" Determining a common secret for the secure exchange of information and hierarchical, deterministic cryptographic keys"

Technical Field

[0001] The present disclosure relates to determining a common secret for two nodes. In some applications, the common secret may be used for cryptography to enable secure communication between two nodes. The invention may be suited for use with, but not limited to, digital wallets, blockchain (e.g. Bitcoin) technologies and personal device security.

Background

[0002] Cryptography involves techniques for secure communication between two or more nodes. A node may include a mobile communication device, a tablet computer, a laptop computer, desktop, other forms of computing devices and communication devices, a server device in a network, a client device in a network, one or more nodes in a distributed network, etc. The nodes may be associated with a natural person, a group of people such as employees of a company, a system such as a banking system, etc.

[0003] In some cases, the two or more nodes may be linked by a communications network that is unsecure. For example, the two nodes may be linked by a communications network where a third party may be able to eavesdrop on the communication between the nodes. Therefore, messages sent between nodes can be sent in encrypted form and where, upon receipt, the intended recipients may decrypt the messages with corresponding decryption key(s) (or other decryption methods). Thus the security of such communication may be dependent on preventing the third party from determining the corresponding decryption key.

[0004] One method of cryptography includes using symmetric-key algorithms. The keys are symmetric in the sense that the same symmetric-key is used for both encryption of a plain text message and decryption of cipher text. One consideration of using symmetric-key algorithms is how to transmit the symmetric-key to both nodes in a secure way to prevent an eavesdropper from acquiring the symmetric-key. This may include, for example, physically delivering the symmetric-key to the (authorised) nodes so that the symmetric-key is never transmitted over an unsecure communications network. However, physical delivery in not

always an option. Therefore a problem in such cryptographic systems is the establishment of the symmetric-key (which may be based on a common secret) between the nodes across an unsecure network. In recent times, situations may make it desirable that transmission of keys is usually done electronically over communications systems such as the internet. Thus this step of providing a shared secret (e.g. the symmetric-key) is a potentially catastrophic vulnerability. As the symmetric-key algorithms (and protocols) are simple and widely used, there is a need for an ability for two nodes to determine a common secret key securely across an unsecure network.

[0005] Other existing cryptography methods include using asymmetric-keys. These may be used in public-key cryptography where they asymmetric-keys include a private key and a corresponding public key. The public key may be made publicly available whereas the private key, as the name implies, is kept private. These asymmetric-keys may be used for public-key encryption and for digital signature amongst other things. Existing protocols include as the Diffie-Hellman Key Exchange and the Three Pass Protocol enable the secure sharing of a secret across unsecure networks. However these methods are computationally expensive in some cases, such as where new secrets are to be continuously generated and shared.

[0006] Alternative asymmetric key hierarchies (such as described in the Bitcoin Developer's Guide) rely on a random seed and an index structure resulting in poor key management. In contrast, embodiments of the present invention may comprise the use of meaningful 'messages' (M) to not only generate asymmetric keys but also deterministic hierarchical shared secrets which are provably associated with specific data.

[0007] Any discussion of documents, acts, materials, devices, articles or the like which has been included in the present specification is not to be taken as an admission that any or all of these matters form part of the prior art base or were common general knowledge in the field relevant to the present disclosure as it existed before the priority date of each claim of this application.

[0008] Throughout this specification the word "comprise", or variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated element, integer or step,

or group of elements, integers or steps, but not the exclusion of any other element, integer or step, or group of elements, integers or steps.

Summary

[0009] According to an aspect of the present invention, there is provided a computer-implemented method of determining, at a first node (C), a common secret (CS) that is common with the first node (C) and a second node (S), wherein the first node (C) is associated with a first asymmetric cryptography pair having a first node master private key (V_{1C}) and a first node master public key (P_{1C}), and the second node (S) is associated with a second asymmetric cryptography pair having a second node master private key (V_{1S}) and a second node master public key (P_{1S}), wherein the method comprises:

- determining a first node second private key (V_{2C}) based on at least the first node master private key (V_{1C}) and a deterministic key (DK);

- determining a second node second public key (P_{2S}) based on at least the second node master public key (P_{1S}) and the deterministic key (DK); and

- determining the common secret (CS) based on the first node second private key (V_{2C}) and the second node second public key (P_{2S}),

wherein the second node (S) has the same common secret (CS) based on a first node second public key (P_{2C}) and a second node second private key (V_{2S}), wherein: the first node second public key (P_{2C}) is based on at least the first node master public key (P_{1C}) and the deterministic key (DK); and the second node second private key (V_{2S}) is based on at least the second node master private key (V_{1S}) and the deterministic key (DK).

[0009] This provides the advantage of enabling the second public keys to be derived independently at each node, thereby increasing security, while also enabling a machine to automate generation of sub-keys. The advantage is also provided of having matched transaction inputs that cannot be tracked, since the relationship between the public keys cannot be determined by third parties. This therefore enables a higher level of anonymity to be achieved, thereby improving security.

[0010] The deterministic key (DK) may be based on a message (M). The method may further comprise: generating a first signed message (SM1) based on the message (M) and the first node second private key (V_{2C}); and sending, over the communications network, the first signed message (SM1) to the second node (S), wherein the first signed message (SM1) can be validated with a first node second public key (P_{2C}) to authenticate the first node (C).

[0011] The method may also comprise: receiving, over the communications network, a second signed message (SM2) from the second node (S); validating the second signed message (SM2) with the second node second public key (P2S); and authenticating the second node (S) based on the result of validating the second signed message (SM2), wherein the second signed message (SM2) was generated based on the message (M), or a second message (M2), and the second node second private key (V_{2S}).

[0012] The method may further comprise generating a message (M); and sending, over a communications network, the message (M) to the second node (S). Alternatively, the method may comprise receiving the message (M), over the communications network, from the second node (S). In yet another alternative, the method may comprise receiving the message (M), over the communications network, from another node. In yet another alternative, the method may comprise receiving the message (M) from a data store, and/or an input interface associated with the first node (C).

[0013] The first node master public key (P_{1C}) and second node master public key (P_{1S}) may be based on elliptic curve point multiplication of respective first node master private key (V_{1C}) and second node master private key (V_{1S}) and a generator (G).

[0014] The method may further comprise the steps of: receiving, over the communications network, the second node master public key (P_{1S}); and storing, at a data store associated with the first node (C), the second node master public key (P_{1S}).

[0015] The method may further comprise the steps of: generating, at a first node (C), the first node master private key (V_{1C}) and the first node master public key (P_{1C}); sending, over the communications network, the first node master public key (P_{1C}) to the second node (S) and/or other node; and storing, in a first data store associated with the first node (C), the first node master private key (V_{1C}).

[0016] The method may also comprise: sending, over the communications network, to the second node, a notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G) for the method of determining a common secret (CS). The step of generating the first node master private key (V_{1C}) and the first node master public key (P_{1C}) may comprise: generating the first node master private key (V_{1C}) based on a random integer in an allowable range specified in the common ECC system; and determining the first node master public key (P_{1C}) based on elliptic curve point multiplication of the first node master private key (V_{1C}) and the common generator (G) according to the following formula:

$$P_{1C} = V_{1C} \times G$$

[0017] The method may further comprise: determining the deterministic key (DK) based on determining a hash of the message (M), wherein the step of determining a first node second private key (V2C) is based on a scalar addition of the first node master private key (V_{1C}) and the deterministic key (DK) according to the following formula:

$$V_{2C} = V_{1C} + DK$$

[0018] The step of determining a second node second public key (P2S) may be based on the second node master public key (P_{1S}) with elliptic curve point addition to the elliptic curve point multiplication of the deterministic key (DK) and the common generator (G) according to the following formula:

$$P_{2S} = P_{1S} + DK \times G.$$

[0019] The deterministic key (DK) may be based on determining a hash of a previous deterministic key.

[0020] The first asymmetric cryptography pair and the second asymmetric cryptography pair may be based on a function of respective previous first asymmetric cryptography pair and previous second asymmetric cryptography pair.

[0021] According to another aspect of the present invention, there is provided a method of secure communication between a first node and a second node with symmetric-key algorithm, wherein the method comprises:

- determining a symmetric-key based on the common secret determined according to the method described above;

- encrypting a first communication message, with the symmetric-key, to an encrypted first communication message; and

- sending, over a communications network, the encrypted first communication message from the first node (C) to the second node (S).

[0022] The method may further comprise: receiving, over a communications network, an encrypted second communication message from the second node (S); and decrypting the encrypted second communication message, with the symmetric-key, to a second communication message.

[0023] According to a further aspect of the present invention, there is provided a method of performing an online transaction between a first node and a second node, wherein the method comprises: determining a symmetric-key based on the common secret determined according to the method according to the above described method; encrypting a first transaction message, with the symmetric-key, to an encrypted first transaction message; sending, over a communications network, the encrypted first transaction message from the first node (C) to the second node (S); receiving, over a communications network, an encrypted second transaction message from the second node (S); and decrypting the encrypted second transaction message, with the symmetric-key, to a second transaction message.

[0024] According to a further aspect of the present invention, there is provided a device for determining, at a first node (c), a common secret (CS) that is common with a second node (S), wherein the first node (C) is associated with a first asymmetric cryptography pair having a first node master private key (V_{1C}) and a first node master public key (P_{1C}), and the second node (S) is associated with a second asymmetric cryptography pair having a second node master private key (V_{1S}) and a second node master public key (P_{1S}), wherein the device comprises a first processing device to perform the method as defined above to determine the common secret.

[0025] According to a further aspect of the present invention, there is provided a device for secure communication, or performing a secure online transaction between a first node and a second node, wherein the device includes a first processing device to: perform the method of secure communication or secure online transaction described above.

[0026] The device may comprise a first data store to store one or more of the first node master private key (V_{1C}). The first data store may also store one or more of the first node master public key (P_{1C}), the second node master public key (P_{1S}), and the message (M).

[0027] The device may further comprise a communications module to send and/or receive, over a communications network, one or more of the message (M), the first node master public key (P_{1C}), the second node master public key (P_{1S}), the first signed message (SM1), the second signed message (SM2), the notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G).

[0028] According to a further aspect of the present invention, there is provided a system for determining a common secret between a first node (C) and a second node (S), wherein:

the first node (C) is associated with a first asymmetric cryptography pair having a first node master private key (V_{1C}) and a first node master public key (P_{1C}); and

- the second node (S) is associated with a second asymmetric cryptography pair having a second node master private key (V_{1S}) and a second node master public key (P_{1S}), and the system comprising:

- a first processing device, associated with the first node (C), configured to:

- determine a first node second private key (V_{2C}) based on at least the first node master private key (V_{1C}) and a deterministic key (DK;

- determine a second node second public key (P_{2S}) based on at least the second node master public key (P_{1S}) and the deterministic key (DK); and

- determine the common secret (CS) based on the first node second private key (V_{2C}) and the second node second public key (P_{2S}); and

- a second processing device, associated with the second node (S), configured to:

- determine a first node second public key (P_{2C}) based on at least the first node master public key (P_{1C}) and the deterministic key (DK); and

- determine a second node second private key (V_{2S}) based on at least the second node master private key (V_{1S}) and the deterministic key (DK); and

- determine the common secret based on the first node second public key (P_{2C}) and a second node second private key (V_{2S}),

wherein the first processing device and the second processing device determine the same common secret (CS).

[0029] In the system, the deterministic key (DK) is based on a message (M), and the first processing device is further configured to: generate a first signed message (SM1) based on the message (M) and the first node second private key (V_{2C}); and send, over the communications network, the first signed message (SM1) to the second node (S). The second processing device may be further configured to: receive the first signed message (SM1); validate the first signed message (SM1) with the first node second public key (P_{2C}); and authenticate the first node (C) based on a result of the validated first signed message (SM1).

[0030] In the system, the second processing device may be further configured to: generate a second signed message (SM2) based on the message (M), or a second message (M2), and the second node second private key (V_{2S}); send the second signed message (SM2) to the first node (C), wherein the first processing device is further configured to: receive the second signed message (SM2); validate the second signed message (SM2) with the second node second public key (P_{2S}); authenticate the second node (S) based on a result of the validated second signed message (SM2).

[0031] In the system, the first processing device may be further configured to: generate the message (M); and send the message (M), wherein the second processing device is configured to: receive the message (M). In one alternative, the message is generated by another node, wherein the first processing device is configured to: receive the message (M), and wherein the second processing device is configured to receive the message (M).

[0032] In yet another alternative, the system comprises a system data store and/or input interface, wherein the first processing device and second processing device receives the message (M), or the second message (M2) from the system data store and/or input interface.

[0033] The first processing device may receive the second node master public key (P_{1S}) from the system data store and/or input device, and the second processing device may receive the first node master public key (P_{1C}) from the system data store and/or input device.

[0034] The first node master public key (P_{1C}), second node master public key (P_{1S}) may be based on elliptic curve point multiplication of respective first node master private key (V_{1C}) and second node master private key (V_{1S}) and a generator (G).

[0035] The system may further comprise: a first data store associated with the first node (C) to store the first node master private key (V_{1C}); and a second data store associated with the second node (S) to store the second node master private key (V_{1S}).

[0036] In the system, the first processing device may be configured to: generate the first node master private key (V_{1C}) and the first node master public key (P_{1C}); send the first node master public key (P_{1C}); and store the first node master private key (V_{1C}) in the first data store, wherein the second processing device is configured to: generate the second node master private key (V_{1S}) and the second node master public key (P_{1S}); send the second node master public key (P_{1S}); and store the second node master private key (V_{1S}) in the second data store.

[0037] In the system, the first data store may receive and store the second node master public key (P_{1S}); and the second data store may receive and store the first node master public key (P_{1C}).

[0038] In the system, the first processing device may be further configured to: generate the first node master private key (V_{1C}) based on a random integer in an allowable range specified in a common elliptic curve cryptography (ECC) system; and determine the first node master public key (P_{1C}) based on elliptic curve point multiplication of the first node master private key (V_{1C}) and a common generator (G) according to the formula:

$$P_{1C} = V_{1C} \times G$$

[0039] The second processing device may be further configured to: generate the second node master private key (V_{1S}) based on a random integer in the allowable range specified in the common ECC system; and determine the second node master public key (P_{1S}) based on elliptic curve point multiplication of the second node master private key (V_{1S}) and the common generator (G) according to the formula:

$$P_{1S} = V_{1S} \text{ X } G.$$

[0040] In the system, the first processing device may be configured to: determine the deterministic key (DK) based on a hash of the message (M), and wherein: the first node second private key (V_{2C}) is based on a scalar addition of the first node master private key (V_{1C}) and the deterministic key (DK) according to the formula:

$$V_{2C} = V_{1C} + DK$$

and the second node second public key (P_{2S}) is based on the second node master public key (P_{1S}) with elliptic curve point addition to the elliptic curve point multiplication of the deterministic key (DK) and the common generator (G) according to the following formula:

$$P_{2S} = P_{1S} + DK \text{ x } G$$

The second processing device may be further configured to: determine the deterministic key (DK) based on a hash of the message (M), and wherein the second node second private key (V_{2S}) is based on a scalar addition of the second node master private key (V_{1S}) and the deterministic key (DK) according to the formula:

$$V_{2S} = V_{1C} + DK$$

and the first node second public key (P_{2C}) is based on the first node master public key (P_{1C}) with elliptic curve point addition to the elliptic curve point multiplication of the deterministic key (DK) and the common generator (G) according to the following formula:

$$P_{2C} = P_{1C} + DK \text{ x } G$$

[0041] The system may further comprise: a first communications module associated with the first processing device to send and/or receive, over a communications network, one or more of the message (M), the first node master public key (P_{1C}), the second node master public key (P_{1S}), the first signed message (SM1), the second signed message (SM2), and a notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G); and a second communications module associated with the second processing device to send and/or receive, over a communications network, one or more of the message (M), the first node master public key (P_{1C}), the second node master public key (P_{1S}), the first signed message (SM1), the second signed message (SM2), and the notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G).

[0042] In the system, the deterministic key (DK) may be based on determining a hash of a previous deterministic key.

[0043] In the system, the first asymmetric cryptography pair and the second asymmetric cryptography pair may be based on a function of respective previous first asymmetric cryptography pair and previous second asymmetric cryptography pair.

[0044] According to a further aspect of the present invention, there is provided a system for secure communication between a first node and a second node with symmetric-key algorithm, wherein the system comprises: a system described above to determine a common secret with the first processing device and the second processing device, wherein the first processing device is further configured to: determine a symmetric-key based on the common secret; encrypt a first communication message, with the symmetric-key, to an encrypted first communication message; and send the encrypted first communication message. The second processing device is further configured to: determine the same symmetric-key based on the common secret; receive the encrypted first communication message; and decrypt the encrypted first communication message, with the symmetric-key, to the first communication message.

[0045] In the system for secure communication, the second processing device may be further configured to: encrypt a second communication message, with the symmetric-key, to the encrypted second communication message; and send the encrypted second communication message. The first processing device may be further configured to: receive the encrypted

second communication message; decrypt the encrypted second communication message, with the symmetric-key, to the second communication message.

[0046] In the above described system, the first and second communication messages may be transaction messages between the first node and second node for an online transaction between the first node and the second node.

[0047] According to a further aspect of the present invention, there is provided a computer program comprising machine-readable instructions to cause a processing device to implement any one of the method described above.

Brief Description of Drawings

[0048] Examples of the present disclosure will be described with reference to:

[0049] Fig. 1 is a schematic diagram of an example system to determine a common secret for a first node and second node;

[0050] Fig. 2 is a flow chart of computer-implemented methods for determining a common secret;

[0051] Fig. 3 is a flow chart of computer-implemented methods to register the first and second nodes;

[0052] Fig. 4 is another flow chart of computer-implemented methods for determining a common secret;

[0053] Fig. 5 is a flow chart of computer-implemented methods of secure communication between the first node and second node;

[0054] Fig. 6 is a schematic diagram of an example system for electronic resource rental;

[0055] Fig. 7 is a schematic diagram of an example system that applies the methods to password replacement;

[0056] Fig. 8 is a flow chart of computer implemented methods to authenticate the first node and the second node;

[0057] Fig. 9 is an example of a tree structure of different keys for different purposes;

[0058] Fig. 10 is an example of a tree structure using the master key spawning method; and

[0059] Fig. 11 illustrates a schematic of an example processing device.

Description of Embodiments

Overview

[0060] A method, device and system to determine a common secret (CS) at a first node (C) that is the same common secret at a second node (S) will now be described. Fig. 1 illustrates a system 1 that includes a first node 3 that is in communication with, over a communications network 5, with a second node 7. The first node 3 has an associated first processing device 23 and the second node 5 has an associated second processing device 27. The first and second nodes 3, 7 may include an electronic device, such as a computer, tablet computer, mobile communication device, computer server etc. In one example, the first node 3 may be a client device and the second node 7 a server.

[0061] The first node 3 is associated with a first asymmetric cryptography pair having a first node master private key (V_{1C}) and a first node master public key (P_{1C}). The second node (7) is associated with a second asymmetric cryptography pair having a second node master private key (V_{1S}) and a second node master public key (P_{1S}). The first and second asymmetric cryptography pairs for the respective first and second nodes 3, 7 may be generated during registration. Methods of registration 100, 200 performed by the first and second nodes 3, 7 will be described in further detail below with reference to Fig. 3. The public key for each node may be shared publicly, such as over the communications network 5

[0062] To determine the common secret (CS) at both the first node 3 and second node 7, the nodes 3, 7 perform steps of respective methods 300, 400 without communicating private keys over the communications network 5.

[0063] The method 300 performed by the first node 3 includes determining 330 a first node second private key (V_{2C}) based on at least the first node master private key (V_{1C}) and a deterministic key (DK). The deterministic key may be based on a message (M) that is a shared between the first and second nodes, which may include sharing the message over the communications network 5 as described in further detail below. The method 300 also includes determining 370 a second node second public key (P_{2S}) based on at least the second node master public key (P_{1S}) and the deterministic key (DK). The method 300 includes determining 380 the common secret (CS) based on the first node second private key (V_{2C}) and the second node second public key (P_{2S}).

[0064] Importantly, the same common secret (CS) can also be determined at the second node 7 by method 400. The method 400 determining 430 a first node second public key (P_{2C}) based on the first node master public key (P_{1C}) and the deterministic key (DK). The method 400 further include determining 470 a second node second private key (V_{2S}) based on the second node master private key (V_{1S}) and the deterministic key (DK). The method 400 includes determining 480 the common secret (CS) based on the second node second private key (V_{2S}) and the first node second public key (P_{2C}).

[0065] The communications network 5, may include a local area network, a wide area network, cellular networks, radio communication network, the internet, etc. These networks, where data may be transmitted via communications medium such as electrical wire, fibre optic, or wirelessly may be susceptible to eavesdropping, such as by an eavesdropper 11. The method 300, 400 may allow the first node 3 and second node 7 to both independently determine a common secret without transmitting the common secret over the communications network 5. Thus one advantage is that the common secret (CS) may be determined securely by each node without having to transmit a private key over a potentially unsecure communications network 5. In turn, the common secret may be used as a secret key (or as the basis of a secret key) for encrypted communication between the first and second nodes 3, 7 over the communications network 5.

[0066] The methods 300, 400 may include additional steps. The method 300 may include, at the first node 3, generating a signed message (SM1) based on the message (M) and the first node second private key (V_{2C}). The method 300 further includes sending 360 the first signed message (SM1), over the communications network, to the second node 7. In turn, the second

node 7 may perform the steps of receiving 440 the first signed message (SM1). The method 400 also includes the step of validating 450 the first signed message (SM1) with the first node second public key (P_{2C}) and authenticating 460 the first node 3 based on the result of validating the first signed message (SM1). Advantageously, this allows the second node 7 to authenticate that the purported first node (where the first signed message was generated) is the first node 3. This is based on the assumption that only the first node 3 has access to the first node master private key (V_{1C}) and therefore only the first node 3 can determine the first node second private key (V_{2C}) for generating the first signed message (SM1). It is to be appreciated that similarly, a second signed message (SM2) can be generated at the second node 7 and sent to the first node 3 such that the first node 3 can authenticate the second node 7, such as in a peer-to-peer scenario.

[0067] Sharing the message (M) between the first and second nodes may be achieved in a variety of ways. In one example, the message may be generated at the first node 3 which is then sent, over the communications network 5, the second node 7. Alternatively, the message may be generated at the second node 7 and then sent, over the communications network 5, to the second node 7. In yet another example, the message may be generated at a third node 9 and the message sent to both the first and second nodes 3, 7. In yet another alternative, a user may enter the message through a user interface 15 to be received by the first and second nodes 3, 7. In yet another example, the message (M) may be retrieved from a data store 19 and sent to the first and second nodes 3, 7. In some examples, the message (M) may be public and therefore may be transmitted over an unsecure network 5.

[0068] In further examples, one or more messages (M) may be stored in a data store 13, 17, 19, where the message may be associated with a session, transaction, etc, between the first node 3 and the second node 7. Thus the messages (M) may be retrieved and used to recreate, at the respective first and second nodes 3, 7, the common secret (CS) associated with that session, or transaction. Advantageously, a record to allow recreation of the common secret (CS) may be kept without the record by itself having to be stored privately or transmitted securely. This may be advantageous if numerous transactions are performed at the first and second nodes 3, 7 and it would be impractical to store all the messages (M) at the nodes themselves.

Method of registration 100, 200

[0069] An example of a method of registration 100, 200 will be described with reference to Fig. 3, where method 100 is performed by the first node 3 and method 200 is performed by the second node 7. This includes establishing the first and second asymmetric cryptography pairs for the respective first and second nodes 3, 7.

[0070] The asymmetric cryptography pairs include associated private and public keys, such as those used in public-key encryption. In this example, the asymmetric cryptography pairs are generated using Elliptic Curve Cryptography (ECC) and properties of elliptic curve operations.

[0071] Standards for ECC may include known standards such as those described by the Standards for Efficient Cryptography Group (www.sceg.org). Elliptic curve cryptography is also described in US 5,600,725, US 5,761,305, US 5889,865, US 5,896,455, US 5,933,504, US 6,122,736, US6,141,420, US 6,618,483, US 6,704,870, US 6,785,813, US 6,078,667, US 6,792,530.

[0072] In the method 100, 200, this includes the first and second nodes settling 110, 210 to a common ECC system and using a common generator (G). In one example, the common ECC system may be based on secp256K1 which is an ECC system used by Bitcoin. The common generator (G) may be selected, randomly generated, or assigned.

[0073] Turning now to the first node 3, the method 100 includes settling 110 on the common ECC system and common generator (G). This may include receiving the common ECC system and common generator from the second node 7, or a third node 9. Alternatively, a user interface 15 may be associated with the first node 3, whereby a user may selectively provide the common ECC system and/or common generator (G). In yet another alternative one or both of the common ECC system and/or common generator (G) may be randomly selected by the first node 3. The first node 3 may send, over the communications network 5, a notice indicative of using the common ECC system with a common generator (G) to the second node 7. In turn, the second node 7 may settle 210 by sending a notice indicative of an acknowledgment to using the common ECC system and common generator (G).

[0074] The method 100 also includes the first node 3 generating 120 a first asymmetric cryptography pair that includes the first node master private key (V_{1C}) and the first node

master public key (P_{1C}). This includes generating the first node master private key (V_{1C}) based, at least in part, on a random integer in an allowable range specified in the common ECC system. This also includes determining the first node master public key (P_{1C}) based on elliptic curve point multiplication of the first node master private key (V_{1C}) and the common generator (G) according to the formula:

$$P_{1C} = V_{1C} \times G \qquad \text{(Equation 1)}$$

[0075] Thus the first asymmetric cryptography pair includes:

V_{1C} : The first node master private key that is kept secret by the first node.

P_{1C}: The first node master public key that is made publicly known.

[0076] The first node 3 may store the first node master private key (V_{1C}) and the first node master public key (P_{1C}) in a first data store 13 associated with the first node 3. For security, the first node master private key (V_{1C}) may be stored in a secure portion of the first data store 13 to ensure the key remains private.

[0077] The method 100 further includes sending 130 the first node master public key (P_{1C}), over the communications network 5, to the second node 7. The second node 7 may, on receiving 220 the first node master public key (P_{1C}), store 230 the first node master public key (P_{1C}) in a second data store 17 associated with the second node 7.

[0078] Similar to the first node 3, the method 200 of the second node 7 includes generating 240 a second asymmetric cryptography pair that includes the second node master private key (V_{1S}) and the second node master public key (P_{1S}). The second node master private key (V_{1S}) is also a random integer within the allowable range. In turn, the second node master public key (P_{1S}) is determined by the following formula:

$$P_{1S} = V_{1S} \times G \qquad \text{(Equation 2)}$$

[0079] Thus the second asymmetric cryptography pair includes:

V_{1S} : The second node master private key that is kept secret by the second node.

P_{1S}: The second node master public key that is made publicly known.

[0080] The second node 7 may store the second asymmetric cryptography pair in the second data store 17. The method 200 further includes sending 250 the second node master public key (P_{1S}) to the first node 3. In turn, the first node 3 may receive 140 and stores 150 the second node master public key (P_{1S}).

[0081] It is to be appreciated that in some alternatives, the respective public master keys may be received and stored at a third data store 19 associate with the third node 9 (such as a trusted third party). This may include a third party that acts as a public directory, such as a certification authority. Thus in some examples, the first node master public key (P_{1C}) may requested and received by the second node 7 only when determining the common secret (CS) is required (and vice versa).

[0082] The registration steps may only need to occur once as an initial setup. Afterwards, the master keys can be reused in a secure matter to generate common secret(s) that are dependent, *inter alia*, on the deterministic key (DK).

Session initiation and determining the common secret by the first node 3

[0083] An example of determining a common secret (CS) will now be described with reference to Fig. 4. The common secret (CS) may be used for a particular session, time, transaction, or other purpose between the first node 3 and the second node 7 and it may not be desirable, or secure, to use the same common secret (CS). Thus the common secret (CS) may be changed between different sessions, time, transactions, etc.

Generating a message (M) 310

[0084] In this example, the method 300 performed by the first node 3 includes generating 310 a message (M). The message (M) may be random, pseudo random, or user defined. In one example, the message (M) is based on Unix time and a nonce (and arbitrary value). For example, the message (M) may be provided as:

$$\text{Message (M)} = \text{UnixTime} + \text{nonce} \qquad \text{(Equation 3)}$$

[0085] In some examples, the message (M) is arbitrary. However it is to be appreciated that the message (M) may have selective values (such as Unix Time, etc) that may be useful in some applications.

[0086] The method 300 includes sending 315 the message (M), over the communications network 3, to the second node 7. The message (M) may be sent over an unsecure network as the message (M) does not include information on the private keys.

Determining a deterministic key 320

[0087] The method 300 further includes the step of determining 320 a deterministic key (DK) based on the message (M). In this example, this includes determining a cryptographic hash of the message. An example of a cryptographic hash algorithm includes SHA-256 to create a 256-bit deterministic key (DK). That is:

$$\text{DK} = \text{SHA-256(M)} \qquad \text{(Equation 4)}$$

[0088] It is to be appreciated that other hash algorithms may be used. This may include other has algorithms in the Secure Hash Algorithm (SHA) family. Some particular examples include instances in the SHA-3 subset, including SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256. Other hash algorithms may include those in the RACE Integrity Primitives Evaluation Message Digest (RIPEMD) family. A particular example may include RIPEMD-160. Other hash functions may include families based on Zémor-Tillich hash function and knapsack-based hash functions.

Determining a first node second private key 330

[0089] The method 300 then includes the step 330 of determining 330 the first node second private key (V_{2C}) based on the second node master private key (V_{1C}) and the deterministic key (DK). This can be based on a scalar addition of the first node master private key (V_{1C}) and the deterministic key (DK) according to the following formula:

$$V_{2C} = V_{1C} + DK \qquad \text{(Equation 5)}$$

[0090] Thus the first node second private key (V_{2C}) is not a random value but is instead deterministically derived from the first node master private key. The corresponding public key in the cryptographic pair, namely the first node second public key (P_{2C}), has the following relationship:

$$P_{2C} = V_{2C} \times G \qquad \text{(Equation 6)}$$

[0091] Substitution of V_{2C} from Equation 5 into Equation 6 provides:

$$P_{2C} = (V_{1C} + DK) \times G \qquad \text{(Equation 7)}$$

[0092] Where the ' + ' operator refers to scalar addition and the 'x' operator refers to elliptic curve point multiplication. Noting that elliptic curve cryptography algebra is distributive, Equation 7 may be expressed as:

$$P_{2C} = V_{1C} \times G + DK \times G \qquad \text{(Equation 8)}$$

[0093] Finally, Equation 1 may be substituted into Equation 7 to provide:

$P_{2C} = P_{1C} + DK \times G$ (Equation 9.1)

$P_{2C} = P_{1C} + SHA\text{-}256(M) \times G$ (Equation 9.2)

[0094] In equations 8 to 9.2, the ' + ' operator refers to elliptic curve point addition. Thus the corresponding first node second public key (P_{2C}) can be derivable given knowledge of the first node master public key (P_{1C}) and the message (M). The second node 7 may have such knowledge to independently determine the first node second public key (P_{2C}) as will be discussed in further detail below with respect to the method 400.

Generate a first signed message (SM1) based on the message and the first node second private key 350

[0095] The method 300 further includes generating 350 a first signed message (SM1) based on the message (M) and the determined first node second private key (V_{2C}). Generating a signed message includes applying a digital signature algorithm to digitally sign the message (M). In one example, this includes applying the first node second private key (V_{2C}) to the message in an Elliptic Curve Digital Signature Algorithm (ECDSA) to obtain the first signed message (SM1).

[0096] Examples of ECDSA include those based on ECC systems with secp256k1, secp256r1, secp384r1, se3cp521r1.

[0097] The first signed message (SM1) can be verified with the corresponding first node second public key (P_{2C}) at the second node 7. This verification of the first signed message (SM1) may be used by the second node 7 to authenticate the first node 3, which will be discussed in the method 400 below.

Determine a second node second public key 370'

[0098] The first node 3 may then determine 370 a second node second public key (P_{2S}). As discussed above, the second node second public key (P_{2S}) may be based at least on the second node master public key (P_{1S}) and the deterministic key (DK). In this example, since the public key is determined 370' as the private key with elliptic curve point multiplication with

the generator (G), the second node second public key (P_{2S}) can be expressed, in a fashion similar to Equation 6, as:

$$P_{2S} = V_{2S} \times G \qquad \text{(Equation 10.1)}$$

$$P_{2S} = P_{1S} + DK \times G \qquad \text{(Equation 10.2)}$$

[0099] The mathematical proof for Equation 10.2 is the same as described above for deriving Equation 9.1 for the first node second public key (P_{2C}). It is to be appreciated that the first node 3 can determine 370 the second node second public key independently of the second node 7.

Determine the common secret 380 at the first node 3

[0100] The first node 3 may then determine 380 the common secret (CS) based on the determined first node second private key (V_{2C}) and the determined second node second public key (P_{2S}). The common secret (CS) may be determined by the first node 3 by the following formula:

$$S = V_{2C} \times P_{2S} \qquad \text{(Equation 11)}$$

Method 400 performed at the second node 7

[0101] The corresponding method 400 performed at the second node 7 will now be described. It is to be appreciated that some of these steps are similar to those discussed above that were performed by the first node 3.

[0102] The method 400 includes receiving 410 the message (M), over the communications network 5, from the first node 3. This may include the message (M) sent by the first node 3 at step 315. The second node 7 then determines 420 a deterministic key (DK) based on the message (M). The step of determining 420 the deterministic key (DK) by the second node 7 is similar to the step 320 performed by the first node described above. In this example, the second node 7 performs this determining step 420 independent of the first node 3.

[0103] The next step includes determining 430 a first node second public key (P_{2C}) based on the first node master public key (P_{1C}) and the deterministic key (DK). In this example, since the public key is determined 430' as the private key with elliptic curve point multiplication with the generator (G), the first node second public key (P_{2C}) can be expressed, in a fashion similar to Equation 9, as:

$$P_{2C} = V_{2C} \times G \qquad \text{(Equation 12.1)}$$

$$P_{2C} = P_{1C} + DK \times G \qquad \text{(Equation 12.2)}$$

[0104] The mathematical proof for Equations 12.1 and 12.2 is the same as those discussed above for Equations 10.1 and 10.2.

The second node 7 authenticating the first node 3

[0105] The method 400 may include steps performed by the second node 7 to authenticate that the alleged first node 3, is the first node 3. As discussed previously, this includes receiving 440 the first signed message (SM1) from the first node 3. The second node 7 may then validate 450 the signature on the first signed message (SM1) with the first node second public key (P_{2C}) that was determined at step 430.

[0106] Verifying the digital signature may be done in accordance with an Elliptic Curve Digital Signature Algorithm (ECDSA) as discussed above. Importantly, the first signed message (SM1) that was signed with the first node second private key (V_{2C}) should only be correctly verified with the corresponding first node second public key (P_{2C}), since V_{2C} and P_{2C} form a cryptographic pair. Since these keys are deterministic on the first node master private key (V_{1C}) and the first node master public key (P_{1C}) that were generated at registration of the first node 3, verifying first signed message (SM1) can be used as a basis of authenticating that an alleged first node sending the first signed message (SM1) is the same first node 3 during registration. Thus the second node 7 may further perform the step of authenticating (460) the first node 3 based on the result of validating (450) the first signed message.

[0107] The above authentication may be suitable for scenarios where one of the two nodes are a trusted node and only one of the nodes need to be authenticated. For example, the first node 3 may be a client and the second node 7 may be a server trusted by the client. Thus the

server (second node 7) may need to authenticate the credentials of the client (first node 3) in order to allow the client access to the server system. It may not be necessary for the server to be authenticate the credentials of the server to the client. However in some scenarios, it may be desirable for both nodes to be authenticated to each other, such as in a peer-to-peer scenario that will be described in another example below.

The second node 7 determining the common secret

[0108] The method 400 may further include the second node 7 determining 470 a second node second private key (V_{2S}) based on the second node master private key (V_{1S}) and the deterministic key (DK). Similar to step 330 performed by the first node 3, the second node second private key (V_{2S}) can be based on a scalar addition of the second node master private key (V_{1S}) and the deterministic key (DK) according to the following formulas:

$$V_{2S} = V_{1S} + DK \qquad \text{(Equation 13.1)}$$

$$V_{2S} = V_{1S} + \text{SHA-256}(M) \qquad \text{(Equation 13.2)}$$

[0109] The second node 7 may then, independent of the first node 3, determine 480 the common secret (CS) based on the second node second private key (V_{2S}) and the first node second public key (P_{2C}) based on the following formula:

$$S = V_{2S} \times P_{2C} \qquad \text{(Equation 14)}$$

Proof of the common secret (CS) determined by the first node 3 and second node 7

[0110] The common secret (CS) determined by the first node 3 is the same as the common secret (CS) determined at the second node 7. Mathematical proof that Equation 11 and Equation 14 provide the same common secret (CS) will now be described.

[0111] Turning to the common secret (CS) determined by the first node 3, Equation 10.1 can be substituted into Equation 11 as follows:

$$S = V_{2C} \times P_{2S} \qquad \text{(Equation 11)}$$

$$S = V_{2C} \times (V_{2S} \times G)$$

$$S = (V_{2C} \times V_{2S}) \times G \qquad \text{(Equation 15)}$$

[0112] Turning to the common secret (CS) determined by the second node 7, Equation 12.1 can be substituted into Equation 14 as follows:

$$S = V_{2S} \times P_{2C} \qquad \text{(Equation 14)}$$

$$S = V_{2S} \times (V_{2C} \times G)$$

$$S = (V_{2S} \times V_{2C}) \times G \qquad \text{(Equation 16)}$$

[0113] Since ECC algebra is commutative, Equation 15 and Equation 16 are equivalent, since:

$$S = (V_{2C} \times V_{2S}) \times G = (V_{2S} \times V_{2C}) \times G \qquad \text{(Equation 17)}$$

The common secret (CS) and secret key

[0114] The common secret (CS) may be used as a secret key, or as the basis of a secret key in a symmetric-key algorithm for secure communication between the first node 3 and second node 7.

[0115] The common secret (CS) may be in the form of an elliptic curve point (x_S, y_S). This may be converted into a standard key format using standard publicly known operations agreed by the nodes 3, 7. For example, the x_S value may be a 256-bit integer that could be used as a key for AES_{256} encryption. It could also be converted into a 160-bit integer using RIPEMD160 for any applications requiring this length key.

[0116] The common secret (CS) may be determined as required. Importantly, the first node 3 does not need to store the common secret (CS) as this can be re-determined based on the message (M). In some examples, the message(s) (M) used may be stored in data store 13, 17, 19 (or other data store) without the same level of security as required for the master private keys. In some examples, the message (M) may be publicly available.

[0117] However depending on some applications, the common secret (CS) could be stored in the first data store (X) associated with the first node provided the common secret (CS) is kept as secure as the first node master private key (V_{1C}).

[0118] Furthermore, the disclosed system may allow determination of multiple common secrets that may correspond to multiple secure secret keys based on a single master key cryptography pair. An advantage of this may be illustrated by the following example.

[0119] In situations where there are multiple sessions, each associated with multiple respective common secrets (CS), it may be desirable to have a record associated with those multiple sessions so that the respective common secrets (CS) can be re-determined for the future. In known systems, this may have required multiple secret keys to be stored in a secure data store, which may be expensive or inconvenient to maintain. In contrast, the present system has the master private keys kept secure at the respective first and second nodes, whilst the other deterministic keys, or message (M), may be stored either securely or insecurely. Despite the deterministic keys (DK, or message (M), being stored insecurely, the multiple common secrets (CS) are kept secure since the master private keys required to determine the common secrets are still secure.

[0120] The method may also be used for generating "session keys" for temporary communication links, such as for securely transmitting login passwords.

Example applications

[0121] The methods, device, and system of the present disclosure may have a number of applications including but not limited to those described below.

Message encryption

[0122] The present disclosure may be used to facilitate secure communication, in particular sending and receiving communication messages, between the first node 3 and second node 7 over a potentially unsecure communications network 5. This may be achieved by using the common secret (CS) as the basis for a symmetric-key. This method of determining a common secret (CS) and using the symmetric-key for encryption and decryption of the communication

messages may be more computationally efficient compared to known public-key encryption methods.

[0123] Methods 500, 600 of secure communication between the first node 3 and second node 7 will now be described with reference to Fig. 5. The first node 3 determines 510 a symmetric-key based on the common secret (CS) determined in the method above. This may include converting the common secret (CS) to a standard key format. Similarly, the second node 7 can also determine 610 the symmetric-key based on the common secret (CS).

[0124] To send a first communication message securely from the first node 3, over the communications network, to the second node, the first communication message needs to be encrypted. Thus the symmetric-key is used by the first node for encrypting 520 a first communication message to form an encrypted first communication message, which is then sent 530, over the communications network 5, to the second node 7. The second node 7, in turn, receives 620 the encrypted first communication message 620, and decrypts 630 the encrypted first communication message, with the symmetric-key, to the first communication message.

[0125] Similarly, the second node 7 may encrypt 640 a second communication message, with the symmetric-key, to an encrypted second communication message, which is then sent 650 to the first node 3. The first node 3 may then receive 540 the encrypted second communication message, and decrypt 550 it to the second communication message.

Cryptocurrency wallet

[0126] In another example, the method may be used for generation and management of common secrets (CS) such as secret keys for cryptocurrency transactions. Cryptocurrency keys, such as those used in Bitcoin transactions, are normally associated with funds and assets that can be exchanged for value.

Electronic resource rental

[0127] An example of using the method and system for facilitating electronic resource rental will be described with reference to Fig. 6. This illustrates a system 701 where the first node 3 is associated with a client 703 and the second node 7 is associated with an electronic resource,

such with a supercomputer facility 707. Thus the client 504 may want to use the remotely located supercomputer facility 707 for processing large amounts of confidential data.

[0128] The supercomputer facility 707 may rent out the supercomputer CPU time on a per time and/or per CPU cycle basis. The client 703 may register with the supercomputer facility by depositing their public key, such as by sending 130, over a communications network 5, the first node master public key (P_{1C}) to the second node 7.

[0129] The supercomputer facility 707 may then provide software to the client 703 for performing background processes such as establishing secure connections using AES encryption and for facilitating the steps in the method 300 described above.

[0130] When performing the method 300, the first node 3 may send 360 a first signed message (SM1) which, in part, is based on a message (M) that includes the Unix Time concatenated with a nonce.

[0131] The second node 7, may receive 440 the first signed message (SM1). The second node 7 may further perform a step of determining if the Unix Time in the message (M) is within an allowed value for the Unix Time. For example, the allowed value for the Unix Time may be set according to Terms and Conditions settled between the client 703 and the supercomputer facility 707. For example, the Unix Time (of the message) may be required to be within a set period (e.g. 300 seconds) of when the supercomputer facility receives 440 the first signed message (SM1). If the Unix Time in the message (M) is outside the allowed time, the exchange of confidential data will not be accepted.

[0132] The above steps may ensure that the resultant session key, that is based on the determined common secret (CS) at steps 380, 480, can never be reproduced at a later time and is unique to the session being established. A protocol may then be used to establish a symmetric session key, such as an AES encryption/decryption key, for the duration of the session. The session key is used for all communications between the first node 3 and the second node 7 for the duration of the session. This allows the client to encrypt code and/or large amounts of data, send these to the supercomputer facility 707 for processing, and receive encrypted results back from the supercomputer facility 707.

Password replacement, supplement or alternative

[0133] The system and method may also be used as a password replacement, supplement, or alternative. Referring to Fig. 7 there is provided a system that includes a first node 3 associated with a user and a plurality of additional nodes 7', 7'', 7'''. The plurality of additional nodes may each be associated with respective institutions participating in the same protocol. For example, the institutions may include banks, service providers, government services, insurance companies, telecommunication providers, retailers, etc.

[0134] The user 803 may wish to communicate with these institutions, in a secure manner, to access services. In known systems, this may require the user to have multiple passwords to login for each of the respective institutions. Using the same password for login for multiple institutions is not desirable for security reasons.

[0135] In this example, the user and the multiple institutions settle on using the same protocol. This may include settling on the ECC system (such as those based on secp256k1, secp256r1, secp384r1, secp521r1) and a generator (G). The user may then register and share the first node master public key (P_{1C}) with the plurality of institutions and associated additional nodes 7', 7'', 7'''. The additional nodes 7', 7'', 7''' may each perform steps of the method similar to the second node 7 as described above.

[0136] Each time the user 803 wishes to log into one of the websites of a participating institution they do not need to use a password. Instead, the protocol replaces the need for passwords for each institution. All that is required at the first node 3 is the Institution's Public Key, which is always available, and registration of the user at the institutions (including registering the first node master public key (P_{1C}) with the institution). Since registration by the user with an institution is a normal practice for using web-based services, this is not a burden on the user 803. Once the registration has been completed, a common secret (CS) can be determined, used and re-used in place of a password. For example at the start of every session, the first node 3 may generate 310 a message (M) that is sent to the additional node 7', 7'', 7''' involved in the session. The message (M) is used to determine 320, 420 a corresponding deterministic key which is then used by both the first node 3 and additional node 7', 7'', 7''' to determine the common secret (CS) as described in the methods above. Alternatively, the message (M) may be generated or received from the additional node 7', 7'',

7''. In yet another alternative, the message (M) may be a predetermined message stored in a data store 13, 17, 19 accessible by the first node 3 and/or additional node 7', 7'', 7'''.

[0137] This technique lifts a significant security burden from the institutions. In particular, they no longer need to keep a password file (secret record of passwords or password hashes) as the common secret can be recalculated from non-secret information. Rather, the institution need only keep their own master private key secure. Furthermore, the user does not need to memorise or securely store many passwords (one for each institution) so long as they can keep their first node master private key (V_{1C}) secure.

Variations

[0138] Some variations will now be described with the following examples.

Peer-to peer authentication

[0139] In a peer-to-peer scenario, the first node 3 and the second node 7 may need to authenticate the credentials of one another. An example of this will now be described with reference to Fig. 8. In this example, the method 300, 400 steps to authenticate the first node 3 based on the validated first signed message (SM1) are similar to those discussed above.

[0140] However, the method 400 performed by the second node 7 further includes generating 462 a second signed message (SM2) based on the message (M) and the second node private key (V_{2S}). In some alternatives, the second signed message (SM2) may be based on a second message (M2) and the second node private key (V_{2S}), where the second message (M2) is shared with the first node 3. The method 400 further includes sending 464 the second signed message (SM2), over the communications network 5, to the first node 3.

[0141] At the first node 3, the method 300 includes receiving the second signed message (SM2) from the second node 7. The method includes validating 374 the signature on the second signed message (SM2) with the second node second public key (P_{2S}) that was determined at step 370. The method 300 may then include authenticating 376 the second node 7 based on the result of validating the second signed message (SM2). This results in the first and second nodes 3, 7 authenticating one another.

Hierarchy of deterministic keys

[0142] In one example, a series of successive deterministic keys may be determined, where each successive key may be determined based on the preceding deterministic key.

[0143] For example, instead of repeating steps 310 to 370 and 410 to 470 to generate successive single-purpose keys, by prior agreement between the nodes, the previously used deterministic key (DK) can be rehashed repeatedly by both parties to establish a hierarchy of deterministic keys. In effect, the deterministic key, based on the hash of a message (M), can be a next generation message (M') for the next generation of deterministic key (DK'). Doing this allows successive generations of shared secrets to be calculated without the need for further protocol-establishment transmissions, in particular transmission of multiple messages for each generation of common secrets. The next generation common secret (CS') can be computed as follows.

[0144] Firstly, both the first node 3 and the second node 7 independently determine the next generation of the deterministic key (DK'). This is similar to steps 320 and 420 but adapted with the following formulas:

$$M' = \text{SHA-256}(M) \qquad \text{(Equation 18)}$$

$$DK' = \text{SHA-256}(M') \qquad \text{(Equation 19.1)}$$

$$DK' = \text{SHA-256}(\text{SHA-256}(M)) \qquad \text{(Equation 19.2)}$$

[0145] The first node 3 may then determine the next generation of the second node second public key (P_{2S}') and the first node second private key (V_{2C}') similar to steps 370 and 330 described above, but adapted with the following formulas:

$$P_{2S}' = P_{1S} + DK' \times G \qquad \text{(Equation 20.1)}$$

$$V_{2C}' = V_{1C} + DK' \qquad \text{(Equation 20.2)}$$

[0146] The second node 7 may then determine the next generation of the first node second public key (P_{2C}') and the second node second private key (V_{2S}') similar to steps 430 and 470 described above, but adapted with the following formulas:

$$P_{2C}' = P_{1C} + DK' \times G \qquad \text{(Equation 21.1)}$$

$$V_{2S}' = V_{1S} + DK' \qquad \text{(Equation 21.2)}$$

[0147] The first node 3 and the second node 7 may then each determine the next generation common secret (CS').

[0148] In particular, the first node 3 determines the next generation common secret (CS') with the formula:

$$CS' = V_{2C}' \times P_{2S}' \qquad \text{(Equation 22)}$$

[0149] The second node 7 determines the next generation common secret (CS') with the formula:

$$CS' = V_{2S}' \times P_{2C}' \qquad \text{(Equation 23)}$$

[0150] Further generations (CS'', CS''', etc.) can be calculated in the same way to create a chain hierarchy. This technique requires that both the first node 3 and the second node 7 keep track of the original Message (M) or the originally calculated deterministic key (DK), and to which node it relates. As this is publicly known information there are no security issues regarding the retention of this information. Accordingly, this information might be kept on 'hash tables' (linking hash values to public keys) and distributed freely across the network 5 (for example using Torrent). Furthermore, if any individual common secret (CS) in the hierarchy is ever compromised, this does not affect the security of any other common secrets in the hierarchy provided the private keys V_{1C}, V_{1S} remain secure.

Tree structure of keys

[0151] As well as a chain (linear) hierarchy as described above, a hierarchy in the form of a tree structure can be created.

[0152] With a tree structure, a variety of keys for different purposes such as authentication keys, encryption keys, signing keys, payment keys, etc. may be determined whereby these keys are all linked to a single securely maintained master key. This is best illustrated in Fig. 9 that shows a tree structure 901 with a variety of different keys. Each of these can be used to create a shared secret with another party.

[0153] Tree branching can be accomplished in several ways, three of which are described below.

(i) Master key spawning

[0154] In the chain hierarchy, each new 'link' (Public/Private key pair) is created by adding a multiply rehashed Message to the original master key. For example, (showing only the private key of the first node 3 for clarity):

$$V_{2C} = V_{1C} + \text{SHA-256}(M) \qquad \text{(Equation 24)}$$

$$V_{2C}' = V_{1C} + \text{SHA-256}(\text{SHA-256}(M)) \qquad \text{(Equation 25)}$$

$$V_{2C}'' = V_{1C} + \text{SHA-256}(\text{SHA-256}(\text{SHA-256}(M))) \qquad \text{(Equation 26)}$$

... and so on.

[0155] To create a branch, any key can be used as a sub-master key. For example V_{2C}' can be used as a sub-master key (V_{3C}) by adding the hash to it as is done for the regular master key:

$$V_{3C} = V_{2C}' + \text{SHA-256}(M) \qquad \text{(Equation 27)}$$

[0156] The sub-master key (V_{3C}) may itself have a next generation key (V_{3C}'), for example:

$$V_{3C}' = V_{2C}' + \text{SHA-256}(\text{SHA-256}(M)) \qquad \text{(Equation 28)}$$

[0157] This provides a tree structure 903 using the master key spawning method as shown in Fig. 10.

(ii) Logical Association

[0158] In this method all the nodes in the tree (public/private key pairs) are generated as a chain (or in any other way) and the logical relationships between the nodes in the tree is maintained by a table in which each node in the tree is simply associated with its parent node in the tree using a pointer. Thus the pointer may be used to determine the relevant public/private key pairs for determining the common secret key (CS) for the session.

(iii) Message Multiplicity

[0159] New private/public key pairs can be generated by introducing a new message at any point in the chain or tree. The message itself may be arbitrary or may carry some meaning or function (e.g. it might be related to a 'real' bank account number, etc). It may be desirable that such new messages for forming the new private/public key pairs are securely retained.

Processing device

[0160] As noted above, the first and second nodes 3, 7 may be an electronic device, such as a computer, tablet computer, mobile communication device, computer server etc. The electronic device may include a processing device 23, 27, a data store 13, 17 and a user interface 15.

[0161] Fig. 11 illustrates an example of a processing device 23, 27. The processing device 23, 27 may be used at the first node 3, second node 7 or other nodes 9. The processing device 23, 27 includes a processor 1510, a memory 1520 and an interface device 1540 that communicate with each other via a bus 1530. The memory 1520 stores instructions and data for implementing the method 100, 200, 300, 400 described above, and the processor 1510 performs the instructions from the memory 1520 to implement the method 100, 200, 300, 400. The interface device 1540, may include a communications module that facilitates communication with the communications network 5 and, in some examples, with the user interface 15 and peripherals such as data store 13, 17, 19. It should be noted that although the processing device 1501 may be independent network elements, the processing device 501 may also be part of another network element. Further, some functions performed by the processing device 1501 may be distributed between multiple network elements. For example,

the first node 3 may have multiple processing devices 23 to perform method 100, 300 in a secure local area network associated with the first node 3.

[0162] Where this disclosure describes that a user, issuer, merchant, provider or other entity performs a particular action (including signing, issuing, determining, calculating, sending, receiving, creating etc.), this wording is used for the sake of clarity of presentation. It should be understood that these actions are performed by the computing devices operated by these entities.

[0163] Signing may comprise executing a cryptographic function. The cryptographic function has an input for a clear text and an input for a key, such as a private key. A processor may execute the function to calculate a number or string that can be used as a signature. The signature is then provided together with the clear text to provide a signed text. The signature changes completely if the message text or the key changes by a single bit. While calculating the signature requires little computational power, recreating a message that has a given signature is practically impossible. This way, the clear text can only be changed and accompanied by a valid signature if the private key is available. Further, other entities can easily verify the signature using the publicly available public key.

[0164] In most circumstances, encrypting and decrypting comprises a processor executing a cryptographic function to calculate an output string representing the encrypted message or a clear text message respectively.

[0165] Keys, tokens, metadata, transactions, offers, contracts, signatures, scripts, metadata, invitations, and the like refer to data represented as numbers, text or strings stored on data memory, such as variables in program code of type "string" or "int" or other types or text files.

[0166] An example of the peer-to-peer ledger is the bitcoin Blockchain. Transferring funds or paying fees in bitcoin currency comprises creating a transaction on the bitcoin Blockchain with the funds or fees being output from the transaction. An example of a bitcoin transaction includes an input transaction hash, a transaction amount, one or more destinations, a public key of a payee or payees and a signature created by using the input transaction as the input message and a private key of a payer to calculate the signature. The transaction can be

verified by checking that the input transaction hash exists in a copy of the bitcoin Blockchain and that the signature is correct using the public key. To ensure that the same input transaction hash has not been used elsewhere already, the transaction is broadcast to a network of computing nodes ('miners'). A miner accepts and records the transaction on the Blockchain only if the input transaction hash is not yet connected and the signatures are valid. A miner rejects the transaction if the input transaction hash is already linked to a different transaction.

[0167] Allocating cryptocurrency for a token comprises creating a transaction with the allocated cryptocurrency and the token represented in a metadata field in the transaction.

[0168] When two items are associated, this means that there is a logical connection between these items. In a database, for example, identifiers for the two items may be stored in the same records to make the two items associated with each other. In a transaction, identifiers for the two items may be included in the transaction string to make the two items associated with each other.

[0169] Using the bitcoin protocol, redeeming a script and/or unlocking a token comprises calculating a signature string of the script and/or transaction using the private key. The script may require more than one signature derived from different private keys or other conditions. The output of this transaction is then provided to a miner.

[0170] Authorising another entity may comprise calculating a signature string of a transaction using a private key and providing the signature string to the entity to allow the entity to use the signature to verify the transaction.

[0171] A user having an account with another entity may comprise the entity storing information about the user, such as email address, name and potentially public keys. For example, the entity may maintain a database, such as SQL, OrientDB, MongoDB or others. In some examples, the entity may also store one or more of the user's private keys.

[0172] The skilled person will appreciate that the present invention provides numerous technical benefits and advantages over the prior art. For example, the BIP32 protocol (e.g. ass described in the Bitcoin developer's guide) uses a random seed to generate the sub-keys. This

gives rise to a need to maintain a database of indices. In accordance with the present invention, however, a meaningful message M is used to generate the sub-keys (and therefore also the sub-shared secrets). Advantageously, this obviates the need for a database of indices, and thus provides a simpler security technique which is more efficient in terms of the computing resources needed to execute it. Additionally, it enables the association of meaningful information with the sub-keys. For example, reusable sub-keys may be used to represent specific bank accounts or client codes, etc. Alternatively, once-only sub-keys may be generated based on hashing a specific invoice or movie (or other data) file etc.

[0173] It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the above-described embodiments, without departing from the broad general scope of the present disclosure as defined by the appended claims. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

CLAIMS:

1. A computer-implemented method of determining, at a first node (C), a common secret (CS) that is common with the first node (C) and a second node (S), wherein the first node (C) is associated with a first asymmetric cryptography pair having a first node master private key (V_{1C}) and a first node master public key (P_{1C}), and the second node (S) is associated with a second asymmetric cryptography pair having a second node master private key (V_{1S}) and a second node master public key (P_{1S}), wherein the method comprises:

- determining a first node second private key (V_{2C}) based on at least the first node master private key (V_{1C}) and a deterministic key (DK);

- determining a second node second public key (P_{2S}) based on at least the second node master public key (P_{1S}) and the deterministic key (DK); and

- determining the common secret (CS) based on the first node second private key (V_{2C}) and the second node second public key (P_{2S}),

wherein the second node (S) has the same common secret (S) based on a first node second public key (P_{2C}) and a second node second private key (V_{2S}), wherein:

- the first node second public key (P_{2C}) is based on at least the first node master public key (P_{1C}) and the deterministic key (DK); and

- the second node second private key (V_{2S}) is based on at least the second node master private key (V_{1S}) and the deterministic key (DK).

2. A method according to claim 1 wherein the deterministic key (DK) is based on a message (M).

3. The method according to claim 2 further comprising:

- generating a first signed message (SM1) based on the message (M) and the first node second private key (V_{2C}); and

- sending, over the communications network, the first signed message (SM1) to the second node (S),

wherein the first signed message (SM1) can be validated with a first node second public key (P_{2C}) to authenticate the first node (C).

4. A method according to either claim 2 or 3 further comprising:

- receiving, over the communications network, a second signed message (SM2) from the second node (S);

- validating the second signed message (SM2) with the second node second public key (P2S); and

- authenticating the second node (S) based on the result of validating the second signed message (SM2),

wherein the second signed message (SM2) was generated based on the message (M), or a second message (M2), and the second node second private key (V_{2S}).

5. A method according to any one of claims 2 to 4 further comprising:

- generating a message (M); and

- sending, over a communications network, the message (M) to the second node (S).

6. A method according to any one of claims 2 to 4 further comprising:

- receiving the message (M), over the communications network, from the second node (S).

7. A method according to any one of claims 2 to 4 further comprising:

- receiving the message (M), over the communications network, from another node.

8. A method according to any one of claims 2 to 4 further comprising:

- receiving the message (M) from a data store, and/or an input interface associated with the first node (C).

9. A method according to any one of the preceding claims wherein the first node master public key (P_{1C}) and second node master public key (P_{1S}) are based on elliptic curve point multiplication of respective first node master private key (V_{1C}) and second node master private key (V_{1S}) and a generator (G).

10. A method according to any one of the preceding claims further comprising the steps of:

- receiving, over the communications network, the second node master public key (P_{1S}); and

- storing, at a data store associated with the first node (C), the second node master public key (P_{1S}).

11. A method according to any preceding claim further comprising the steps of:

- generating, at a first node (C), the first node master private key (V_{1C}) and the first node master public key (P_{1C});

- sending, over the communications network, the first node master public key (P_{1C}) to the second node (S) and/or other node; and

- storing, in a first data store associated with the first node (C), the first node master private key (V_{1C}).

12. A method according to claim 11 further comprising:

- sending, over the communications network, to the second node, a notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G) for the method of determining a common secret (CS), and

wherein the step of generating the first node master private key (V_{1C}) and the first node master public key (P_{1C}) comprises:

- generating the first node master private key (V_{1C}) based on a random integer in an allowable range specified in the common ECC system; and

- determining the first node master public key (P_{1C}) based on elliptic curve point multiplication of the first node master private key (V_{1C}) and the common generator (G) according to the following formula:

$$P_{1C} = V_{1C} \times G.$$

13. A method according to claim 12 further comprising:

- determining the deterministic key (DK) based on determining a hash of the message (M), and

wherein the step of determining a first node second private key (V2C) is based on a scalar addition of the first node master private key (V_{1C}) and the deterministic key (DK) according to the following formula:

$$V_{2C} = V_{1C} + DK \text{ , and}$$

wherein the step of determining a second node second public key (P2S) is based on the second node master public key (P_{1S}) with elliptic curve point addition to the elliptic curve point multiplication of the deterministic key (DK) and the common generator (G) according to the following formula:

$$P_{2S} = P_{1S} + DK \times G.$$

14. A method according to any one of the preceding claims wherein the deterministic key (DK) is based on determining a hash of a previous deterministic key.

15. A method according to any one of the preceding claims wherein the first asymmetric cryptography pair and the second asymmetric cryptography pair are based on a function of

respective previous first asymmetric cryptography pair and previous second asymmetric cryptography pair.

16. A method of secure communication between a first node and a second node with symmetric-key algorithm, wherein the method comprises:

- determining a symmetric-key based on the common secret determined according to the method according to any one of the preceding claims;

- encrypting a first communication message, with the symmetric-key, to an encrypted first communication message; and

- sending, over a communications network, the encrypted first communication message from the first node (C) to the second node (S).

17. The method according to claim 16, wherein the method further comprises:

- receiving, over a communications network, an encrypted second communication message from the second node (S); and

- decrypting the encrypted second communication message, with the symmetric-key, to a second communication message.

18. A method of performing an online transaction between a first node and a second node, wherein the method comprises:

- determining a symmetric-key based on the common secret determined according to the method according to any one of claims 1 to 15;

- encrypting a first transaction message, with the symmetric-key, to an encrypted first transaction message;

- sending, over a communications network, the encrypted first transaction message from the first node (C) to the second node (S);

- receiving, over a communications network, an encrypted second transaction message from the second node (S); and

- decrypting the encrypted second transaction message, with the symmetric-key, to a second transaction message.

19. A device for determining, at a first node (c), a common secret (CS) that is common with a second node (S), wherein the first node (C) is associated with a first asymmetric cryptography pair having a first node master private key (V_{1C}) and a first node master public key (P_{1C}), and the second node (S) is associated with a second asymmetric cryptography pair having a second node master private key (V_{1S}) and a second node master public key (P_{1S}), wherein the device comprises a first processing device to perform the method according to any one of the preceding claims to determine the common secret.

20. A device for secure communication, or performing a secure online transaction between a first node and a second node, wherein the device includes a first processing device to:

- perform the method according to any one of claims 16 to 18.

21. A device according to either claim 19 or 20 further comprising a first data store to store one or more of the first node master private key (V_{1C}).

22. A device according to claim 21 wherein the first data store further stores one or more of the first node master public key (P_{1C}), the second node master public key (P_{1S}), and the message (M).

23. A device according to any one of claims 19 to 22, further comprising a communications module to send and/or receive, over a communications network, one or more of the message (M), the first node master public key (P_{1C}), the second node master public key (P_{1S}), the first signed message (SM1), the second signed message (SM2), the notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G).

24. A system for determining a common secret between a first node (C) and a second node (S), wherein:

- the first node (C) is associated with a first asymmetric cryptography pair having a first node master private key (V_{1C}) and a first node master public key (P_{1C}); and

- the second node (S) is associated with a second asymmetric cryptography pair having a second node master private key (V_{1S}) and a second node master public key (P_{1S}), and the system comprising:

- a first processing device, associated with the first node (C), configured to:

- determine a first node second private key (V_{2C}) based on at least the first node master private key (V_{1C}) and a deterministic key (DK;

- determine a second node second public key (P_{2S}) based on at least the second node master public key (P_{1S}) and the deterministic key (DK); and

- determine the common secret (CS) based on the first node second private key (V_{2C}) and the second node second public key (P_{2S}); and

- a second processing device, associated with the second node (S), configured to:

- determine a first node second public key (P_{2C}) based on at least the first node master public key (P_{1C}) and the deterministic key (DK); and

- determine a second node second private key (V_{2S}) based on at least the second node master private key (V_{1S}) and the deterministic key (DK); and

- determine the common secret based on the first node second public key (P_{2C}) and a second node second private key (V_{2S}),

wherein the first processing device and the second processing device determine the same common secret.

25. The system according to claim 24 wherein the deterministic key (DK) is based on a message (M), and the first processing device is further configured to:

- generate a first signed message (SM1) based on the message (M) and the first node second private key (V_{2C}); and

- send, over the communications network, the first signed message (SM1) to the second node (S),

wherein the second processing device is further configured to:

- receive the first signed message (SM1);

- validate the first signed message (SM1) with the first node second public key (P_{2C}); and

- authenticate the first node (C) based on a result of the validated first signed message (SM1).

26. The system according to any one of claims 25 wherein the second processing device is further configured to:

- generate a second signed message (SM2) based on the message (M), or a second message (M2), and the second node second private key (V_{2S}); and

- send the second signed message (SM2) to the first node (C),

wherein the first processing device is further configured to:

- receive the second signed message (SM2);

- validate the second signed message (SM2) with the second node second public key (P_{2S}); and

- authenticate the second node (S) based on a result of the validated second signed message (SM2).

27. The system according to either claim 25 or 26 wherein the first processing device is configured to:

- generate the message (M); and

- send the message (M),

wherein the second processing device is configured to:

- receive the message (M).

28. The system according to either claim 25 or 26 wherein the message is generated by another node, wherein the first processing device is configured to:

- receive the message (M),

wherein the second processing device is configured to:

- receive the message (M).

29. The system according any one of claims 25 to 28 further comprising a system data store and/or input interface, wherein the first processing device and second processing device receives the message (M), or the second message (M2) from the system data store and/or input interface.

30. The system according to claim 29, wherein the first processing device receives the second node master public key (P_{1S}) from the system data store and/or input device, and the second processing device receives the first node master public key (P_{1C}) from the system data store and/or input device.

31. The system according to any one of claims 24 to 30 wherein the first node master public key (P_{1C}), second node master public key (P_{1S}) are based on elliptic curve point multiplication of respective first node master private key (V_{1C}) and second node master private key (V_{1S}) and a generator (G).

32. The system according to any one of claims 24 to 31 further comprising:

- a first data store associated with the first node (C) to store the first node master private key (V_{1C}); and

- a second data store associated with the second node (S) to store the second node master private key (V_{1S}).

33. The system according to claim 32, wherein the first processing device is configured to:

- generate the first node master private key (V_{1C}) and the first node master public key (P_{1C});

- send the first node master public key (P_{1C}); and

- store the first node master private key (V_{1C}) in the first data store,

wherein the second processing device is configured to:

- generate the second node master private key (V_{1S}) and the second node master public key (P_{1S});

- send the second node master public key (P_{1S}); and

- store the second node master private key (V_{1S}) in the second data store.

34. The system according to either claim 32 or 33, wherein:

- the first data store receives and stores the second node master public key (P_{1S}); and

- the second data store receives and stores the first node master public key (P_{1C}).

35. The system according to any one claims 24 to 34 wherein the first processing device is further configured to:

- generate the first node master private key (V_{1C}) based on a random integer in an allowable range specified in a common elliptic curve cryptography (ECC) system; and

- determine the first node master public key (P_{1C}) based on elliptic curve point multiplication of the first node master private key (V_{1C}) and a common generator (G) according to the formula:

$P_{1C} = V_{1C} \times G$

and wherein the second processing device is further configured to:

- generate the second node master private key (V_{1S}) based on a random integer in the allowable range specified in the common ECC system; and

- determine the second node master public key (P_{1S}) based on elliptic curve point multiplication of the second node master private key (V_{1S}) and the common generator (G) according to the formula:

$P_{1S} = V_{1S} \times G$.

36. The system according to any one of claims 25 to 35 wherein the first processing device is configured to:

- determine the deterministic key (DK) based on a hash of the message (M), and wherein:

- the first node second private key (V_{2C}) is based on a scalar addition of the first node master private key (V_{1C}) and the deterministic key (DK) according to the formula:

$V_{2C} = V_{1C} + DK$; and

- the second node public key (P_{2S}) is based on the second node master public key (P_{1S}) with elliptic curve point addition to the elliptic curve point multiplication of the deterministic key (DK) and the common generator (G) according to the following formula:

$P_{2S} = P_{1S} + DK \times G$

and wherein the second processing device is configured to:

- determine the deterministic key (DK) based on a hash of the message (M), and wherein:

- the second node second private key (V_{2S}) is based on a scalar addition of the second node master private key (V_{1S}) and the deterministic key (DK) according to the formula:

$V_{2S} = V_{1C} + DK$; and

- the first node public key (P_{2C}) is based on the first node master public key (P_{1C}) with elliptic curve point addition to the elliptic curve point multiplication of the deterministic key (DK) and the common generator (G) according to the following formula:

$P_{2C} = P_{1C} + DK \times G$.

37. A system according to any one of claims 24 to 36 further comprising:

- a first communications module associated with the first processing device to send and/or receive, over a communications network, one or more of the message (M), the first node master public key (P_{1C}), the second node master public key (P_{1S}), the first signed message (SM1), the second signed message (SM2), and a notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G); and

- a second communications module associated with the second processing device to send and/or receive, over a communications network, one or more of the message (M), the first node master public key (P_{1C}), the second node master public key (P_{1S}), the first signed message (SM1), the second signed message (SM2), and the notice indicative of using a common elliptic curve cryptography (ECC) system with a common generator (G).

38. A system according to any one of claims 24 to 37, wherein the deterministic key (DK) is based on determining a hash of a previous deterministic key.

39. A method according to any one of claims 24 to 38 wherein the first asymmetric cryptography pair and the second asymmetric cryptography pair are based on a function of respective previous first asymmetric cryptography pair and previous second asymmetric cryptography pair.

40. A system for secure communication between a first node and a second node with symmetric-key algorithm, wherein the system comprises:

- a system according to any one of claims 24 to 39 to determine a common secret with the first processing device and the second processing device, wherein the first processing device is further configured to:

- determine a symmetric-key based on the common secret;

- encrypt a first communication message, with the symmetric-key, to an encrypted first communication message; and

- send the encrypted first communication message;

wherein the second processing device is further configured to:

- determine the same symmetric-key based on the common secret;

- receive the encrypted first communication message; and

- decrypt the encrypted first communication message, with the symmetric-key, to the first communication message.

41. A system for secure communication between a first node and a second node with symmetric-key algorithm, wherein the second processing device is further configured to:

- encrypt a second communication message, with the symmetric-key, to the encrypted second communication message; and

- send the encrypted second communication message;

wherein the first processing device is further configured to:

- receive the encrypted second communication message;

- decrypt the encrypted second communication message, with the symmetric-key, to the second communication message.

42. A system according to either claim 40 or 41 wherein the first and second communication messages are transaction messages between the first node and second node for an online transaction between the first node and the second node.

43. Computer program comprising machine-readable instructions to cause a processing device to implement the method according to any one of claim 1 to 18.