element k_j are then signed using a signing function associated with the self-certified Identity Based Signature Scheme, Sign ( ), and the private key sk_j to generate signature σ_j which is defined as σ_j = Sign (sk_j, g^a || g^b || U).

At step 320, device 110 then transmits the signature σ_j, i.e. σ_j = Sign (sk_j, g^a || g^b || U), and the second element k_j, to device 105.

Upon receiving σ_j and k_j from device 110, device 105 then proceeds to verify the received signature σ_j using a verification function associated with the self-certified Identity Based Signature Scheme, Verify( ), and the identity of the device 110 id_j. This takes place at step 325 as Verify (id_j, σ_j). If the verification function returns a negative result, meaning if the signature σ_j may not be verified using the identity of the device 110 id_j, device 105 will abort the common session key generation process.

Conversely, if the verification function is successful in verifying the signature σ_j, proceeds to generate element u_i as $u_i = g^{ab}$ and device 105 then proceeds to calculate its common-secret cs_i as $cs_i = y^{u_i}$ where *y* is the master public key *mpk.*

The common-secret cs_i is then provided to a Key Deriving Function (KDF) to obtain verification key vk_i as vk_i = KDF (cs_i). The obtained verification key vk_i is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_i as vd_i = AdDF(vk_i). The calculated common-secret cs_i is also simultaneously provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_i).

Once device 105 has completed the computation of the common session key SK, device 105 will then communicate verification data vd_i to device 110. Alternatively, device 105 may also communicate verification data vd_i to device 110 once it has computed verification data vd_i. This takes place at step 330.

Upon receiving verification data vd_i at step 335, device 110 will then proceed to calculate its common-secret cs_j as $cs_j = y^{u_j}$ where *y* is the master public key *mpk*. The common-secret cs_j is then provided to a Key Deriving Function (KDF) to obtain verification key vk_j as vk_j = KDF (cs_j). The obtained verification key vk_j is then used with an Authentication Data Deriving function (AdDF) to generate verification data vd_j as vd_j = AdDF(vk_j). Verification data vd_j is then compared with verification data vd_i and if a match is not found, device 110 will cancel the process. If a match is found, the calculated common-secret cs_j is then provided to a Key Deriving Function (KDF) to generate a common session key SK as SK = KDF (cs_j).

Similarly, for this embodiment of the invention, if server 120 were to receive all communications that were exchanged between device 105 and device 110, server 120 would be able to compute $cs_{kgc} = U^x$.

Those skilled in the art may construct alternatives to the above embodiments based on self-certified IBS with strong forward security by observing that device 105 may compute and transmit $U = g^{u_i}$ and $\sigma_i = \text{Sign}(sk_i, g^b||U)$ to device 110 at step 330, while k_j for device 110 may be defined as $k_j = (g^b)$ and σ_j for device 110 as $\sigma_j = \text{Sign}(sk_j, g^a || g^b)$.

In accordance with an embodiment of the invention, a method for a device “i” to generate a common session key SK for encoding digital communications between device “i" and a second device “j”, comprises the following steps:

Step 1, receive a master public key *mpk* and the private key $sk_{i;}$

Step 2, compute a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*;

Step 3, communicate to the second device the first element k_i and the identity id_i of the first device such that upon receiving the communication, the second device is configured to:

compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*;

compute signature σ_j by signing the first element k_i and the second element k_j using a signing function associated with an Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature $\sigma_{j;}$

Step 4, verify the signature σ_j using a verification function associated with the Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing the second element k_j using the signing function associated with the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified; and

Step 5, compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

In order to provide such a system or method, a process is needed for a device “i” to generate a common session key SK for encoding digital communications between the

device “i" and a second device “j”. The following description and Figure 4 describe embodiments of processes that provide processes in accordance with this invention.

Figure 4 illustrates process 400 that is performed by a first device “i" to generate a common session key SK for encoding digital communications between the first device “i" and a second device “j”. Process 400 begins at step 405 with process 400 receiving a master public key *mpk* and the private key sk_i from a secure server configured as a Key Generation Centre. At step 410, process 400 then computes a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*. At step 415, process 400 communicates to the second device the first element k_i and the identity id_i of the first device. Upon receiving the communication, the second device is then configured to: compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*, compute signature σ_j by signing the first element k_i and the second element k_j using a signing function associated with the Identity Based Signature Scheme and the private key sk_j, and then transmit to the first device the second element k_j and the signature σ_j. Process 400 then verifies the signature σ_j using a verification function associated with the Identity Based Signature Scheme and the identity of the second device id_j, and proceeds to compute signature σ_i by signing the second element k_j using the signing function associates with the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified. Process 400 then computes a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and then generates the common session key SK by providing the common-secret cs_i to a Key Deriving Function at step 425. Process 400 then ends.

The above is a description of embodiments of a system and process in accordance with the present invention as set forth in the following claims. It is envisioned that others may and will design alternatives that fall within the scope of the following claims.

CLAIMS:

1. A system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

a secure server configured to:

generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*;

generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device;

communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

the first device configured to:

compute a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*;

communicate to the second device the first element k_i and the identity id_i of the first device ;

the second device configured to:

compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*;

compute signature σ_j by signing the first element k_i and the second element k_j using a signing function of an Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature σ_j;

the first device configured to:

verify the signature σ_j using a verification function of the Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified;

compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and

generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

2. The system according to claim 1 wherein the first device is further configured to communicate the signature σ_i to the second device, whereby the system further comprises:

the second device being configured to:

verify the signature σ_i using the verification function of the Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified,

compute a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*, and generate the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

3. The system according to claims 1 or 2, wherein

the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve and s is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = a \cdot P$;

the second element k_j is defined as $k_j = b \cdot P$; and

the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* defined over the elliptic curve and cs_i is defined as $cs_i = e(b \cdot P, s \cdot P)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot P)^b$.

4. The system according to claim 3 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices;

compute a common-secret cs_{kgc} based on the first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot P)^s$.

5. The system according to claims 1 or 2 wherein

the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = a \cdot P$;

the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and

the common-secret cs_i and the common secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where G is cyclic group with prime order q defined over the elliptic curve, and cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

6. The system according to claim 5 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices; and

compute a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* and the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a \cdot P, b \cdot Q)^s$.

7. A system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

a secure server configured to:

generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*;

generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device;

communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

the first device configured to:

compute a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i;

communicate to the second device the first element k_i and the identity id_i of the first device ;

the second device configured to:

compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i;

compute a signature σ_j by signing a part of the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature σ_j;

the sever configured to:

verify the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and

compute signature σ_i by signing a part of the second element k_j and a part of the private key sk_j as combined with the random value a using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified;

compute a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ); and

generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

8. The system according to claim 7 wherein the first device is further configured to communicate the part of the private key sk_j as combined with the random value a and the signature σ_i to the second device, whereby the system further comprises:

the second device being configured to:

verify the signature σ_i using the verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified,

compute a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, a part of the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( ); and

generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function.

9. The system according to any one of claims 7 or 8, wherein

the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*;

the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and *H*() is a collision-resistant hash function;

the first element k_i is defined as $k_i = (R_i, g^a)$;

the second element k_j is defined as $k_j = ({R_i}^b, g^b)$;

the two input function f( )comprises a concatenation function or an exclusive-OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j,R_j)}, {(g^b)^{s_i}}/{R_i{}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i,R_i)}, {(g^a)^{s_j}}/{R_j{}^a})$.

10. The system according to claim 9 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices; and

compute a common-secret cs_{kgc} by providing the master secret key *msk*, a part of the first element k_i, the identity of the second device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

11. A system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

a secure server configured to:

generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*;

generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device;

communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

the first device configured to:

compute a first element k_i based on a first random value *a* generated by the first device;

communicate to the second device the first element k_i and the identity id_i of the first device ;

the second device configured to:

compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device;

compute a second element k_j based on the element u_j and the second random value *b*;

generate a signature σ_j by signing the first element k_i and the second element k_j using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature σ_j;

the first device configured to:

verify the signature σ_j using a verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute a signature σ_i by signing a part of the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified;

compute an element u_i based on the part of the second element k_j and the first random value a;

compute a common-secret cs_i based on the element u_i and the master public key *mpk*; and

generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

12. The system according to claim 11 wherein the first device is further configured to communicate the signature σ_i to the second device, whereby the system further comprises:

the second device being configured to:

verify the signature σ_i using the verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified,

compute a common-secret cs_j based on the element u_j and the master public key *mpk;* and

generate the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

13. The system according to any one of claims 11 or 12, wherein

the master public key mpk comprises g^x where g is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = g^a$;

the second element k_j is defined as $k_j = (U, g^b)$ where U is defined as $U = g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$;

the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

14. The system according to claim 13 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices;

compute a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein cs_{kgc} is defined as $cs_{kgc} = U^x$.

15. A system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

a secure server configured to:

generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*;

generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device;

communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

the first device configured to:

compute a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*;

compute signature σ_i by signing the first element k_i using a signing function of an Identity Based Signature Scheme and the private key sk_i;

communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device;

the second device configured to:

verify the signature σ_i using a verification function of the Identity Based Signature Scheme and the identity of the first device id_i;

compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*, when the signed first element k_i is verified;

compute a signature σ_j by signing the first element k_i and the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature σ_j;

the first device configured to:

verify the signature σ_j using the verification function of the Identity Based Signature Scheme and the identity of the second device id_j and compute a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*, when the signature σ_j is verified;

generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function;

compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and

generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

16. The system according to claim 15 wherein the first device is further configured to communicate the verification data vd_i to the second device, whereby the system further comprises:

the second device being configured to:

compute a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*;

generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function;

compute a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF);

determine if the verification data vd_i matches with the verification data vd_j; and

generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

17. The system according to claims 15 or 16, wherein

the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* and s is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = a \cdot P$;

the second element k_j is defined as $k_j = b \cdot P$; and

the common-secret cs_i and the common secret cs_j are computed using a symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order q and where cs_i is defined as $cs_i = e(s \cdot P, b \cdot P)^a$ and cs_j is defined as $cs_j = e(s \cdot P, a \cdot P)^b$.

18. The system according to claim 17 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices;

compute a common-secret cs_{kgc} based on the first element k_i, the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.P)^s$.

19. The system according to claims 15 or 16, wherein

the master public key mpk comprises (s·Q, Q, P), where P is a generator of cyclic group G_1 with prime order q, Q is a generator of cyclic group G_2 with prime order q and s is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = a \cdot P$;

the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and

the common-secret cs_i and the common secret cs_j are computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$ and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

20. The system according to claim 19 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices;

compute a common-secret cs_{kgc} based on the first element k_i, a part of the second element k_j and a parameter s obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = e(a.P, b.Q)^s$.

21. A system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

a secure server configured to:

generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*;

generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device;

communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

the first device configured to:

compute a first element k_i based on a first random value *a* generated by the first device, and the private key sk_i;

compute a signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i;

communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device;

the second device configured to:

verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute a second element k_j based on a second random value *b* generated by the second device, and the first element k_i, when the signature σ_i is verified;

compute a signature σ_j by signing a part of the first element k_i and the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature $\sigma_{j;}$

the first device configured to:

verify the signature σ_j using the verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j;

compute a common-secret cs_i by providing the master public key *mpk*, the first random value a, the identity of the second device id_j, a part of the private key sk_j, the second element k_j and a part of the private key sk_i to a two input function f( ), when the signature σ_j is verified;

generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function;

compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and

generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

22. The system according to claim 21 wherein the first device is further configured to communicate the verification data vd_i and the part of the private key sk_j as combined with the random value a to the second device, whereby the system further comprises:

the second device being configured to:

compute a common-secret cs_j by providing the master public key *mpk*, the second random value b, the identity of the first device id_i, the first element k_i, the private key sk_j and the communicated part of the private key sk_j as combined with the random value a to a two input function f( );

generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function;

compute a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF);

determine if the verification data vd_i matches with the verification data vd_j; and

generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

23. The system according to claims 21 or 22, wherein

the master public key mpk comprises g^x where *g* is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*;

the private key sk_i is defined as $sk_i = (R_i = g^{r_i}, s_i = r_i + xH(R_i, id_i))$ and the private key sk_j is define as $sk_j = (R_j = g^{r_j}, s_j = r_j + xH(R_j, id_j))$ where r_i and r_j are random numbers and *H*() is a collision-resistant hash function;

the first element k_i is defined as $k_i = (g^a)$;

the second element k_j is defined as $k_j = ({R_i}^b, g^b)$;

the two input function f( )comprises a concatenation function or an exclusive-OR function where cs_i is defined as $cs_i = f(y^{a.h(id_j, R_j)}, {(g^b)^{s_i}}/{{R_i}^b})$ and cs_j is defined as $cs_j = f(y^{b.h(id_i, R_i)}, {(g^a)^{s_j}}/{{R_j}^a})$.

24. The system according to claim 23 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices; and

compute a common-secret cs_{kgc} by providing the master secret key *msk*, the first element k_i, the identity of the second device id_j with a part of the private key sk_j, a part of the second element k_j, the identity of the first device id_i with a part of the private key sk_i, to a two input function f( ) wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = f(y^{a.h(id_j, R_j)}, y^{b.h(id_i, R_i)})$.

25. A system for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

a secure server configured to:

generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*;

generate a private key sk_i based on an identity id_i of the first device and generate a private key sk_j based on an identity id_j of the second device;

communicate the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

the first device configured to:

compute a first element k_i based on a first random value *a* generated by the first device;

compute signature σ_i by signing the first element k_i using a signing function of a self-certified Identity Based Signature Scheme and the private key sk_i;

communicate to the second device the signature σ_i, the first element k_i and the identity id_i of the first device;

the second device configured to:

verify the signature σ_i using a verification function of the self-certified Identity Based Signature Scheme and the identity of the first device id_i, and compute an element u_j based on the first element k_i and on a second random value *b* generated by the second device, when the signature σ_i is verified;

compute a second element k_j based on the element *u* and the second random value *b*;

compute signature σ_j by signing the first element k_i and the second element k_j using the signing function of the self-certified Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature σ_j;

the first device configured to:

verify the signature σ_j using the verification function of the self-certified Identity Based Signature Scheme and the identity of the second device id_j, and compute an element u_i based on a part of the second element k_j and the first random value a, when the signature σ_j is verified;

compute a common-secret cs_i based on the element u_i and the master public key *mpk*;

generate a verification key vk_i by providing the common-secret cs_i to a Key Deriving Function;

compute a verification data vd_i by providing the verification key vk_i to a Authentication Data Deriving function (AdDF); and

generate the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

26. The system according to claim 25 wherein the first device is further configured to communicate the verification data vd_i to the second device, whereby the system further comprises:

the second device being configured to:

compute a common-secret cs_j based on the element u_j and the master public key *mpk*;

generate a verification key vk_j by providing the common-secret cs_j to the Key Deriving Function;

compute a verification data vd_j by providing the verification key vk_j to the Authentication Data Deriving function (AdDF);

determine if the verification data vd_i matches with the verification data vd_j; and

generate the common session key SK by providing the common-secret cs_j to a Key Deriving Function, when the verification data vd_i matches with the verification data vd_j.

27. The system according to claims 25 or 26, wherein

the master public key mpk comprises g^x where g is a generator of a cyclic multiplicative group G and x is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = g^a$;

the second element k_j is defined as $k_j = (U, g^b)$ where U is defined as $U = g^{u_j}$, and the element u_j is defined as $u_j = g^{ab}$;

the common-secret cs_i is defined as $cs_i = g^{xu_i}$ and the common secret cs_j is defined as $cs_j = g^{xu_j}$ where the element u_i is defined as $u_i = g^{ab}$.

28. The system according to claim 27 wherein the secure server is further configured to:

receive all communications and transmissions exchanged between the first and second devices;

compute a common-secret cs_{kgc} based on the element u_j *or* the element u_i and a parameter x obtained from the master secret key *msk* wherein the common-secret cs_{kgc} is defined as $cs_{kgc} = U^x$.

29. A method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

generating, by a secure server, a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

receiving a first element k_i and a first random value *a* from the first device; wherein the a first element k_i is computed based on the first random value *a* generated by the first device;

receiving second element k_j and a second random value b from the second device; wherein the second element k_j is computed based on the second random value b generated by the second device;

computing a common-secret cs_{kgc} based on the first element k_i, the second element k_j, the first random value *a,* the second random value *b* and a parameter s obtained from the master secret key *msk.*

30. The method according to claim 29 , wherein

the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve;

the first element k_i is defined as $k_i = a \cdot P$; and

the second element k_j is defined as $k_j = b \cdot P$.

31. The method according to claim 29, wherein

the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve, and *Q* is a generator of cyclic group G_2 with prime order *q*;

the first element k_i is defined as $k_i = a \cdot P$; and

the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$.

32. A server for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

a processor; and

a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to:

generate a master secret key *msk* and a master public key *mpk*, wherein the master secret key *msk* corresponds to the master public key *mpk*, a private key sk_i based on an identity id_i of the first device, a private key sk_j based on an identity id_j of the second device, and communicating the master public key *mpk* and the private key sk_i to the first device, and the master public key *mpk* and the private key sk_j to the second device;

receive a first element k_i and a first random value *a* from the first device; wherein the a first element k_i is computed based on the first random value *a* generated by the first device;

receive second element k_j and a second random value b from the second device; wherein the second element k_j is computed based on the second random value b generated by the second device;

compute a common-secret cs_{kgc} based on the first element k_i, the second element k_j, the first random value *a,* the second random value *b* and a parameter s obtained from the master secret key *msk.*

33. A method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

receiving, by the first device, a master public key *mpk* and a private key sk_i from a secure server;

computing, by the first device, a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*;

communicating, by the first device, to the second device the first element k_i and the identity id_i of the first device;

receiving a second element k_j and a signature σ_j from the second device;

verifying, by the first device, the signature σ_j using a verification function of the Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified;

computing, by the first device, a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and

generating, by the first device, the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

34. The method according to claim 33 wherein

the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = a \cdot P$;

the common-secret cs_i is computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where G is cyclic group with prime order q defined over the elliptic curve, and cs_i is defined as $cs_i = e(b \cdot P, s \cdot Q)^a$.

35. The method according to claims 33, wherein

the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve and s is a parameter obtained from the master secret key *msk*;

the first element k_i is defined as $k_i = a \cdot P$; and

the common-secret cs_i is computed using a symmetric bilinear map e: $G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* defined over the elliptic curve and cs_i is defined as $cs_i = e(b \cdot P, s \cdot P)^a$.

36. A first device *i* for generating a common session key SK for encoding digital communications between the first device *i* and a second device *j*, comprising:

a processor; and

a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to:

receive, by the first device, a master public key *mpk* and a private key sk_i from a secure server;

compute, by the first device, a first element k_i based on a first random value *a* generated by the first device, and the master public key *mpk*;

communicate, by the first device, to the second device the first element k_i and the identity id_i of the first device;

receive a second element k_j and a signature σ_j from the second device;

verify, by the first device, the signature σ_j using a verification function of the Identity Based Signature Scheme and the identity of the second device id_j, and compute signature σ_i by signing the second element k_j using the signing function of the Identity Based Signature Scheme and the private key sk_i, when the signature σ_j is verified;

compute, by the first device, a common-secret cs_i based on the second element k_j, the master public key *mpk*, and the first random value *a*; and

generate, by the first device, the common session key SK by providing the common-secret cs_i to a Key Deriving Function.

37. A method for generating a common session key SK for encoding digital communications between a first device *i* and a second device *j*, comprising:

receiving, by the second device, a master public key *mpk* and a private key from a secure server;

receiving a first element k_i and the identity id_i of the first device from the first device i*;*

compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*;

compute signature σ_j by signing the first element k_i and the second element k_j using a signing function of an Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature σ_j;

receiving a signature σ_i from the first device;

verifying, by the second device, the signature σ_i using the verification function of the Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified,

computing a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*, and generating the common session key SK by providing the common-secret cs_j to the Key Deriving Function.

38. The method according to claim 37, wherein

the master public key mpk comprises (P, s·P) where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve and s is a parameter obtained from the master secret key *msk*;

the second element k_j is defined as $k_j = b \cdot P$; and

the common secret cs_j is computed using a symmetric bilinear map $e: G_1 \times G_1 \rightarrow G_2$ where G_2 is a cyclic group with prime order *q* defined over the elliptic curve and cs_j is defined as $cs_j = e(a \cdot P, s \cdot P)^b$.

39. The method according to claim 37 wherein

the master public key mpk comprises (s·Q, Q, P), where *P* is a generator of cyclic group G_1 with prime order *q* defined over an elliptic curve, *Q* is a generator of cyclic group G_2 with prime order *q* and s is a parameter obtained from the master secret key *msk*;

the second element k_j is defined as $k_j = (b \cdot P, b \cdot Q)$; and

the common secret cs_j is computed using an asymmetric bilinear map e: $G_1 \times G_2 \rightarrow G$ where G is cyclic group with prime order q defined over the elliptic curve, and cs_j is defined as $cs_j = e(a \cdot P, s \cdot Q)^b$.

40. A second device *j* for generating a common session key SK for encoding digital communications between the first device *i* and the second device *j*, comprising:

a processor; and

a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to:

receiving, by the second device, a master public key *mpk* and a private key from a secure server;

receiving a first element k_i and the identity id_i of the first device from the first device *I;*

compute a second element k_j based on a second random value *b* generated by the second device, and the master public key *mpk*;

compute signature σ_j by signing the first element k_i and the second element k_j using a signing function of an Identity Based Signature Scheme and the private key sk_j;

transmit to the first device the second element k_j and the signature σ_j;

receive a signature σ_i from the first device;

verify, by the second device, the signature σ_i using the verification function of the Identity Based Signature Scheme and the identity of the first device id_i, and when the signature σ_i is verified,

comput a common-secret cs_j based on the first element k_i, the master public key *mpk*, and the second random value *b*, and generating the common session key SK by providing the common-secret cs_j to the Key Deriving Function.