

SYSTEM AND METHOD FOR CONSTRUCTING A SELF-AUTHENTICATING MESSAGE USING IDENTITY-BASED CRYPTOGRAPHY

Field of the Invention

This invention relates to a system and method for constructing a self-authenticating message that is to be sent from a first electronic device to a second electronic device. In particular, this invention relates to a system and method for constructing a self-authenticating message using identity-based cryptography whereby plaintext is encrypted and decrypted using encryption/decryption keys that were generated using identities of the sender and/or receiver.

Summary of the Prior Art

As technology advances, more and more devices seamlessly communicate wirelessly with one another through the Internet or directly between themselves. This convergence has resulted in the vision of the Internet of Things ("IoT") gaining more traction through recent years. In Internet of Things networks, millions of entities or devices (i.e. Things) will be connected to one another. These devices, which may include 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 indirectly via the Internet.

At the basic level, devices provided in IoT networks communicate with each other by exchanging secure messages through wireless or wired networks. However, such messages are easily intercepted by malicious entities. Hence, instead of expanding resources on preventing messages from being intercepted, the content of these messages may instead be encrypted from plaintext into ciphertext before these messages are sent. An authorized recipient would then decrypt the ciphertext to read the plaintext contained within.

Presently, IoT devices communicate securely by establishing a secure tunnel between a sender and a recipient. In order to establish the secure tunnel, a sender will first authenticate the recipient and subsequently, the recipient will authenticate the sender. Once this is done, both parties will then establish and share a common session key that is utilized to encrypt and decrypt messages exchanged between these two parties.

In addition to this secure tunnel method, another method of securing communications between two parties involves the use of self-authenticating messages. A self-authenticating message is essentially an encrypted message that has been appended with an authentication tag. A recipient of the message will utilize the authentication tag appended to

the received message to determine the veracity of the encrypted message - that is whether the message has been tampered with and/or whether the message is sent from the expected sender. Self-authenticating messages are typically constructed using symmetric key encryption or asymmetric key encryption techniques.

For the symmetric key encryption approach, prior to the transmission of a series of messages between two devices, these devices will both have to pre-share or pre-exchange symmetric keys. For such a method, data messages that are to be transmitted between the entities will be attached with an authentication code which is generated using this pre-shared or pre-agreed-upon secret key. If the receiving entity correctly verify the authentication code attached with 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 decrypt the received 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 IoT networks, 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 pre-shared symmetric key beforehand.

For the asymmetric key encryption approach, the self-authentication message is encrypted using a public encryption algorithm whereby a signature generated by this algorithm acts as the authentication tag. However, the signature generated by this public key algorithm is typically quite large in size and as such, causes the size of the self-authenticated message to greatly expand. It is not practical to transmit such large messages between IoT devices hence asymmetric key encryption approaches are typically not suitable for such use in IoT networks.

For the above reasons, those skilled in the art are constantly striving to come up with a system and method to construct a self-authenticating message that does not require the pre-sharing of session keys between communicating parties and also does not result in a huge expansion in the size of the resulting transmitted message.

Summary of the Invention

Systems and methods to improve the construction of a self-authenticating message using identity-based cryptography are proposed and these systems and methods are set out below in accordance with embodiments of the invention.

A first improvement proposed by embodiments of systems and methods in accordance with the invention is that a symmetric key does not need to be pre-shared between communicating devices and that the final constructed self-authenticating message does not greatly expand in size.

A second improvement proposed by embodiments of systems and methods in accordance with the invention is that the computation of an intermediate key may be outsourced to other devices and need not necessarily be carried out within the processors of the communicating devices.

The above improvements 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 method for constructing a self-authenticating message to be sent from a first device to a second device comprises obtaining, by the first device, a first partial key and a second partial key for the first device; computing an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; generating a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device ; generating an encryption key K using the shared key K_{SR} ; encrypting a message using the generated encryption key K ; forming the self-authenticating message based at least on the encrypted message; and transmitting the formed self-authenticated message to the second device.

With reference to the first aspect, in a first possible implementation manner of the first aspect, the forming the self-authenticating message based at least on the encrypted message comprises: calculating, by the first device, a first Message Authentication Code (MAC) using the encryption key K and the encrypted message; and forming the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC.

With reference to the first aspect or the first possible implementation manner of the first aspect, in a second possible implementation manner of the first aspect, the obtaining the first partial key and the second partial key for the first device comprises: transmitting, by the first device, the identity of the first device ID_1 to a private key generator such that the generator generates a private key K_{first_device} based on a master secret key s and the identity of the first device ID_1 ; receiving the private key K_{first_device} from the private key generator; and computing the first partial key and the second partial key for the first device whereby the

second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{first_device} .

With reference to the first aspect, the first or second possible implementation manners of the first aspect, in a third possible implementation manner of the first aspect, the computing the intermediate key IK_{SR} by the first device using the identity of the second device ID_2 and the first partial key of the first device comprises: transmitting, by the first device, the identity of the second device ID_2 and the first partial key of the first device to a computer server such that the computer server computes the intermediate key IK_{SR} using the identity of the second device ID_2 and the first partial key of the first device; and receiving the computed intermediate key IK_{SR} from the computer server.

With reference to the first aspect, the first, second or third possible implementation manners of the first aspect, in a fourth possible implementation manner of the first aspect, the step of generating, by the first device, the encryption key K using the shared key K_{SR} , this step further comprises: using a generated cryptographic nonce together with the shared key K_{SR} to generate the encryption key K , wherein at the step of forming the self-authenticating message, this step further comprises adding, by the first device, the cryptographic nonce to the self-authenticating message.

According to a second aspect of the invention, a method for constructing a self-authenticating message to be sent from a first device to a second device comprises: obtaining, by the second device, a first partial key and a second partial key for the second device; receiving the self-authenticated message transmitted from the first device; computing an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device; generating a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device; generating an encryption key K using the shared key K_{SR} ; authenticating the received self-authenticating message using the encryption key K ; and decrypting an encrypted message in the received self-authenticating message using the generated encryption key K , when the received self-authenticating message is authenticated successfully.

With reference to the second aspect, in a first possible implementation manner of the second aspect, before the step of authenticating the received self-authenticating message using the encryption key K , the method further comprises: calculating, by the second device, a second MAC using the generated encryption key K and the encrypted message from the received self-authenticating message, wherein the step of authenticating the received self-

authenticating message using the encryption key K , comprises: determining, by the second device, whether the second MAC matches with a first MAC obtained from the received self-authenticating message or not; authenticating the self-authenticating message successfully when the second MAC matches with the obtained first MAC.

With reference to the second aspect or the first possible implementation manner of the second aspect, in a second possible implementation manner of the second aspect, the obtaining the first partial key and the second partial key for the second device comprises: transmitting, by the second device, an identity of the second device ID_2 to a private key generator such that the private key generator generates a private key K_{second_device} based on a master secret key s and the identity of the second device ID_2 ; receiving the private key K_{second_device} from the private key generator; and computing the first partial key and the second partial key for the second device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{second_device} .

With reference to the second aspect, the first or second possible implementation manners of the second aspect, in a third possible implementation manner of the second aspect, the computing the intermediate key IK_{RS} by the second device using the identity of the first device ID_1 and the first partial key of the second device comprises: transmitting, by the second device, the identity of the first device ID_1 and the first partial key of the second device to a computer server such that the computer server computes the intermediate key IK_{RS} using the identity of the first device ID_1 and the first partial key of the second device; and receiving the computed intermediate key IK_{RS} from the computer server.

With reference to the second aspect, the first, second or third possible implementation manners of the second aspect, in a fourth possible implementation manner of the second aspect, at the step of generating, by the second device, the encryption key K using the shared key K_{SR} , the method further comprises: retrieving, by the second device, a cryptographic nonce from the received self-authenticating message; and generating, by the second device, the encryption key K , using the retrieved cryptographic nonce and the shared key K_{SR} .

According to a third aspect of the invention, a method for constructing a self-authenticating message to be sent from a first device to a second device comprises: obtaining, by the first device, a first partial key and a second partial key for the first device; computing an intermediate key IK_{SR} using an identity of the second device ID_2 and the first

partial key of the first device; generating a shared key K_{SR} based on the computed intermediate key IK_{SR} and the second partial key of the first device; generating an encryption key K using the shared key K_{SR} together with a first constant and encrypting a message by the generated encryption key K , and generating an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and the second constants are obtained from the first device; calculating a first Message Authentication Code (MAC) using the authentication key K_A and the encrypted message; forming the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC; and transmitting the formed self-authenticated message to the second device.

With reference the third aspect, in a first possible implementation manner of the third aspect of the invention, prior to the step of generating the encryption key K and the authentication key K_A , the method comprises the step of: generating, by the first device, a cryptographic nonce, wherein the encryption key K is derived as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A is derived as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$, where $\text{HMAC}()$ is a key-hash message authentication code function.

According to a fourth aspect of the invention, a method for constructing a self-authenticating message to be sent from a first device to a second device comprises: obtaining, by the second device, a first partial key and a second partial key for the second device; receiving the self-authenticating message transmitted from the first device; computing an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device, and using the computed intermediate key IK_{RS} and the second partial key of the second device to generate a shared key K_{SR} ; computing an encryption key K using the shared key K_{SR} and a first constant and an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and second constants are obtained from the second device, calculating a second MAC using the computed authentication key K_A and an encrypted message from the received self-authenticating message, determining if the second MAC matches with a first MAC obtained from the received self-authenticating message; and decrypting the encrypted message in the received self-authenticating message using the generated encryption key K , when the second MAC matches with the obtained first MAC.

With reference to the fourth aspect, in a first possible implementation manner of the fourth aspect, at the step of computing, by the second device, the encryption key K and the authentication key K_A , the encryption key K is derived as: $K = \text{HMAC}(K_{SR}, \text{cryptographic}$

nonce || first constant), and the authentication key K_A is derived as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} || \text{second constant})$ whereby the first and second constants are obtained from the received self-authenticating message.

According to a fifth aspect of the invention, a method for constructing a self-authenticating message to be sent from a first device to a second device comprises: obtaining, by the first device, a first partial key and a second partial key for the first device; computing an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; generating a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device; generating an encryption key K using the shared key K_{SR} ; calculating a first Message Authentication Code (MAC) using the encryption key K and a message; and forming the self-authenticating message based at least on the identity of the first device ID_1 , the message and the first MAC; and transmitting the formed self-authenticated message to the second device.

According to a sixth aspect of the invention, a method for constructing a self-authenticating message to be sent from a first device to a second device comprises: obtaining, by the second device, a first partial key and a second partial key for the second device; receiving the self-authenticated message transmitted from the first device; computing an intermediate key IK_{RS} using an identity of a first device ID_1 and the first partial key of the second device; generating a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device; computing an encryption key K , and calculating a second MAC using the generated encryption key K and a message from the received self-authenticating message; determining if the second MAC matches with a first MAC obtained from the received self-authenticating message; and authenticating the self-authenticating message when the second MAC matches with the obtained first MAC.

According to a seventh aspect of the invention, a system is provided for constructing a self-authenticating message to be sent from a first device to a second device, the system comprising: a processor provided within the first device; and a non-transitory media readable by the processor provided within the first device, the media storing instructions that when executed by the processor, cause the processor to: obtain a first partial key and a second partial key for the first device; compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; generate a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device ; generate an encryption key K using the shared key K_{SR} ; encrypt a message using the

generated encryption key K ; form the self-authenticating message based at least on the encrypted message; and transmit the formed self-authenticated message to the second device.

With reference to the seventh aspect, in a first possible implementation manner of the seventh aspect, the instructions to form the self-authenticating message based at least on the encrypted message comprises: instructions for directing the processor to: calculate a first Message Authentication Code (MAC) using the encryption key K and the encrypted message; and form the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC.

With reference to the seventh aspect or the first possible implementation manner of the seventh aspect, in a second possible implementation manner of the seventh aspect, the instructions to obtain the first partial key and the second partial key for the first device comprises: instructions for directing the processor to: transmit the identity of the first device ID_1 to a private key generator such that the generator generates a private key K_{first_device} based on a master secret key s and the identity of the first device ID_1 ; receive the private key K_{first_device} from the private key generator; and compute the first partial key and the second partial key for the first device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{first_device} .

With reference to the seventh aspect, the first or second possible implementation manners of the seventh aspect, in a third possible implementation manner of the seventh aspect, the instructions for directing the processor to compute the intermediate key IK_{SR} by the first device using the identity of the second device ID_2 and the first partial key of the first device comprises: instructions for directing the processor to: transmit the identity of the second device ID_2 and the first partial key of the first device to a computer server such that the computer server computes the intermediate key IK_{SR} using the identity of the second device ID_2 and the first partial key of the first device; and receive the computed intermediate key IK_{SR} from the computer server.

With reference to the seventh aspect, the first, second or third possible implementation manners of the seventh aspect, in a fourth possible implementation manner of the seventh aspect, the instructions for directing the processor to generate the encryption key K using the shared key K_{SR} , further comprises: instructions for directing the processor to: use a generated cryptographic nonce together with the shared key K_{SR} to generate the

encryption key K , wherein the instructions for directing the processor to form the self-authenticating message further comprises instructions for directing the processor to add the cryptographic nonce to the self-authenticating message.

According to an eighth aspect of the invention, a system is provided for constructing a self-authenticating message to be sent from a first device to a second device comprising: a processor provided within the second device; and a non-transitory media readable by the processor provided within the second device, the media storing instructions that when executed by the processor, cause the processor to: obtain a first partial key and a second partial key for the second device; receive the self-authenticated message transmitted from the first device; compute an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device; generate a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device; generate an encryption key K using the shared key K_{SR} ; authenticate the received self-authenticating message using the encryption key K ; and decrypt an encrypted message in the received self-authenticating message using the generated encryption key K , when the received self-authenticating message is authenticated successfully.

With reference to the eighth aspect, in a first possible implementation manner of the eighth aspect, before the instructions for directing the processor to authenticate the received self-authenticating message using the encryption key K , the system further comprises: instructions for directing the processor to calculate a second MAC using the generated encryption key K and the encrypted message from the received self-authenticating message, wherein the instructions for directing the processor to authenticate the received self-authenticating message using the encryption key K , comprises: instructions for directing the processor to: determine whether the second MAC matches with a first MAC obtained from the received self-authenticating message or not; and authenticate the self-authenticating message successfully when the second MAC matches with the obtained first MAC.

With reference to the eighth aspect or the first possible implementation manner of the eighth aspect, in a second possible implementation manner of the eighth aspect, the instructions for directing the processor to obtain the first partial key and the second partial key for the second device comprises: instructions for directing the processor to: transmit an identity of the second device ID_2 to a private key generator such that the private key generator generates a private key K_{second_device} based on a master secret key s and the identity of the second device ID_2 ; receive the private key K_{second_device} from the private key generator; and compute the first partial key and the second partial key for the second device

whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key $K_{\text{second_device}}$.

With reference to the eighth aspect, the first or second possible implementation manners of the eighth aspect, in a third possible implementation manner of the eighth aspect, the instructions for directing the processor to compute the intermediate key IK_{RS} by the second device using the identity of the first device ID_1 and the first partial key of the second device comprises: instructions for directing the processor to: transmit the identity of the first device ID_1 and the first partial key of the second device to a computer server such that the computer server computes the intermediate key IK_{RS} using the identity of the first device ID_1 and the first partial key of the second device; and receive the computed intermediate key IK_{RS} from the computer server.

With reference to the eighth aspect, the first, second or third possible implementation manners of the eighth aspect, in a fourth possible implementation manner of the eighth aspect, the instructions for directing the processor to generate the encryption key K further comprises: instructions for directing the processor to: retrieve a cryptographic nonce from the received self-authenticating message; and generate the encryption key K , using the retrieved cryptographic nonce and the shared key K_{SR} .

According to a ninth aspect of the invention, a system is provided for constructing a self-authenticating message to be sent from a first device to a second device comprising: a processor provided within the first device; and a non-transitory media readable by the processor provided within the first device, the media storing instructions that when executed by the processor, cause the processor to: obtain a first partial key and a second partial key for the first device; compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; generate a shared key K_{SR} based on the computed intermediate key IK_{SR} and the second partial key of the first device; generate an encryption key K using the shared key K_{SR} together with a first constant and encrypting a message by the generated encryption key K , and generating an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and the second constants are obtained from the first device; calculate a first Message Authentication Code (MAC) using the authentication key K_A and the encrypted message; form the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC; and transmit the formed self-authenticated message to the second device.

With reference the ninth aspect, in a first possible implementation manner of the ninth aspect of the invention, prior to the instructions for directing the processor to generate the encryption key K and the authentication key K_A , the system comprises: instructions for directing the processor to: generate a cryptographic nonce, wherein the encryption key K is derived as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A is derived as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$, where $\text{HMAC}()$ is a key-hash message authentication code function.

According to a tenth aspect of the invention, a system is provided for constructing a self-authenticating message to be sent from a first device to a second device comprising: a processor provided within the second device; and a non-transitory media readable by the processor provided within the second device, the media storing instructions that when executed by the processor, cause the processor to: obtain a first partial key and a second partial key for the second device; receive the self-authenticating message transmitted from the first device; compute an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device, and using the computed intermediate key IK_{RS} and the second partial key of the second device to generate a shared key K_{SR} ; compute an encryption key K using the shared key K_{SR} and a first constant and an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and second constants are obtained from the second device, calculate a second MAC using the computed authentication key K_A and an encrypted message from the received self-authenticating message, determine if the second MAC matches with a first MAC obtained from the received self-authenticating message; and decrypt the encrypted message in the received self-authenticating message using the generated encryption key K , when the second MAC matches with the obtained first MAC.

With reference to the tenth aspect, in a first possible implementation manner of the tenth aspect, the instructions for directing the processor to compute the encryption key K and the authentication key K_A , comprises: instructions for directing the processor to: derive the encryption key K as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$ whereby the first and second constants are obtained from the received self-authenticating message.

According to a eleventh aspect of the invention, a method for constructing a self-authenticating message to be sent from a first device to a second device comprises, a system is provided for constructing a self-authenticating message to be sent from a first

device to a second device comprising: a processor provided within the first device; and a non-transitory media readable by the processor provided within the first device, the media storing instructions that when executed by the processor, cause the processor to: obtain a first partial key and a second partial key for the first device; compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; generate a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device; generate an encryption key K using the shared key K_{SR} ; calculate a first Message Authentication Code (MAC) using the encryption key K and a message; and form the self-authenticating message based at least on the identity of the first device ID_1 , the message and the first MAC; and transmit the formed self-authenticated message to the second device.

According to a twelfth aspect of the invention, a system is provided for constructing a self-authenticating message to be sent from a first device to a second device comprising: a processor provided within the second device; and a non-transitory media readable by the processor provided within the second device, the media storing instructions that when executed by the processor, cause the processor to: obtain a first partial key and a second partial key for the second device; receive the self-authenticated message transmitted from the first device; compute an intermediate key IK_{RS} using an identity of a first device ID_1 and the first partial key of the second device; generate a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device; compute an encryption key K , and calculating a second MAC using the generated encryption key K and a message from the received self-authenticating message; determine if the second MAC matches with a first MAC obtained from the received self-authenticating message; and authenticate the self-authenticating message when the second MAC matches with the obtained first MAC.

According to a thirteenth aspect of the invention, a first device is provided for constructing a self-authenticating message to be sent from the first device to a second device, the first device comprising an obtaining unit, a computing unit, a generating unit, an encrypting unit, a forming unit and a transmitting unit whereby the obtaining unit is configured to obtain a first partial key and a second partial key for the first device; the computing unit is configured to compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; the generating unit is configured to generate a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device, and generate an encryption key K using the shared key K_{SR} ; the encrypting unit is configured to encrypt a message using the generated encryption key K ;

the transmitting unit is configured to form the self-authenticating message based at least on the encrypted message; and the transmitting unit is configured to transmit the formed self-authenticated message to the second device.

With reference to the thirteenth aspect, in a first possible implementation manner of the thirteenth aspect of the invention, the forming unit is further configured to calculate a first Message Authentication Code (MAC) using the encryption key K and the encrypted message; and form the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC.

With reference to the thirteenth aspect or the first possible implementation manner of the thirteenth aspect of the invention, in a second possible implementation manner of the thirteenth aspect of the invention, the obtaining unit configured to obtain the first partial key and the second partial key for the first device is further configured to transmit the identity of the first device ID_1 to a private key generator such that the generator generates a private key K_{first_device} based on a master secret key s and the identity of the first device ID_1 ; receive the private key K_{first_device} from the private key generator; and compute the first partial key and the second partial key for the first device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{first_device} .

With reference to the thirteenth aspect, the first or second possible implementation manner of the thirteenth aspect of the invention, in a third possible implementation manner of the thirteenth aspect of the invention, the computing unit configured to compute the intermediate key IK_{SR} by the first device using the identity of the second device ID_2 and the first partial key of the first device is further configured to: transmit the identity of the second device ID_2 and the first partial key of the first device to a computer server such that the computer server computes the intermediate key IK_{SR} using the identity of the second device ID_2 and the first partial key of the first device; and receive the computed intermediate key IK_{SR} from the computer server.

With reference to the thirteenth aspect, the first, second or third possible implementation manner of the thirteenth aspect of the invention, in a fourth possible implementation manner of the thirteenth aspect of the invention, the generating unit configured to generate the encryption key K using the shared key K_{SR} , is further configured to: use a generated cryptographic nonce together with the shared key K_{SR} to generate the encryption key K , wherein the instructions for directing the processor to form the self-

authenticating message further comprises instructions for directing the processor to add the cryptographic nonce to the self-authenticating message.

According to a fourteenth aspect of the invention, a second device is provided for receiving a self-authenticating message to be sent from a first device to the second device, the second device comprising an obtaining unit, a receiving unit, a computing unit, a generating unit, an authenticating unit, and a decrypting unit whereby the obtaining unit is configured to obtain a first partial key and a second partial key for the second device; the receiving unit is configured to receive the self-authenticated message transmitted from the first device; the computing unit is configured to compute an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device; the generating unit is configured to generate a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device and to generate an encryption key K using the shared key K_{SR} ; the authenticating unit is configured to authenticate the received self-authenticating message using the encryption key K ; and the decrypting unit is configured to decrypt an encrypted message in the received self-authenticating message using the generated encryption key K , when the received self-authenticating message is authenticated successfully.

With reference to the fourteenth aspect, in a first possible implementation manner of the fourteenth aspect, before the authenticating unit is configured to authenticate the received self-authenticating message using the encryption key K , a calculating unit is configured to: calculate a second MAC using the generated encryption key K and the encrypted message from the received self-authenticating message, wherein the authenticating unit configured to authenticate the received self-authenticating message using the encryption key K , is further configured to: determine whether the second MAC matches with a first MAC obtained from the received self-authenticating message or not; and authenticate the self-authenticating message successfully when the second MAC matches with the obtained first MAC.

With reference to the fourteenth aspect or the first possible implementation manner of fourteenth aspect, in a second possible implementation manner of the fourteenth aspect, the obtaining unit configured to obtain the first partial key and the second partial key for the second device is further configured to: transmit an identity of the second device ID_2 to a private key generator such that the private key generator generates a private key K_{second_device} based on a master secret key s and the identity of the second device ID_2 ; receive the private key K_{second_device} from the private key generator; and compute the first partial key and the

second partial key for the second device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key $K_{\text{second_device}}$.

With reference to the fourteenth aspect, the first or second possible implementation manners of fourteenth aspect, in a third possible implementation manner of the fourteenth aspect, the computing unit configured to compute the intermediate key IK_{RS} by the second device using the identity of the first device ID_1 and the first partial key of the second device is further configured to: transmit the identity of the first device ID_1 and the first partial key of the second device to a computer server such that the computer server computes the intermediate key IK_{RS} using the identity of the first device ID_1 and the first partial key of the second device; and receive the computed intermediate key IK_{RS} from the computer server.

With reference to the fourteenth aspect, the first, second or third possible implementation manners of the fourteenth aspect, in a fourth possible implementation manner of fourteenth aspect, the generating unit configured to generate the encryption key K is further configured to: retrieve a cryptographic nonce from the received self-authenticating message; and generate the encryption key K , using the retrieved cryptographic nonce and the shared key K_{SR} .

According to a fifteenth aspect, a first device is provided for constructing a self-authenticating message to be sent from the first device to a second device, the first device comprising an obtaining unit, a computing unit, a generating unit, a calculating unit, a forming unit and a transmitting unit whereby the obtaining unit is configured to obtain a first partial key and a second partial key for the first device; the computing unit is configured to compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; the generating unit is configured to generate a shared key K_{SR} based on the computed intermediate key IK_{SR} and the second partial key of the first device; the generating unit is configured to generate an encryption key K using the shared key K_{SR} together with a first constant and encrypting a message by the generated encryption key K , and generating an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and the second constants are obtained from the first device; the calculating unit is configured to calculate a first Message Authentication Code (MAC) using the authentication key K_A and the encrypted message; the forming unit is configured to form the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC; and the transmitting unit is configured to transmit the formed self-authenticated message to the second device.

With reference to the fifteenth aspect, wherein prior to the generating unit being configured to generate the encryption key K and the authentication key K_A , the generating unit is configured to: generate a cryptographic nonce, wherein the encryption key K is derived as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A is derived as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$, where $\text{HMAC}()$ is a key-hash message authentication code function.

According to a sixteenth aspect of the invention, a second device is provided for receiving a self-authenticating message to be sent from a first device to the second device, the second device comprising an obtaining unit, a receiving unit, a computing unit, a calculating unit, a determining unit, and a decrypting unit whereby the obtaining unit is configured to obtain a first partial key and a second partial key for the second device; a receiving unit configured to receive the self-authenticating message transmitted from the first device; a computing unit configured to compute an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device, and using the computed intermediate key IK_{RS} and the second partial key of the second device to generate a shared key K_{SR} ; a computing device configured to compute an encryption key K using the shared key K_{SR} and a first constant and an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and second constants are obtained from the second device, a calculating unit configured to calculate a second MAC using the computed authentication key K_A and an encrypted message from the received self-authenticating message, a determining unit configured to determine if the second MAC matches with a first MAC obtained from the received self-authenticating message; and a decrypting unit configured to decrypt the encrypted message in the received self-authenticating message using the generated encryption key K , when the second MAC matches with the obtained first MAC.

With reference to the sixteenth aspect, in a first possible implementation manner of the sixteenth aspect of the invention, the computing unit configured to compute the encryption key K and the authentication key K_A , is further configured to: derive the encryption key K as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$ whereby the first and second constants are obtained from the received self-authenticating message.

According to a seventeenth aspect of the invention, a second device is provided for receiving a self-authenticating message to be sent from a first device to the second device, the second device comprising an obtaining unit, a receiving unit, a computing unit, a calculating unit, a forming unit, and a transmitting unit whereby the obtaining unit is configured to obtain a first partial key and a second partial key for the first device; a computing unit is configured to compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device; a generating unit is configured to generate a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device and to generate an encryption key K using the shared key K_{SR} ; a calculating unit is configured to calculate a first Message Authentication Code (MAC) using the encryption key K and a message; and the forming unit is configured to form the self-authenticating message based at least on the identity of the first device ID_1 , the message and the first MAC; and the transmitting unit is configured to transmit the formed self-authenticated message to the second device.

According to an eighteenth aspect of the invention, a second device is provided for receiving a self-authenticating message to be sent from a first device to the second device, the second device comprising an obtaining unit configured to obtain a first partial key and a second partial key for the second device; a receiving unit configured to receive the self-authenticated message transmitted from the first device; a computing unit configured to compute an intermediate key IK_{RS} using an identity of a first device ID_1 and the first partial key of the second device; a generating unit configured to generate a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device; a computing unit configured to compute an encryption key K , and calculating a second MAC using the generated encryption key K and a message from the received self-authenticating message; a determining unit configured to determine if the second MAC matches with a first MAC obtained from the received self-authenticating message; and an authenticating unit configured to authenticate the self-authenticating message when the second MAC matches with the obtained first MAC.

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 an exemplary block diagram of a network in which devices are exchanging self-authenticating messages in accordance with embodiments of the invention;

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

Figure 3 illustrating a timing diagram for the generation of an intermediate key IK_{SR} using identity-based cryptography in accordance with embodiments of the invention;

Figure 4 illustrating a flow diagram showing a process for the construction of a self-authenticating message having a sender's identity, a cryptographic nonce and a Message Authentication Code in accordance with embodiments of the invention;

Figure 5 illustrating a flow diagram showing a process for the construction of a self-authenticating message having a sender's identity and a Message Authentication Code in accordance with embodiments of the invention;

Figure 6 illustrating a flow diagram showing a process for the construction of a self-authenticating message having only a cryptographic nonce in accordance with embodiments of the invention;

Figure 7 illustrating a flow diagram of a process for constructing a self-authenticating message at a sending device in accordance with embodiments of the invention; and

Figure 8 illustrating a flow diagram of a process for decrypting a received self-authenticating message at a receiving device in accordance with embodiments of the invention.

Detailed Description

This invention relates to a system and method for constructing a self-authenticating message that is to be sent from a first electronic device to a second electronic device. In particular, this invention relates to a system and method for constructing a self-authenticating message using identity-based cryptography whereby plaintext to be transmitted is first encrypted using an encryption key that was generated using the identity of the receiver and the private key of the sender. Upon receiving the encrypted message, the receiver will first authenticate the veracity of the receiving message and upon successfully authenticating the received message, the receiver will then generate a decryption key using the sender's identity and the receiver's private key. The receiver then proceeds to decrypt the message using the decryption key.

Figure 1 illustrates a block diagram of two devices which are exchanging self-authenticating messages in accordance with embodiments of the invention. The system illustrated in Figure 1 comprises electronic devices 105, 110, that are wirelessly connected

to each other either through direct means 130 or through indirect means 125 – that is via Internet 115. Electronic devices 105 and 110 may also be wirelessly connected to secure server 120 via the same direct 130 and/or indirect 125 means. Electronic devices 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 computer, 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 server 120, this server may comprise a secure cloud server or a remotely located secure server which is able to communicate wirelessly with electronic devices 105 and 110 either through Internet 115 or directly with electronic devices 105 and 110. If server 120 is configured to communicate with electronic devices 105 and 110 through Internet 115, server 120 may do so via indirect means 125. These indirect means may comprise wired networks or wireless networks 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 through direct means 130 with electronic devices 105 and 110, this may be accomplished through wireless and/or wired networks. These networks may include, but are not limited to, networks such as Wireless-Fidelity (Wi-Fi), Bluetooth, or Near Field Communication (NFC). It should be noted that electronic devices 105 and 110 may utilize indirect means 125 (via the Internet) or direct means 130 (direct communication) to exchange data messages with one another.

Figure 2 illustrates a block diagram representative of components of module 200 that is provided within electronic devices 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 module 200 may vary and Figure 2 is provided by way of example only.

In embodiments of the invention, module 200 comprises controller 201 and user interface 202. User interface 202 is arranged to enable manual interactions between a user and module 200 and for this purpose includes the input/output components required for the user to enter instructions to control module 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, processor 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 module 200 and may not be accessed by any user of module 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 “processor” 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, processor 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, processor 205 may be a single core or multi-core processor with memory addressable space. In one example, processor 205 may be multi-core, comprising—for example—an 8 core CPU.

In accordance with embodiments of the invention, processor 205 may be configured to have a secure environment virtual processor (hereinafter known as “secure environment”) and a normal environment virtual processor (hereinafter known as “normal environment”).

The secure environment is typically hardware-separated from the normal environment thereby ensuring that applications, kernels, modules and operating systems located within the normal environment may not access applications, kernels, modules and operating systems provided within the secure environment. Such a hardware-based separation may be provided by, but are not limited to, processor architectures provided by Advanced RISC Machines (ARM). In ARM processors, such a hardware-based separation is implemented as Trust Zone technology. The Trust Zone technology of ARM processors enables hardware-based separation for dividing code execution on a single physical processor core into two environments, a secure environment and a normal environment. Components installed within the secure environment are subject to strict security requirements and as result, may perform limited functions and may only be accessed or modified by manufacturers of the device or super-users of the device.

Figure 3 illustrates a timing diagram for the generation of an intermediate key IK_{SR} using identity-based cryptography in accordance with embodiments of the invention. In embodiments of the invention, non-interactive keys are generated or realized using mathematical pairings such as Weil or Tate pairings. These pairings are usually associated with Elliptic curves or Abelian varieties. Such pairings typically involve two cyclic groups such as G_1 and G_2 whereby both these groups have a same large prime order λ . The first G_1 comprises a group of points provided on an Elliptic curve or Abelian variety and the group law for G_1 is written additively. Further, a generator of group G_1 may be denoted as P . As for the second group G_2 , this group is a multiplicative subgroup of a finite field and the group law on G_2 is written multiplicatively. Pairing function, $\hat{e}: G_1 \times G_1 \rightarrow G_2$, may then be used to map two elements from the first group G_1 to one element of the second group G_2 . An important feature of the pairing function is that this pair is bilinear. This means that if points Q_1 and Q_2 are in G_1 and a and b comprises two integers, then $\hat{e}(aQ_1, bQ_2) = \hat{e}(Q_1, Q_2)^{ab}$. Similarly, based on the same theory, this means that $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab} = \hat{e}(bP, aP)$.

In this embodiment of the invention, it shall be assumed that electronic device 105 intends to construct a self-authenticating message in accordance with embodiments of the invention, whereby this self-authenticating message is then sent to device 110.

Electronic device 105 will initiate the process to generate its private key, Key_{105} based on which its first and second partial keys are further generated, by transmitting its identity ID_{105} to server 120. This takes place at step 305 in Figure 3. In this embodiment of the invention, server 120 acts as a Private Key Generator (PKG) server whereby server 120 will generate secret keys for registered devices using the registered device's identity. The

identity transmitted from device 105 to server 120 may include, but is not limited to, the device's name, IP address, media access control address, or any type of unique alphanumeric or numeric identifier that may be associated with device 105. Upon receiving the identity ID_{105} of device 105, server 120 will then proceed to compute the private key for device 105 at step 310. Using a master key s provided within server 120, (whereby s is a random number selected from Z_{λ}^* with a large prime order λ whereby Z_{λ}^* refers to integers in the set $[1, \lambda]$) server 120 then computes the private key for device 105 using the following equation:

$$Key_{105} = s \cdot H(ID_{105})$$

whereby Key_{105} is the private key for device 105 and $H()$ is a hash function $H: \{0,1\}^* \rightarrow G1$.

Private key Key_{105} is then transmitted securely back to device 105 at step 315. At step 320, device 105 will then utilize the received private key Key_{105} to compute a first partial key FPK_{105} and a second partial key SPK_{105} . In embodiments of the invention, the first and second partial keys FPK_{105} and SPK_{105} may be calculated using the following equations:

$$FPK_{105} = r_1 \cdot Key_{105}$$

$$SPK_{105} = r_1$$

whereby r_1 is a random number selected from Z_{λ}^* with a large prime order λ whereby Z_{λ}^* refers to integers in the set $[1, \lambda]$.

Once electronic device 105 has obtained the first and second partial keys, electronic device 105 may then proceed to compute its intermediate key IK_{SR_105} . In an embodiment of the invention, the intermediate key IK_{SR_105} may be computed at remote server 330. In this embodiment, the first partial key FPK_{105} and the identity of a device receiving the self-authenticating message, i.e. the identity of device 110 ID_{110} , will both be transmitted to remote server 330 from device 105. At remote server 330, the intermediate key IK_{SR_105} may then be calculated using the following equation:

$$IK_{SR_105} = \hat{e}(FPK_{105}, H(ID_{110})) \quad \dots (1)$$

where FPK_{105} is the first partial key for device 105 and ID_{110} is the identity of device 110. Once server 330 has completed the calculation of the intermediate key IK_{SR_105} , server 330 will then transmit the intermediate key IK_{SR_105} back to device 105.

In another embodiment of the invention, instead of carrying out the calculation of the intermediate key IK_{SR_105} at a remote server, the calculation of the intermediate key IK_{SR_105}

may instead be carried out at the normal environment 325 of a processor in electronic device 105. In this embodiment, the first partial key FPK_{105} and the identity ID_{110} will both be provided to the normal environment 325. Device 105 will then execute equation 1 above within normal environment 325 to obtain the intermediate key IK_{SR_105} .

Once the intermediate key IK_{SR_105} has been obtained either through the use of server 330 or through the use of normal environment 325, device 105 may then proceed to calculate the shared key K_{SR} using the following equation:

$$K_{SR} = IK_{SR_105}^{1/r1} \quad \dots (2a)$$

where $r1$ is a random number selected from Z_{λ}^* with a large prime order λ and IK_{SR_105} is the intermediate key. Equation (2a) above may be expanded to provide the following equation:

$$K_{SR} = \hat{e}(H(ID_{105}), H(ID_{110}))^s \quad \dots (2b)$$

where $H(\cdot)$ is a hash function $H: \{0,1\}^* \rightarrow G1$ and s is a random number selected from Z_{λ}^* with a large prime order λ . In accordance with embodiments of the invention, the shared key K_{SR} may be generated within the secure environment of device 105 to ensure that the shared key K_{SR} is secured and may not be easily accessed by unauthorized users.

Although it is not explicitly shown in Figure 3, one skilled in the art will recognize that the intermediate secret for device 110, i.e. IK_{RS_110} , and the shared secret K_{SR} may similarly be calculated by device 110. For completeness, the process for generating these two parameters at device 110 is set out below.

To recap, in this embodiment of the invention, device 110 is the recipient of a self-authenticating message sent from device 105. At electronic device 110, this device will then initiate the process to generate its own private key Key_{110} by transmitting its identity ID_{110} to server 120. This similarly takes place at step 305 in Figure 3. Upon receiving the identity ID_{110} of device 110, server 120 will then proceed to compute the private key for device 110. Using a similar master key s provided within server 120, server 120 then computes the private key for device 110 using the following equation:

$$Key_{110} = s \cdot H(ID_{110})$$

whereby Key_{110} is the secret key for device 110 and $H(\cdot)$ is a hash function $H: \{0,1\}^* \rightarrow G1$.

Private key Key_{110} is then transmitted securely back to device 110. Device 110 will then utilize the received private key Key_{110} to compute a first partial key FPK_{110} and a

second partial key SPK_{110} . In embodiments of the invention, the first and second partial keys FPK_{110} and SPK_{110} may be calculated using the following equations:

$$FPK_{110} = r_2 \cdot Key_{110}$$

$$SPK_{110} = r_2$$

whereby r_2 is a random number selected Z_λ^* with from a large prime order λ .

Similarly, once electronic device 110 has obtained the first and second partial keys, electronic device 110 may then proceed to compute its intermediate key IK_{RS_110} . In accordance with embodiments of the invention, the intermediate key IK_{RS_110} may be computed at remote server 330 or within normal environment 325 as previously described. Regardless whether the computation is done at server 330 or normal environment 325, the intermediate key IK_{RS_110} may be calculated using the following equation:

$$IK_{RS_110} = \hat{e}(FPK_{110}, H(ID_{105})) \quad \dots (3)$$

where FPK_{110} is the first partial key for device 110 and ID_{105} is the identity of device 105.

Once the intermediate key IK_{RS_110} has been obtained either through the use of server 330 or through the use of normal environment 325, device 110 may then proceed to calculate the shared key K_{SR} using the following equation:

$$K_{SR} = IK_{SR_110}^{1/r_2} \quad \dots (4a)$$

where r_2 is a random number selected from Z_λ^* with a large prime order λ and IK_{RS_110} is the intermediate key. Equation (2a) may be expanded to provide the following equation:

$$K_{SR} = \hat{e}(H(ID_{110}), H(ID_{105}))^s \quad \dots (4b)$$

where $H(\cdot)$ is a hash function $H: \{0,1\}^* \rightarrow G_1$ and s is a random number selected from Z_λ^* with a large prime order λ . Similarly, in accordance with embodiments of the invention, the shared key K_{SR} may be generated within the secure environment of device 110 to ensure that the shared key K_{SR} is secured.

A flow diagram showing a process for constructing a self-authenticating message 430 in accordance with an embodiment of the invention is shown in Figure 4. In this embodiment of the invention, the constructed self-authenticating message 430 comprises a sender's identity, a cryptographic nonce, a ciphertext and a Message Authentication Code. For the purposes of this embodiment, it should be understood that a first device $Device_1$ is constructing the self-authenticating message 430 to be sent to a second device $Device_2$. It

should also be noted that the first partial key $FPK_{DEVICE1}$ and the second partial key $SPK_{DEVICE1}$ for the first device and the first partial key $FPK_{DEVICE2}$ and the second partial key $SPK_{DEVICE2}$ for the second device has been pre-generated respectively by these two devices using the equations discussed in the previous sections.

Process 400 begins at step 405 whereby the first device $Device_1$ will utilize an identity of the second device $ID_{DEVICE2}$ and its previously calculated first partial key $FPK_{DEVICE1}$ to generate its intermediate key $IK_{SR_DEVICE1}$. It should be noted that in accordance with embodiments of the invention, the intermediate key $IK_{SR_DEVICE1}$ may either be computed at a remote server or within a normal environment of the first device. Regardless where the intermediate key is computed, the intermediate key may be calculated by process 400 using equation (1). Based on this equation, the intermediate key is obtained at step 405 as:

$$IK_{SR_DEVICE1} = \hat{e}(FPK_{DEVICE1}, H(ID_{DEVICE2}))$$

where $H(\cdot)$ is a hash function $H: \{0,1\}^* \rightarrow G_1$.

Process 400 then proceeds to step 410 whereby process 400 will utilize the second partial key of the first device $SPK_{DEVICE1}$ and the computed intermediate key $IK_{SR_DEVICE1}$ to generate the shared key K_{SR} . The shared key K_{SR} may then be computed using equation (2a) as:

$$K_{SR} = IK_{SR_DEVICE1}^{1/r_1}$$

where $r_1 = SPK_{DEVICE1}$. In accordance with embodiments of the invention, the shared key K_{SR} may instead be generated within the secure environment of the first device.

At step 415, process 400 will then generate a cryptographic nonce in the first device. This cryptographic nonce is then utilized together with the shared key K_{SR} to generate an encryption key K . In embodiments of the invention, encryption key K may be generated as follows. The shared key K_{SR} is first truncated to an intermediate key K_{in} . A key-hash message authentication code (HMAC) is then utilized to generate the encryption key K and the equation for this may be written as:

$$K = \text{HMAC}(K_{in}, \text{nonce})$$

where nonce represents the value of the generated cryptographic nonce and $\text{HMAC}(\cdot)$ is a key-hash message authentication code function.

The plaintext message that is to be transmitted from the first device to the second device is then encrypted into ciphertext using encryption key K at step 420. The encryption process carried out at this step may be executed using any symmetric encryption algorithm such as, but is not limited to, the Advanced Encryption Standard (AES) also known as Rijndael or the Data Encryption Standard (DES). Process 400 then proceeds to step 425 whereby process 400 generates a message authentication code (MAC) using the encryption key K and the ciphertext. The MAC may be generated by a MAC function such as:

$$\text{MAC} = \text{HMAC} (K, \text{ciphertext})$$

Once this is done, process 400 will then proceed to form the self-authenticating message 430 by appending the identity of the transmitting device, i.e. $\text{ID}_{\text{DEVICE1}}$, the cryptographic nonce and the MAC to the ciphertext. Process 400 then transmits self-authenticating message 430 to the second device, Device_2 .

In accordance with an embodiment of the invention, at step 415, encryption key K may instead be generated using the following equation:

$$K = \text{HMAC} (K_{\text{in}}, \text{nonce} \parallel \text{first constant})$$

where K_{in} is the truncated shared key K_{SR} , nonce represents the value of the generated cryptographic nonce, the first constant is a value that is pre-defined while the system was set up and $\text{HMAC}()$ is a key-hash message authentication code function.

Process 400 may also generate authentication key K_A at step 415 using the following equation:

$$K_A = \text{HMAC} (K_{\text{in}}, \text{nonce} \parallel \text{second constant})$$

where K_{in} is the truncated shared key K_{SR} , nonce represents the value of the generated cryptographic nonce, the second constant is a value that is pre-defined while the system was set up and $\text{HMAC}()$ is a key-hash message authentication code function.

In this embodiment of the invention, process 400 will then proceed to encrypt the message that is to be transmitted into ciphertext using encryption key K at step 420. At step 425, process 400 will then generate a message authentication code (MAC) using the authentication key K_A and the ciphertext. The MAC may be generated by a MAC function such as:

$$\text{MAC} = \text{HMAC} (K_A, \text{ciphertext})$$

Once this is done, process 400 will then proceed to form the self-authenticating message 430 by appending the identity of the transmitting device, i.e. $ID_{DEVICE1}$, the cryptographic nonce, and the MAC together with the ciphertext. Process 400 then transmits self-authenticating message 430 to the second device, Device₂.

In yet another embodiment of the invention, after the encryption key K has been generated at step 415, process 400 will skip step 420. This means that in this embodiment of the invention, the plaintext will not be encrypted. Process 400 then proceeds to step 425 whereby process 400 generates a message authentication code (MAC) using the encryption key K, and with plaintext 416 replacing the ciphertext as previously discussed, i.e. $MAC = HMAC(K, \text{plaintext})$. Once this is done, process 400 will then proceed to form the self-authenticating message 430 by appending the identity of the transmitting device, i.e. $ID_{DEVICE1}$, the cryptographic nonce and the MAC to the plaintext. In other words, the ciphertext in message 430 is replaced with plaintext 416 instead. Process 400 then transmits self-authenticating message 430 to the second device, Device₂.

At the second device Device₂, upon receiving the self-authenticating message 430, the second device will first utilize an identity of the first device $ID_{DEVICE1}$ and its previously calculated first partial key $FPK_{DEVICE2}$ to generate its intermediate key $IK_{RS_DEVICE2}$. It should be noted that in accordance with embodiments of the invention, the intermediate key $IK_{RS_DEVICE2}$ may either be computed at a remote server or within a normal environment of the second device. Regardless where the intermediate key is computed, the intermediate key may be calculated using equation (3). Based on this equation, the intermediate key is obtained as:

$$IK_{RS_DEVICE2} = \hat{e}(FPK_{DEVICE2}, H(ID_{DEVICE1}))$$

where $H(\cdot)$ is a hash function $H: \{0,1\}^* \rightarrow G_1$.

The second device will then utilize the second partial key of the second device $SPK_{DEVICE2}$ and the computed intermediate key $IK_{RS_DEVICE2}$ to generate the shared key K_{SR} . The shared key K_{SR} may then be computed using equation (4a) as:

$$K_{SR} = IK_{RS_DEVICE2}^{1/r_2}$$

where $r_2 = SPK_{DEVICE2}$. In accordance with embodiments of the invention, the shared key K_{SR} may instead be generated within the secure environment of the first device.

The second device will then retrieve the cryptographic nonce from the received self-authenticating message 430. This cryptographic nonce is then utilized together with the

shared key K_{SR} to compute an encryption key K . In embodiments of the invention, encryption key K may be generated as follows. The shared key K_{SR} is first truncated to an intermediate key K_{in} . A key-hash message authentication code (HMAC) is then utilized to generate the encryption key K and the equation for this may be written as:

$$K = \text{HMAC}(K_{in}, \text{nonce})$$

where nonce represents the value of the generated cryptographic nonce and $\text{HMAC}()$ is a key-hash message authentication code function.

The second device then utilizes the computed encryption key K and the ciphertext retrieved from the received self-authenticating message 430 to calculate MAC' . The MAC' may be generated using the MAC function:

$$\text{MAC}' = \text{HMAC}(K, \text{ciphertext})$$

The second device will then compare the computed MAC' with the MAC retrieved from the received self-authenticating message 430. If MAC' matches with the retrieved MAC, this means that the received message has been authenticated. If MAC' does not match with the retrieved MAC, this would mean that the received message has not been authenticated and as such the received message will be discarded. Once the message has been authenticated, ciphertext retrieved from the received self-authenticating message 430 is then decrypted using encryption key K . The decryption process may be executed using any symmetric decryption algorithm such as, but is not limited to, the Advanced Encryption Standard (AES) also known as Rijndael or the Data Encryption Standard (DES).

In accordance with the other embodiment of the invention, if the encryption key K was generated at the first device using the equation:

$$K = \text{HMAC}(K_{in}, \text{nonce} \parallel \text{first constant})$$

and if the authentication key K_A was generated using the equation:

$$K_A = \text{HMAC}(K_{in}, \text{nonce} \parallel \text{second constant})$$

this means that the encryption key K and the authentication key K_A also have to be generated by the second device using the same equations above. In order to generate the encryption key K , the second device will retrieve the nonce and the first constant from a secure database or memory provided within the second device and to generate the authentication key K_A , the second device will retrieve the nonce and the second constant from the similar secure database or memory provided within the second device.

In this embodiment of the invention, once the authentication key K_A has been obtained, the authentication key K_A and the ciphertext is used to calculate MAC' using the MAC function:

$$\text{MAC}' = \text{HMAC}(K_A, \text{ciphertext})$$

The second device will then compare the computed MAC' with the MAC retrieved from the received self-authenticating message 430. If MAC' matches with the retrieved MAC, this means that the received message has been authenticated. If MAC' does not match with the retrieved MAC, this would mean that the received message has not been authenticated and as such the received message will be discarded.

Once the message has been authenticated, ciphertext retrieved from the received self-authenticating message 430 is then decrypted using encryption key K.

In accordance with yet another embodiment of the invention, if the received message 430 contains plaintext instead of ciphertext, after the encryption key K has been generated, the second device instead utilizes the computed encryption key K and the ciphertext retrieved from the received self-authenticating message 430 to calculate MAC' as $\text{HMAC}(K, \text{plaintext})$. The second device will then compare the computed MAC' with the MAC retrieved from the received self-authenticating message 430. If MAC' matches with the retrieved MAC, this means that the received message has been authenticated. If MAC' does not match with the retrieved MAC, this would mean that the received message has not been authenticated and as such the received message will be discarded. Once the message has been authenticated, the plaintext retrieved from the received self-authenticating message 430 is then further processed as required by the second device.

Another embodiment of the invention is illustrated in Figure 5. The flow diagram illustrated in Figure 5 shows a process for constructing a self-authenticating message 530 in accordance with this embodiment of the invention. In this embodiment of the invention, the constructed self-authenticating message 530 differs from the self-authenticating message 430 in that self-authenticating message 530 does not include a cryptographic nonce. For the purposes of this embodiment, it should be understood that a first device Device_1 is constructing the self-authenticating message 530 to be sent to a second device Device_2 . It should also be noted that the first partial key $\text{FPK}_{\text{DEVICE1}}$ and the second partial key $\text{SPK}_{\text{DEVICE1}}$ for the first device and the first partial key $\text{FPK}_{\text{DEVICE2}}$ and the second partial key $\text{SPK}_{\text{DEVICE2}}$ for the second device has been pre-generated respectively by these two devices using the equations discussed in the previous sections.

Process 500 begins at step 405 whereby the first device Device_1 will utilize an identity of the second device $\text{ID}_{\text{DEVICE2}}$ and its previously calculated first partial key $\text{FPK}_{\text{DEVICE1}}$ to generate its intermediate key $\text{IK}_{\text{SR_DEVICE1}}$. As before, the intermediate key may be calculated using equation (1). Based on this equation, the intermediate key is obtained at step 505 as:

$$\text{IK}_{\text{SR_DEVICE1}} = \hat{e}(\text{FPK}_{\text{DEVICE1}}, \text{H}(\text{ID}_{\text{DEVICE2}}))$$

where $\text{H}(\)$ is a hash function $\text{H}: \{0,1\}^* \rightarrow \text{G1}$.

Process 500 then proceeds to step 510 whereby process 500 will utilize the second partial key of the first device $\text{SPK}_{\text{DEVICE1}}$ and the computed intermediate key $\text{IK}_{\text{SR_DEVICE1}}$ to generate the shared key K_{SR} . The shared key K_{SR} may then be computed using equation (2a) as:

$$\text{K}_{\text{SR}} = \text{IK}_{\text{SR_DEVICE1}}^{1/r1}$$

where $r1 = \text{SPK}_{\text{DEVICE1}}$.

At step 515, process 500 will then only utilize the shared key K_{SR} to generate an encryption key K . In embodiments of the invention, encryption key K may be generated by truncating the shared key K_{SR} to the encryption key K and this may be written as:

$$\text{K} = \text{Truncate}(\text{K}_{\text{SR}})$$

The plaintext message that is to be transmitted from the first device to the second device is then encrypted into ciphertext using encryption key K at step 520. The encryption process carried out at this step may be executed using any symmetric encryption algorithm. Process 500 then proceeds to step 525 whereby process 500 generates a message authentication code (MAC) using the encryption key K and the ciphertext. The MAC may be generated by a MAC function such as:

$$\text{MAC} = \text{HMAC}(\text{K}, \text{ciphertext})$$

Once this is done, process 500 will then proceed to form the self-authenticating message 530 by appending the identity of the transmitting device, i.e. $\text{ID}_{\text{DEVICE1}}$ and the MAC to the ciphertext. Process 400 then transmits self-authenticating message 530 to the second device, Device_2 .

In accordance with an embodiment of the invention, at step 515, encryption key K may instead be generated using the following equation:

$$K_{in} = \text{HMAC} (K_{SR}, \text{first constant})$$

$$K = \text{Truncate}(K_{in});$$

where the first constant is a value that is pre-defined while the system is set up and $\text{HMAC}()$ is a key-hash message authentication code function.

Process 500 may also generate authentication key K_A at step 515 using the following equation:

$$K_{A-in} = \text{HMAC} (K_{SR}, \text{second constant})$$

$$K_A = \text{HMAC}(K_{A-in})$$

where the second constant is a value that is pre-defined while the system was set up and $\text{HMAC}()$ is a key-hash message authentication code function.

In this embodiment of the invention, process 500 will then proceed to encrypt the message that is to be transmitted into ciphertext using encryption key K at step 520. At step 525, process 500 will then generate a message authentication code (MAC) using the authentication key K_A and the ciphertext. The MAC may be generated by a MAC function such as:

$$\text{MAC} = \text{HMAC} (K_A, \text{ciphertext})$$

Once this is done, process 500 will then proceed to form the self-authenticating message 530 by appending the identity of the transmitting device, i.e. $\text{ID}_{\text{DEVICE1}}$, and the MAC to the ciphertext. Process 500 then transmits self-authenticating message 530 to the second device, Device_2 .

At the second device Device_2 , upon receiving the self-authenticating message 530, the second device will first utilize an identity of the first device $\text{ID}_{\text{DEVICE1}}$ and its previously calculated first partial key $\text{FPK}_{\text{DEVICE2}}$ to generate its intermediate key $\text{IK}_{\text{RS_DEVICE2}}$ and this can be done using equation (3). Based on this equation, the intermediate key is obtained as:

$$\text{IK}_{\text{RS_DEVICE2}} = \hat{e} (\text{FPK}_{\text{DEVICE2}}, H (\text{ID}_{\text{DEVICE1}}))$$

where $H()$ is a hash function $H: \{0,1\}^* \rightarrow G_1$.

The second device will then utilize the second partial key of the second device $\text{SPK}_{\text{DEVICE2}}$ and the computed intermediate key $\text{IK}_{\text{RS_DEVICE2}}$ to generate the shared key K_{SR} . The shared key K_{SR} may then be computed using equation (4a) as:

$$K_{SR} = IK_{RS_DEVICE2}^{1/r2}$$

where $r2 = SPK_{DEVICE2}$.

The shared key K_{SR} is then used to compute an encryption key K . In embodiments of the invention, encryption key K may be generated by truncating the shared key K_{SR} to the encryption key K and this may be written as:

$$K = \text{Truncate}(K_{SR})$$

The second device then utilizes the computed encryption key K and the ciphertext retrieved from the received self-authenticating message 530 to calculate MAC' . The MAC' may be generated using the MAC function:

$$MAC' = \text{HMAC}(K, \text{ciphertext})$$

The second device will then compare the computed MAC' with the MAC retrieved from the received self-authenticating message 530. If MAC' matches with the retrieved MAC , this means that the received message has been authenticated. If MAC' does not match with the retrieved MAC , this would mean that the received message has not been authenticated and as such the received message will be discarded. Once the message has been authenticated, ciphertext retrieved from the received self-authenticating message 530 is then decrypted using encryption key K . The decryption process may be executed using any symmetric decryption algorithm as previously discussed.

In accordance with the other embodiment of the invention, if the encryption key K was generated at the first device using the equation:

$$K_{in} = \text{HMAC}(K_{SR}, \text{first constant})$$

$$K = \text{Truncate}(K_{in})$$

and if the authentication key K_A was generated using the equation:

$$K_{A-in} = \text{HMAC}(K_{SR}, \text{second constant})$$

$$K_A = \text{Truncate}(K_{A-in})$$

this means that the encryption key K and the authentication key K_A also have to be generated by the second device using the same equations above. In order to generate the encryption key K , the second device will retrieve the first constant from a secure database or memory provided within the second device and to generate the authentication key K_A , the

second device will retrieve the second constant from the similar secure database or memory provided within the second device.

In this embodiment of the invention, once the authentication key K_A has been obtained, the authentication key K_A and the ciphertext is used to calculate MAC' using the MAC function:

$$MAC' = HMAC(K_A, ciphertext)$$

The second device will then compare the computed MAC' with the MAC retrieved from the received self-authenticating message 530. If MAC' matches with the retrieved MAC, this means that the received message has been authenticated. If MAC' does not match with the retrieved MAC, this would mean that the received message has not been authenticated and as such the received message will be discarded. Once the message has been authenticated, ciphertext retrieved from the received self-authenticating message 530 is then decrypted using encryption key K .

One skilled in the art will recognize that the ciphertext in message 530 may be replaced with plaintext 516 without departing from this invention. If plaintext 516 is utilized to replace the ciphertext in message 530, the encryption step 520 at the first device and the subsequent decryption step at the second device may then be omitted. Similarly, the MAC at the sender and receiver will then be generated using the encryption key K and the plaintext instead of the ciphertext.

Yet another embodiment of the invention is illustrated in Figure 6. The flow diagram illustrated in Figure 6 shows a process for constructing a self-authenticating message 630 in accordance with this embodiment of the invention. In this embodiment of the invention, the constructed self-authenticating message 630 differs from the self-authenticating messages 430 and 530 in that self-authenticating message 630 only comprises a cryptographic nonce and the ciphertext.

Similarly, for the purposes of this embodiment, it should be understood that a first device $Device_1$ is constructing the self-authenticating message 630 to be sent to a second device $Device_2$. It should also be noted that the first partial key $FPK_{DEVICE1}$ and the second partial key $SPK_{DEVICE1}$ for the first device and the first partial key $FPK_{DEVICE2}$ and the second partial key $SPK_{DEVICE2}$ for the second device has been pre-generated respectively by these two devices using the equations discussed in the previous sections.

Process 600 begins at step 605 whereby the first device $Device_1$ will utilize an identity of the second device $ID_{DEVICE2}$ and its previously calculated first partial key

FPK_{DEVICE1} to generate its intermediate key IK_{SR_DEVICE1} and this intermediate key may be calculated by process 600 using equation (1). It should be noted that in accordance with embodiments of the invention, the intermediate key IK_{SR_DEVICE2} may either be computed at a remote server or within a normal environment of the second device. Based on this equation, the intermediate key is obtained at step 605 as:

$$IK_{SR_DEVICE1} = \hat{e} (FPK_{DEVICE1}, H (ID_{DEVICE2}))$$

where H() is a hash function $H: \{0,1\}^* \rightarrow G1$.

Process 600 then proceeds to step 610 whereby process 600 will utilize the second partial key of the first device SPK_{DEVICE1} and the computed intermediate key IK_{SR_DEVICE1} to generate the shared key K_{SR}. The shared key K_{SR} may then be computed using equation (2a) as:

$$K_{SR} = IK_{SR_DEVICE1}^{1/r1}$$

where r1 = SPK_{DEVICE1}.

At step 615, process 600 will then generate a cryptographic nonce in the first device. The shared key K_{SR} is used to generate an encryption key K. In embodiments of the invention, encryption key K may be generated as follows. The shared key K_{SR} is first hashed using a hash function such as, but not limited to, a Secure Hash Algorithm (SHA-1), SHA-2 or SHA-3. The obtained result is then truncated using a truncation function such as a floor function. The equation to derive the encryption key K may be written as:

$$K = \text{Truncate} (H (K_{SR}))$$

where Truncate() is a truncation function and H() is a hash function.

At step 620, process 600 will then generate a cryptographic nonce in the first device. This cryptographic nonce is then utilized together with encryption key K to encrypt the plaintext message that is to be transmitted from the first device to the second device into ciphertext. The encryption process carried out at this step may be executed using any symmetric encryption algorithm such as, but is not limited to, the Advanced Encryption Standard (AES) in Counter (CTR) mode.

Once this is done, process 600 will then proceed to form the self-authenticating message 630 by appending the cryptographic nonce and to the ciphertext. Process 600 then transmits self-authenticating message 630 to the second device, Device₂.

At the second device Device₂, upon receiving the self-authenticating message 630, the second device will first utilize an identity of the first device ID_{DEVICE1} and its previously calculated first partial key FPK_{DEVICE2} to generate its intermediate key IK_{RS_DEVICE2}. It should be noted that in accordance with embodiments of the invention, the intermediate key IK_{RS_DEVICE2} may either be computed at a remote server or within a normal environment of the second device. Regardless where the intermediate key is computed, the intermediate key may be calculated using equation (3). Based on this equation, the intermediate key is obtained as:

$$IK_{RS_DEVICE2} = \hat{e}(FPK_{DEVICE2}, H(ID_{DEVICE1}))$$

where H() is a hash function $H: \{0,1\}^* \rightarrow G_1$.

The second device will then utilize the second partial key of the second device SPK_{DEVICE2} and the computed intermediate key IK_{RS_DEVICE2} to generate the shared key K_{SR}. The shared key K_{SR} may then be computed using equation (4a) as:

$$K_{SR} = IK_{RS_DEVICE2}^{1/r_2}$$

where $r_2 = SPK_{DEVICE2}$. In accordance with embodiments of the invention, the shared key K_{SR} may instead be generated within the secure environment of the first device.

The second device will then proceed to compute the encryption key K as follows. The shared key K_{SR} is first hashed using a hash function such as, but not limited to, a Secure Hash Algorithm (SHA-1), SHA-2 or SHA-3. The obtained result is then truncated using a truncation function such as a floor function. The equation to derive the encryption key K may be written as:

$$K = \text{Truncate}(H(K_{SR}))$$

where Truncate() is a truncation function and H() is a hash function.

The second device will then retrieve the cryptographic nonce from the received self-authenticating message 630. This cryptographic nonce is then utilized together with the encryption key K to decrypt ciphertext retrieved from the received self-authenticating message 630. The decryption process may be executed using any symmetric decryption algorithm such as, but is not limited to, the Advanced Encryption Standard (AES) in Counter (CTR) mode.

In accordance with an embodiment of the invention, a method for constructing a self-authenticating message to be sent from a first device to a second device comprises the following six steps:

Step 1, obtaining, by the first device, a first partial key and a second partial key for the first device;

Step 2, computing, by the first device, an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device, and using the computed intermediate key IK_{SR} and the second partial key of the first device to generate a shared key K_{SR} ;

Step 3, generating, by the first device, an encryption key K using the shared key K_{SR} and using the generated encryption key K to encrypt a message;

Step 4, forming, by the first device, the self-authenticating message using the encrypted message and transmitting the formed self-authenticated message to the second device;

In order to provide such a system or method, a process is needed for constructing a self-authenticating message to be sent from a first device to a second device. The following description and Figures 7 and 8 describe embodiments of processes that provide processes in accordance with this invention.

Figure 7 illustrates process 700 that is performed by a module installed within an electronic device to construct a self-authenticating message in accordance with embodiments of this invention. Process 700 begins at step 705 whereby the module obtains a first partial key and a second partial key using a previously received private key. In embodiments of the invention, process 700 may instead begin at step 710 if the first and second partial keys have already been pre-loaded into the module. At step 710, process 700 then proceeds to compute the intermediate key IK_{SR} using an identity of the receiver and the first partial key generated at step 705. It should be noted that in embodiments of the invention, step 710 may take place in the normal environment of the electronic device or it may take place at a remote server. If the calculation of the intermediate key takes place at a remote server, the first partial key and the identity of the receiver will have to be transmitted to the remote server first and then once the computation process has been completed, the generated intermediate key will then be transmitted back to the electronic device.

Process 700 will then, at step 715, compute the shared key using the intermediate key generated at step 710 and the second partial key generated at step 705. In

embodiments of the invention, step 715 may take place in the secure environment of the electronic device.

Once the shared key has been obtained, process 700 then utilizes the shared key to compute the encryption key K . This takes place at step 720. The encryption key K is then used to encrypt a plaintext message into ciphertext. The ciphertext is then appended with any other required values to form the self-authenticating message at step 725. Process 700 then transmits the self-authenticating message to the receiving device at step 730 and process 700 then ends.

Figure 8 illustrates process 800 that is performed by a module installed within an electronic device to receive and decrypt a self-authenticating message in accordance with embodiments of this invention. Prior to step 805, it is assumed that the module would have obtained or pre-loaded a first partial key and a second partial key using a previously received private key. At step 805, process 800 then receives the self-authenticating message transmitted from the sender. Process 800 then proceeds to compute the intermediate key IK_{SR} using an identity of the sender of the message and the first partial key generated at the receiver. It should be noted that in embodiments of the invention, step 810 may take place in the normal environment of the receiver or it may take place at a remote server. If the calculation of the intermediate key takes place at a remote server, the first partial key and the identity of the sender will have to be transmitted to the remote server first and then once the computation process has been completed, the generated intermediate key will then be transmitted back to the electronic device.

Process 800 will then, at step 815, compute the shared key using the intermediate key generated at step 810 and the second partial key previously generated at the receiver. In embodiments of the invention, step 815 may take place in the secure environment of the electronic device.

Once the shared key has been obtained, process 800 then utilizes the shared key to compute the encryption key K . This takes place at step 820. The encryption key K is then used to authenticate the received message at step 825. If the message may not be authenticated, process 800 will then discard the message. Conversely, if the message is authenticated, process 800 will then proceed to step 830 whereby process 800 utilizes the encryption key k to decrypt the ciphertext contained in the received self-authenticating message. Process 800 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 method for constructing a self-authenticating message to be sent from a first device to a second device comprising:
 - obtaining, by the first device, a first partial key and a second partial key for the first device;
 - computing an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device;
 - generating a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device ;
 - generating an encryption key K using the shared key K_{SR} ;
 - encrypting a message using the generated encryption key K ;
 - forming the self-authenticating message based at least on the encrypted message; and
 - transmitting the formed self-authenticated message to the second device.
2. The method according to claim 1 wherein the forming the self-authenticating message based at least on the encrypted message comprises:
 - calculating, by the first device, a first Message Authentication Code (MAC) using the encryption key K and the encrypted message; and
 - forming the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC.
3. The method according to claim 1 or 2 wherein the obtaining the first partial key and the second partial key for the first device comprises:
 - transmitting, by the first device, the identity of the first device ID_1 to a private key generator such that the generator generates a private key K_{first_device} based on a master secret key s and the identity of the first device ID_1 ;
 - receiving the private key K_{first_device} from the private key generator; and
 - computing the first partial key and the second partial key for the first device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{first_device} .
4. The method according to any one of claims 1 to 3 wherein the computing the intermediate key IK_{SR} by the first device using the identity of the second device ID_2 and the first partial key of the first device comprises:

transmitting, by the first device, the identity of the second device ID_2 and the first partial key of the first device to a computer server such that the computer server computes the intermediate key IK_{SR} using the identity of the second device ID_2 and the first partial key of the first device; and

receiving the computed intermediate key IK_{SR} from the computer server.

5. The method according to any one of claims 1 to 4 wherein at the step of generating, by the first device, the encryption key K using the shared key K_{SR} , this step further comprises:

using a generated cryptographic nonce together with the shared key K_{SR} to generate the encryption key K ,

wherein at the step of forming the self-authenticating message, this step further comprises adding, by the first device, the cryptographic nonce to the self-authenticating message.

6. The method for constructing a self-authenticating message to be sent from a first device to a second device comprising, comprising:

obtaining, by the second device, a first partial key and a second partial key for the second device;

receiving the self-authenticated message transmitted from the first device;

computing an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device;

generating a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device;

generating an encryption key K using the shared key K_{SR} ;

authenticating the received self-authenticating message using the encryption key K ; and

decrypting an encrypted message in the received self-authenticating message using the generated encryption key K , when the received self-authenticating message is authenticated successfully.

7. The method according to claim 6 wherein before the step of authenticating the received self-authenticating message using the encryption key K , the method further comprises:

calculating, by the second device, a second MAC using the generated encryption key K and the encrypted message from the received self-authenticating message,

wherein the step of authenticating the received self-authenticating message using the encryption key K , comprises:

determining, by the second device, whether the second MAC matches with a first MAC obtained from the received self-authenticating message or not;

authenticating the self-authenticating message successfully when the second MAC matches with the obtained first MAC.

8. The method according to claim 6 or 7 wherein the obtaining the first partial key and the second partial key for the second device comprises:

transmitting, by the second device, an identity of the second device ID_2 to a private key generator such that the private key generator generates a private key K_{second_device} based on a master secret key s and the identity of the second device ID_2 ;

receiving the private key K_{second_device} from the private key generator; and

computing the first partial key and the second partial key for the second device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{second_device} .

9. The method according to any one of claims 6 to 8 wherein the computing the intermediate key IK_{RS} by the second device using the identity of the first device ID_1 and the first partial key of the second device comprises:

transmitting, by the second device, the identity of the first device ID_1 and the first partial key of the second device to a computer server such that the computer server computes the intermediate key IK_{RS} using the identity of the first device ID_1 and the first partial key of the second device; and

receiving the computed intermediate key IK_{RS} from the computer server.

10. The method according to any one of claims 6 to 9 wherein at the step of generating, by the second device, the encryption key K using the shared key K_{SR} , the method further comprises:

retrieving, by the second device, a cryptographic nonce from the received self-authenticating message; and

generating, by the second device, the encryption key K , using the retrieved cryptographic nonce and the shared key K_{SR} .

11. A method for constructing a self-authenticating message to be sent from a first device to a second device comprising:

obtaining, by the first device, a first partial key and a second partial key for the first device;

computing an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device;

generating a shared key K_{SR} based on the computed intermediate key IK_{SR} and the second partial key of the first device ;

generating an encryption key K using the shared key K_{SR} together with a first constant and encrypting a message by the generated encryption key K , and generating an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and the second constants are obtained from the first device;

calculating a first Message Authentication Code (MAC) using the authentication key K_A and the encrypted message;

forming the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC; and

transmitting the formed self-authenticated message to the second device.

12. The method according to claim 11 wherein prior to the step of generating the encryption key K and the authentication key K_A , the method comprises the step of:

generating, by the first device, a cryptographic nonce, wherein the encryption key K is derived as: $K = \text{HMAC} (K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A is derived as: $K_A = \text{HMAC} (K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$, where $\text{HMAC} ()$ is a key-hash message authentication code function.

13. A method for constructing a self-authenticating message to be sent from a first device to a second device comprising:

obtaining, by the second device, a first partial key and a second partial key for the second device;

receiving the self-authenticating message transmitted from the first device;

computing an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device, and using the computed intermediate key IK_{RS} and the second partial key of the second device to generate a shared key K_{SR} ;

computing an encryption key K using the shared key K_{SR} and a first constant and an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and second constants are obtained from the second device,

calculating a second MAC using the computed authentication key K_A and an encrypted message from the received self-authenticating message,

determining if the second MAC matches with a first MAC obtained from the received self-authenticating message; and

decrypting the encrypted message in the received self-authenticating message using the generated encryption key K , when the second MAC matches with the obtained first MAC.

14. The method according to claim 13, wherein at the step of computing, by the second device, the encryption key K and the authentication key K_A , the encryption key K is derived as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A is derived as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$ whereby the first and second constants are obtained from the received self-authenticating message.
15. A method for constructing a self-authenticating message to be sent from a first device to a second device comprising:
 - obtaining, by the first device, a first partial key and a second partial key for the first device;
 - computing an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device;
 - generating a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device;
 - generating an encryption key K using the shared key K_{SR} ;
 - calculating a first Message Authentication Code (MAC) using the encryption key K and a message; and
 - forming the self-authenticating message based at least on the identity of the first device ID_1 , the message and the first MAC; and
 - transmitting the formed self-authenticated message to the second device.
16. A method for constructing a self-authenticating message to be sent from a first device to a second device comprising:
 - obtaining, by the second device, a first partial key and a second partial key for the second device;
 - receiving the self-authenticated message transmitted from the first device;
 - computing an intermediate key IK_{RS} using an identity of a first device ID_1 and the first partial key of the second device;
 - generating a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device;

computing an encryption key K , and calculating a second MAC using the generated encryption key K and a message from the received self-authenticating message;

determining if the second MAC matches with a first MAC obtained from the received self-authenticating message; and

authenticating the self-authenticating message when the second MAC matches with the obtained first MAC.

17. A system for constructing a self-authenticating message to be sent from a first device to a second device, the system comprising:

a processor provided within the first device; and

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

obtain a first partial key and a second partial key for the first device;

compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device;

generate a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device ;

generate an encryption key K using the shared key K_{SR} ;

encrypt a message using the generated encryption key K ;

form the self-authenticating message based at least on the encrypted message;

and

transmit the formed self-authenticated message to the second device.

18. The system according to claim 17 wherein the instructions to form the self-authenticating message based at least on the encrypted message comprises:

instructions for directing the processor to:

calculate a first Message Authentication Code (MAC) using the encryption key K and the encrypted message; and

form the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC.

19. The system according to claim 17 or 18 wherein the instructions to obtain the first partial key and the second partial key for the first device comprises:

instructions for directing the processor to:

transmit the identity of the first device ID_1 to a private key generator such that the generator generates a private key K_{first_device} based on a master secret key s and the identity of the first device ID_1 ;

receive the private key K_{first_device} from the private key generator; and

compute the first partial key and the second partial key for the first device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{first_device} .

20. The system according to any one of claims 17 to 19 wherein the instructions for directing the processor to compute the intermediate key IK_{SR} by the first device using the identity of the second device ID_2 and the first partial key of the first device comprises:

instructions for directing the processor to:

transmit the identity of the second device ID_2 and the first partial key of the first device to a computer server such that the computer server computes the intermediate key IK_{SR} using the identity of the second device ID_2 and the first partial key of the first device; and

receive the computed intermediate key IK_{SR} from the computer server.

21. The system according to any one of claims 17 to 20 wherein the instructions for directing the processor to generate the encryption key K using the shared key K_{SR} , further comprises:

instructions for directing the processor to:

use a generated cryptographic nonce together with the shared key K_{SR} to generate the encryption key K ,

wherein the instructions for directing the processor to form the self-authenticating message further comprises instructions for directing the processor to add the cryptographic nonce to the self-authenticating message.

22. The system for constructing a self-authenticating message to be sent from a first device to a second device comprising, comprising:

a processor provided within the second device; and

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

obtain a first partial key and a second partial key for the second device;

receive the self-authenticated message transmitted from the first device;

compute an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device;

generate a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device;

generate an encryption key K using the shared key K_{SR} ;

authenticate the received self-authenticating message using the encryption key K ;
and

decrypt an encrypted message in the received self-authenticating message using the generated encryption key K , when the received self-authenticating message is authenticated successfully.

23. The system according to claim 22 wherein before the instructions for directing the processor to authenticate the received self-authenticating message using the encryption key K , the system further comprises:

instructions for directing the processor to

calculate a second MAC using the generated encryption key K and the encrypted message from the received self-authenticating message,

wherein the instructions for directing the processor to authenticate the received self-authenticating message using the encryption key K , comprises:

instructions for directing the processor to:

determine whether the second MAC matches with a first MAC obtained from the received self-authenticating message or not; and

authenticate the self-authenticating message successfully when the second MAC matches with the obtained first MAC.

24. The system according to claim 22 or 23 wherein the instructions for directing the processor to obtain the first partial key and the second partial key for the second device comprises:

instructions for directing the processor to:

transmit an identity of the second device ID_2 to a private key generator such that the private key generator generates a private key K_{second_device} based on a master secret key s and the identity of the second device ID_2 ;

receive the private key K_{second_device} from the private key generator; and

compute the first partial key and the second partial key for the second device whereby the second partial key comprises a random number and the first partial key is generated based on the second partial key and the private key K_{second_device} .

25. The system according to any one of claims 22 to 24 wherein the instructions for directing the processor to compute the intermediate key IK_{RS} by the second device using the identity of the first device ID_1 and the first partial key of the second device comprises:
instructions for directing the processor to:
- transmit the identity of the first device ID_1 and the first partial key of the second device to a computer server such that the computer server computes the intermediate key IK_{RS} using the identity of the first device ID_1 and the first partial key of the second device; and
 - receive the computed intermediate key IK_{RS} from the computer server.
26. The system according to any one of claims 22 to 25 wherein the instructions for directing the processor to generate the encryption key K further comprises:
instructions for directing the processor to:
- retrieve a cryptographic nonce from the received self-authenticating message;
 - and
 - generate the encryption key K , using the retrieved cryptographic nonce and the shared key K_{SR} .
27. A system for constructing a self-authenticating message to be sent from a first device to a second device comprising:
- a processor provided within the first device; and
 - a non-transitory media readable by the processor provided within the first device, the media storing instructions that when executed by the processor, cause the processor to:
 - obtain a first partial key and a second partial key for the first device;
 - compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device;
 - generate a shared key K_{SR} based on the computed intermediate key IK_{SR} and the second partial key of the first device ;
 - generate an encryption key K using the shared key K_{SR} together with a first constant and encrypting a message by the generated encryption key K , and generating an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and the second constants are obtained from the first device;
 - calculate a first Message Authentication Code (MAC) using the authentication key K_A and the encrypted message;
 - form the self-authenticating message based at least on the identity of the first device ID_1 , the encrypted message and the first MAC; and
 - transmit the formed self-authenticated message to the second device.

28. The system according to claim 27 wherein prior to the instructions for directing the processor to generate the encryption key K and the authentication key K_A , the system comprises:

instructions for directing the processor to:

generate a cryptographic nonce, wherein the encryption key K is derived as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A is derived as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{second constant})$, where $\text{HMAC}()$ is a key-hash message authentication code function.

29. A system for constructing a self-authenticating message to be sent from a first device to a second device comprising:

a processor provided within the second device; and

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

obtain a first partial key and a second partial key for the second device;

receive the self-authenticating message transmitted from the first device;

compute an intermediate key IK_{RS} using an identity of the first device ID_1 and the first partial key of the second device, and using the computed intermediate key IK_{RS} and the second partial key of the second device to generate a shared key K_{SR} ;

compute an encryption key K using the shared key K_{SR} and a first constant and an authentication key K_A using the shared key K_{SR} and a second constant, wherein the first and second constants are obtained from the second device,

calculate a second MAC using the computed authentication key K_A and an encrypted message from the received self-authenticating message,

determine if the second MAC matches with a first MAC obtained from the received self-authenticating message; and

decrypt the encrypted message in the received self-authenticating message using the generated encryption key K , when the second MAC matches with the obtained first MAC.

30. The system according to claim 29, wherein the instructions for directing the processor to compute the encryption key K and the authentication key K_A , comprises:

instructions for directing the processor to:

derive the encryption key K as: $K = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel \text{first constant})$, and the authentication key K_A as: $K_A = \text{HMAC}(K_{SR}, \text{cryptographic nonce} \parallel$

second constant) whereby the first and second constants are obtained from the received self-authenticating message.

31. A system for constructing a self-authenticating message to be sent from a first device to a second device comprising:
 - a processor provided within the first device; and
 - a non-transitory media readable by the processor provided within the first device, the media storing instructions that when executed by the processor, cause the processor to:
 - obtain a first partial key and a second partial key for the first device;
 - compute an intermediate key IK_{SR} using an identity of the second device ID_2 and the first partial key of the first device;
 - generate a shared key K_{SR} using the computed intermediate key IK_{SR} and the second partial key of the first device;
 - generate an encryption key K using the shared key K_{SR} ;
 - calculate a first Message Authentication Code (MAC) using the encryption key K and a message; and
 - form the self-authenticating message based at least on the identity of the first device ID_1 , the message and the first MAC; and
 - transmit the formed self-authenticated message to the second device.

32. A system for constructing a self-authenticating message to be sent from a first device to a second device comprising:
 - a processor provided within the second device; and
 - a non-transitory media readable by the processor provided within the second device, the media storing instructions that when executed by the processor, cause the processor to:
 - obtain a first partial key and a second partial key for the second device;
 - receive the self-authenticated message transmitted from the first device;
 - compute an intermediate key IK_{RS} using an identity of a first device ID_1 and the first partial key of the second device;
 - generate a shared key K_{SR} using the computed intermediate key IK_{RS} and the second partial key of the second device;
 - compute an encryption key K , and calculating a second MAC using the generated encryption key K and a message from the received self-authenticating message;
 - determine if the second MAC matches with a first MAC obtained from the received self-authenticating message; and
 - authenticate the self-authenticating message when the second MAC matches with the obtained first MAC.