

METHOD AND SYSTEM FOR PERFORMING COLLECTIVE AUTHENTICATION IN A COMMUNICATION NETWORK

Field of Invention

5 The invention generally relates to a method and system for performing collective authentication in a communication network that allows a verifying node (verifying device) to collectively authenticate a plurality of proving nodes (proving devices) in a communication network.

Background

10 The existing Fiat-Shamir method allows one to one interaction between one verifying node and one proving node to authenticate the proving node in a communication network. However, in many situations, a plurality of proving nodes in a communication network need to be authenticated as a whole. A typical example is a network of electricity meters for which overall integrity must
15 be checked by a verifying node. Using the existing Fiat-Shamir protocol, such an interaction will require as many one-to-one sessions as there are electricity meters.

A second drawback of the existing Fiat-Shamir protocol is the fact that each proving node must send its information directly to the verifying node. It is
20 known that the energy required to transmit a message increases with the distance between the sender and the receiver. Internet of Things (IoT) nodes is simple, energy-restricted devices. As such, transmitting energy to the verifying

node may be costly and reduce the lifetime of the devices. It is therefore desirable to find a solution for authenticating proving nodes in a communication network to reduce energy and processing time of proving nodes.

Summary of Invention

5 The embodiments of the invention provide a solution for collectively authenticating a plurality of proving node in a communication network by a verifying node. In this solution, each proving node only transmits information to its closest neighbors, i.e. its parent node/child node(s), to be verified by the verifying node, therefore the transmitting energy to the verifying node is
10 significantly decreased. Interchangeably, collective authentication can be called as swarm authentication.

 According to a first aspect of the invention, a method for for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto is provided. The method
15 comprises:

 calculating, by each proving node in the network, a first value for the each proving node based on a predetermined public system parameter and a random value selected by the each proving node;

 calculating, by the verifying node, a first verification value based on a
20 plurality of first values calculated by the plurality of proving nodes in the network;

 generating a random number for the each proving node in the network;

calculating, by the each proving node in the network, a second value for the each proving node based on the random number generated for the each proving node, the random value selected by the each proving node, the private key of the each proving node, and the predetermined public system parameter;

5 calculating, by the verifying node, a second verification value based on a plurality of second values calculated by the plurality of proving nodes in the network; and

determining, by the verifying node, whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the
10 second verification value and the random numbers generated for the plurality of proving nodes in the network satisfy a predetermined authentication condition.

With reference to the first aspect, in a first possible implementation manner of the first aspect, the method further comprising:

sending, by each proving node in the network, a first unit value to a parent
15 node of the each proving node, wherein if one of the proving nodes in the network is directly connected to at least one child node, the first unit value is calculated based on the first values calculated by both the proving node and the at least one child node of the proving node; and if one of the proving nodes in the network is a leave node, the first unit value is determined based on the first
20 value calculated by the proving node;

wherein the step of calculating a first verification value based on the plurality of first values calculated by the proving nodes, further comprises:

calculating, by the verifying node, the first verification value based on at least one first unit value, wherein each of the at least one first unit value is respectively calculated by at least one proving node directly connected to the verifying node in the network.

5 With reference to the first aspect or the first possible implementation manner of the first aspect, in a second possible implementation manner of the first aspect, the method further comprising:

 sending, by the each proving node in the network, a second unit value to a parent node of the each proving node, wherein if one of the proving nodes is
10 directly connected to at least one child node, the second unit value is calculated based on the second values calculated by both the proving node and the at least one child node of the proving node; and if one of the proving nodes is a leave node, the second unit value is determined based on the second value calculated by the proving node;

15 wherein the step of calculating a second verification value based on a plurality of second values calculated by the proving nodes, further comprises:

 calculating, by the verifying node, the second verification value based on at least one second unit value, wherein each of the at least one second unit value is respectively calculated by at least one proving node directly connected
20 to the verifying node in the network.

 With reference to the first aspect, or any one of the first and second possible implementation manner of the first aspect, in a third possible

implementation manner of the first aspect, the step of generating a random number for the each proving node in the network, further comprises:

generating, by the verifying node, a random number for the each proving node in the network; and

- 5 sending the generated random number for the each proving node to the each proving node directly or through a parent node of the each proving node in the network.

With reference to the first aspect, or any one of the first and second possible implementation manner of the first aspect, in a fourth possible
10 implementation manner of the first aspect, the step of generating a random number for the each proving node in the network, further comprises:

generating, by the verifying node, a root random value and sharing the generated root random value in the network;

- generating, by the each proving node, a random number for the each
15 proving node based on the root random value and a pre-determined function.

With reference to the first aspect, or any one of the first to fourth possible implementation manner of the first aspect, in a fifth possible implementation manner of the first aspect, the method further comprises:

- calculating, by the each proving node, or the verifying node or a trusted
20 party, a public key or a private key of the each proving node in the network.

With reference to the fifth possible implementation manner of the first aspect, in a sixth possible implementation manner of the first aspect, the public key or the private key of the each proving node is calculated based on the predetermined public system parameter and an identity of the each proving node.

With reference to the sixth possible implementation manner of the first aspect, in a seventh possible implementation manner of the first aspect, the predetermined system parameter is a public modulus which is a product of a plurality of prime numbers, wherein at least two of the prime numbers are different.

With reference to the sixth or seventh possible implementation manner of the first aspect, in an eighth possible implementation manner of the first aspect, the public key or private key for the each proving node is calculated according to the following Equation:

$$v_{i,j} = \frac{1}{s_{i,j}^c} \bmod n, \text{ or } v_{i,j} = s_{i,j}^c \bmod n; \quad i = 1, \dots, u; j = 1, \dots, k$$

wherein $v_i = (v_{i,1}, \dots, v_{i,k})$ is the public key of the proving node P_i , $s_i = (s_{i,1}, \dots, s_{i,k})$ is the private key of the proving node P_i , n is the predetermined public system parameter called public modulus, u is the number of the proving nodes in the network, k is a pre-set integer no less than 1, and c is a pre-set integer no less than 2.

With reference to the eighth possible implementation manner of the first aspect, in a ninth possible implementation manner of the first aspect, the first

value is calculated by the each proving node in the network according to the following equation:

$$x_i = r_i^c \text{ mod } n$$

wherein x_i is the first value calculated by the proving node P_i , n is the predetermined public modulus value, r_i is the random value selected by the proving node P_i , c is the pre-set integer no less than 2.

With reference to the ninth possible implementation manner of the first aspect, in a tenth possible implementation manner of the first aspect, the first verification value is calculated according to the following equation:

$$x = \prod_{i=1}^u x_i \text{ mod } n$$

wherein x is the first verification value, x_i is the first value calculated by the proving node P_i , n is the predetermined system parameter called public modulus, u is the number of the proving nodes in the network.

With reference to the tenth possible implementation manner of the first aspect, in an eleventh possible implementation manner of the first aspect, the second verification value is calculated according to the following equation:

$$y = \prod_{i=1}^u y_i \text{ mod } n;$$

wherein y is the second verification value, y_i is the second value calculated by the proving node P_i , n is the predetermined public modulus value, u is the number of the proving nodes in the network, wherein the second value y_i is calculated according to the following equation:

$$y_i = r_i \prod_{j=1}^k s_{i,j}^{e_{i,j}} \bmod n$$

wherein n is the predetermined public modulus value, r_i is the random value selected by the proving node P_i , $s_i = (s_{i,1}, \dots, s_{i,k})$ is the private key of the proving node P_i , k is the pre-set integer $k \geq 1$, and $e_i = (e_{i,1}, \dots, e_{i,k})$ is the random number generated for the proving node P_i , wherein $e_{i,j}$ is the j -th bit of e_i .

5 With reference to the eleventh possible implementation manner of the first aspect, in a twelfth possible implementation manner of the first aspect, the step of determining whether authentication of the plurality of proving nodes is successful, further comprises:

10 determining the authentication of the plurality of proving nodes is successful if the following equation is satisfied:

$$x = y^c \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = \frac{1}{s_{i,j}^c} \bmod n$$

$$y^c = x \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = s_{i,j}^c \bmod n$$

wherein x is the first verification value, y is the second verification value, c is the pre-set integer $c \geq 2$, $v_i = (v_{i,1}, \dots, v_{i,k})$ is the public key of the proving node P_i , and $e_i = (e_{i,1}, \dots, e_{i,k})$ is the random number generated for the proving node P_i .

15 With reference to any one of the fifth to the twelfth possible implementation manner of the first aspect, in a thirteenth possible implementation manner of the first aspect, the private key of each proving node is derived from a random seed by evaluating a cryptographically strong

pseudorandom function with respect to at least the random seed and other relevant information.

With reference to any one of the fifth to the twelfth possible implementation manner of the first aspect, in a fourteenth possible
5 implementation manner of the first aspect, the private key of each proving node is derived by the proving node itself or by a trusted party which has been informed the predetermined public modulus value.

With reference to any one of the fifth to the twelfth possible implementation manner of the first aspect, in a fifteenth possible implementation
10 manner of the first aspect, the private key of each proving node is generated by randomly selecting a first group of words from a predetermined second group of words, and calculating a product of the words in the first group based on a preset table which consists of all pairwise products of the second group of words.

With reference to the fifth possible implementation manner of the first
15 aspect, in a sixteenth possible implementation manner of the first aspect, the public or private key for the each proving node is calculated according to the following Equation:

$$v_i = g^{s_i}$$

wherein v_i is the public key of the proving node P_i , s_i is the private key of the proving node P_i , and g is the generator of a cyclic group G_q , which has a prime
20 order q .

With reference to the sixteenth possible implementation manner of the first aspect, in a seventeenth possible implementation manner of the first aspect, the first value is calculated by each proving node according to the following equation:

$$x_i = g^{r_i}$$

- 5 wherein x_i is the first value calculated by the proving node P_i , g is the generator of a cyclic group G_q , r_i is the random value selected by the proving node P_i .

With reference to the seventeenth possible implementation manner of the first aspect, in an eighteenth possible implementation manner of the first aspect, the first verification value is calculated by the verifying node according to the following equation:

10

$$x = \prod_{i=1}^u x_i = \prod_{i=1}^u g^{r_i}$$

wherein x is the first verification value, u is the number of the proving nodes in the network.

With reference to the eighteenth possible implementation manner of the first aspect, in a nineteenth possible implementation manner of the first aspect, the second verification value is calculated according to the following equation:

15

$$y = \sum_{i=1}^u y_i \bmod q$$

wherein y is the second verification value, y_i is the second value calculated by the proving node P_i , u is the number of the proving nodes in the

network, q is the prime order of the cyclic group G_q wherein the second value y_i is calculated according to the following equation:

$$y_i = r_i + s_i e_i \bmod q, \text{ or}$$

$$y_i = r_i - s_i e_i \bmod q$$

wherein r_i is the random value selected by the proving node P_i , s_i is the
 5 private key of the proving node P_i , and e_i is the random number generated for the
 proving node P_i .

With reference to the nineteenth possible implementation manner of the
 first aspect, in a twentieth possible implementation manner of the first aspect, the
 step of determining whether authentication of the plurality of proving nodes is
 10 successful, further comprises:

if the first verification value, the second verification value and the random
 numbers generated for the plurality of proving nodes in the network satisfy a
 predetermined authentication condition shown in the following equation, the
 authentication of the plurality of proving nodes is successful:

$$15 \quad g^y = x \prod_{i=1}^u v_i^{e_i}, \text{ if } y_i \text{ is computed by } y_i = r_i + s_i e_i \bmod q;$$

$$x = g^y \prod_{i=1}^u v_i^{e_i}, \text{ if } y_i \text{ is computed by } y_i = r_i - s_i e_i \bmod q,$$

wherein x is the first verification value, y is the second verification value,
 v_i is the public key of the proving node P_i , and e_i is the random number
 generated for the proving node P_i .

With reference to the twelfth or the twentieth possible implementation manner of the first aspect, in a twenty-first possible implementation manner of the first aspect, the method further comprising: if the authentication of the plurality of proving nodes is determined unsuccessful,

5 classifying, by the verifying node, the plurality of proving nodes in the network into a plurality of subsets of proving nodes, and authenticating whether each subset of the proving nodes satisfy the predetermined authentication condition; or

performing, by the verifying node, existing Fiat-Shamir method to
10 determine whether each of the proving nodes in the network satisfy the authentication condition.

With reference to the first aspect or any one of the preceding possible implementation manner of the first aspect, in a twenty-second possible implementation manner of the first aspect, before the step of calculating a
15 plurality of first values, the method further comprising:

 sending, by the verifying node, an authentication request message (AR-message) to the each proving node directly or through a parent node directly connected to the each proving node.

According to a second aspect of the invention, a method for performing
20 collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto is provided. The method comprising:

calculating, by the verifying node, a first verification value based on a plurality of first values, wherein each of the first values is individually calculated by one of the plurality of proving nodes in the network based on a predetermined public system parameter and a random value selected by the proving node;

5 calculating, by the verifying node, a second verification value based on a plurality of second values, wherein each of the second value is individually calculated by one of the plurality of proving nodes in the network based on a random number generated for the proving node, the random value selected by the proving node and the predetermined public system parameter; and

10 determining, by the verifying node, whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the second verification value and the random numbers generated for the plurality of proving nodes in the network satisfy a predetermined authentication condition.

With reference to the second aspect, in a first possible implementation
15 manner of the second aspect, the step of calculating the first verification value based on a plurality of first values, further comprises:

calculating, by the verifying node, the first verification value based on at least one first unit value respectively received from at least one proving node directly connected to the verifying node, wherein each of the at least one first
20 unit value is calculated based on a first value calculated by a proving node directly connected to the verifying node and at least one first value respectively

calculated by at least one child node of the proving node directly connected to the verifying node.

With reference to the second aspect or the first possible implementation manner of the second aspect, in a second possible implementation manner of the second aspect, the step of calculating the second verification value based on
5 a plurality of second values, further comprising:

calculating, by the verifying node, the second verification value based on at least one second unit value respectively received from at least one proving node directly connected to the verifying node, wherein each of the at least one
10 second unit value is calculated based on individually second value calculated by a proving node directly connected to the verifying node and at least one second value respectively calculated by at least one child node of the proving node directly connected to the verifying node.

With reference to the second aspect or any one of the first and second
15 possible implementation manner of the second aspect, in a third possible implementation manner of the second aspect, the method further comprising:

generating, by the verification node, a random number for each proving node in the network and sending the generated random number to the each proving node in the network.

20 With reference to the second aspect, or any one of the first and second possible implementation manner of the second aspect, in a fourth possible implementation manner of the second aspect, the method further comprising:

generating, by the verifying node, a root random value and sharing the generated root random value in the network;

wherein the random number for each proving node in the network is generated by the each proving node based on the root random value and a pre-determined function.

According to a third aspect of the invention, a method for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto is provided. The method comprises:

calculating, by each proving node in the network, a first value based on a predetermined public system parameter and a random value selected by the each proving node and sending the calculated first value to a parent node directly connected to the each proving node; and

calculating, by the each proving node in the network, a second value based on a random number generated for the each proving node, the random value selected by the each proving node, the private key of the each proving node, and the predetermined public system parameter, and sending the calculated second value to a parent node directly connected to the each proving node.

With reference to the third aspect, in a first possible implementation manner of the third aspect, the method further comprising:

receiving, by the each proving node in the network, a root random value from the verifying node, and

generating, by the each proving node, the random number for calculating the second value for the each proving node based on the received root random value and a predefined function.

According to a fourth aspect of the invention, a system for performing collective authentication in a communication network is provided. The system comprises:

a verifying node and a plurality of proving nodes,

wherein each of the proving nodes is configured to calculate a first value based on a predetermined public modulus value and a random value selected by the each of the proving nodes; and calculate a second value based on a random number generated for the each of the proving nodes, the random value selected by the each of the proving nodes, the private key of the each of the proving nodes, and the predetermined public modulus value; and

the verifying node is configured to calculate a first verification value based on a plurality of first values calculated by the plurality of proving nodes in the network; calculate a second verification value based on a plurality of second values calculated by the plurality of proving nodes in the network; and determine whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the second verification value and the random

numbers generated for the plurality of the proving nodes in the network satisfy a predetermined authentication condition.

With reference to the fourth aspect, in a first possible implementation manner of the fourth aspect, each of the proving nodes is further configured to
5 send a first unit value to a parent node of the each of the proving nodes in the network, wherein if one of the proving nodes in the network is directly connected to at least one child node in the network, the first unit value is calculated based on the first values calculated by both the proving node and the at least one child node of the proving node in the network; and if one of the proving nodes in the
10 network is a leave node, the first unit value is determined based on the first value calculated by the proving node;

wherein the verifying node is further configured to calculate the first verification value based on at least one first unit value, wherein each of the at least one first unit value is respectively calculated by at least one proving node
15 directly connected to the verifying node in the network.

With reference to the fourth aspect or the first possible implementation manner of the fourth aspect, in a second possible implementation manner of the fourth aspect, each of the proving nodes in the network is further configured to send a second unit value to a parent node of the each proving node in the
20 network, wherein if one of the proving nodes is directly connected to at least one child node in the network, the second unit value is calculated based on the second values calculated by both the proving node and the at least one child node of the proving node in the network; and if the proving node is a leave node,

the second unit value is determined based on the second value calculated by the proving node;

wherein the verifying node is further configured to calculate the second verification value based on at least one second unit value, wherein each of the at least one second unit value is respectively calculated by at least one proving node directly connected to the verifying node in the network.

With reference to the fourth aspect, or any one of the first and the second possible implementation manner of the fourth aspect, in a third possible implementation manner of the fourth aspect, the verifying node is further configured to generate the random number for the each proving node in the network and send the generated random number for the each proving node to the each proving node directly or through a parent node of the each proving node in the network.

With reference to the fourth aspect, or any one of the first to third possible implementation manner of the fourth aspect, in a fourth possible implementation manner of the fourth aspect, the verifying node is further configured to generate a root random value and sharing the generated root random value in the network;

wherein the each of the proving nodes is further configured to generate a random number for the each of the proving nodes based on the root random value and a pre-determined function.

36 With reference to the fourth aspect, or any one of the first to fourth possible implementation manner of the fourth aspect, in a fifth possible implementation manner of the fourth aspect, each proving node is further configured to calculate a public key or the private key of the each proving node; or the verifying node or a trusted party is configured to calculate the public key or the private key of the each proving node in the network.

With reference to the fifth possible implementation manner of the fourth aspect, in a sixth possible implementation manner of the fourth aspect, the each of the proving nodes or the trusted party is further configured to calculate the public key or private key of the each of the proving nodes based on the predetermined modulus value.

With reference to the sixth possible implementation manner of the fourth aspect, in a seventh possible implementation manner of the fourth aspect, the predetermined public modulus value is a product of a plurality of prime numbers wherein at least two of the prime numbers are different.

With reference to the sixth or seventh possible implementation manner of the fourth aspect, in an eighth possible implementation manner of the fourth aspect, each of the proving nodes or the trusted party is further configured to calculate the public key or private key for each proving node according to the following Equation:

$$v_{i,j} = \frac{1}{s_{i,j}^c} \bmod n, \text{ or } v_{i,j} = s_{i,j}^c \bmod n; \quad i = 1, \dots, u; j = 1, \dots, k$$

wherein $v_i=(v_{i,1}, \dots, v_{i,k})$ is the public key of the proving node P_i , $s_i=(s_{i,1}, \dots, s_{i,k})$ is the private key of the proving node P_i , n is the predetermined public modulus value, u is the number of the proving nodes in the network; k is a pre-set integer no less than 1; and c is a pre-set integer no less than 2.

5 With reference to the eighth possible implementation manner of the fourth aspect, in a ninth possible implementation manner of the fourth aspect, each of the proving nodes is further configured to calculate the first value according to the following equation:

$$x_i = r_i^c \bmod n$$

10 wherein x_i is the first value calculated by the proving node P_i , n is the predetermined public modulus value, r_i is the random value selected by the proving node P_i , c is the pre-set integer no less than 2.

 With reference to the ninth possible implementation manner of the fourth aspect, in a tenth possible implementation manner of the fourth aspect, the verifying node is further configured to calculate the first verification value
15 according to the following equation:

$$x = \prod_{i=1}^u x_i \bmod n$$

 wherein x is the first verification value, x_i is the first value calculated by the proving node P_i , n is the predetermined public modulus value, and u is the number of the proving nodes in the network.

With reference to the tenth possible implementation manner of the fourth aspect, in an eleventh possible implementation manner of the fourth aspect, each of the proving nodes in the network is further configured to calculate the second value according to the following equation:

$$y_i = r_i \prod_{j=1}^k s_{i,j}^{e_{i,j}} \bmod n$$

5 wherein the verifying node is further configured to calculate the second verification value is calculated according to the following equation:

$$y = \prod_{i=1}^u y_i \bmod n;$$

wherein y_i is the second value calculated by the proving node P_i , r_i is the random value selected by the proving node P_i , $s_i = (s_{i,1}, \dots, s_{i,k})$ is the private key
 10 of the proving node P_i , k is the pre-set integer $k \geq 1$, and $e_i = (e_{i,1}, \dots, e_{i,k})$ is the random number generated for the proving node P_i , wherein $e_{i,j}$ is the j -th bit of e_i , n is the predetermined public modulus value, y is the second verification value, u is the number of the proving nodes in the network.

With reference to the eleventh possible implementation manner of the
 15 fourth aspect, in a twelfth possible implementation manner of the fourth aspect, the verifying node is further configured to determine whether the authentication of the plurality of proving nodes is successful based on whether the first verification value, the second verification value and the random numbers generated for the plurality of proving nodes in the network satisfy the
 20 predetermined authentication condition indicated in the following equation:

$$x = y^c \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = \frac{1}{s_{i,j}^c} \bmod n$$

$$y^c = x \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = s_{i,j}^c \bmod n$$

wherein x is the first verification value, y is the second verification value, c is the pre-set integer $c \geq 2$, $v_i = (v_{i,1}, \dots, v_{i,k})$ is the the public key of the proving node P_i .

5 With reference to the any one of the fifth to twelfth possible implementation manner of the fourth aspect, in a thirteenth possible implementation manner of the fourth aspect, the private key of each proving node is derived from a random seed by evaluating of a cryptographically strong pseudorandom function with respect to the random seed and other relevant information.

10 With reference to the any one of the fifth to twelfth possible implementation manner of the fourth aspect, in a fourteenth possible implementation manner of the fourth aspect, the private key of each proving node is derived by the proving node itself or by a trusted party which has been informed the predetermined public modulus value.

15 With reference to the any one of the fifth to twelfth possible implementation manner of the fourth aspect, in a fifteenth possible implementation manner of the fourth aspect, the private key of each proving node is generated by randomly selecting a first group of words from a predetermined second group of words, and calculating a product of the words in

the first group based on a preset table which consists of all pairwise products of the second group of words.

With reference to the fifth possible implementation manner of the fourth aspect, in a fifteenth possible implementation manner of the fourth aspect, each
5 of the proving nodes or the trusted party is further configured to calculate the public or private key for each proving node according to the following Equation:

$$v_i = g^{s_i}$$

wherein v_i is the public key of the proving node P_i , s_i is the private key of the proving node P_i , and g is the generator of a cyclic group G_q , which has a prime order q .

10 With reference to the fifteenth possible implementation manner of the fourth aspect, in a sixteenth possible implementation manner of the fourth aspect, the first value is calculated by each proving node according to the following equation:

$$x_i = g^{r_i}$$

wherein x_i is the first value calculated by the proving node P_i , g is the generator
15 of a cyclic group G_q , and r_i is the random value selected by the proving node P_i .

With reference to the sixteenth possible implementation manner of the fourth aspect, in a seventeenth possible implementation manner of the fourth aspect, the verifying node is further configured to calculate the first verification value according to the following equation:

$$x = \prod_{i=1}^u x_i = \prod_{i=1}^u g^{r_i}$$

wherein x is the first verification value, and u is the number of the proving nodes in the network.

With reference to the seventeenth possible implementation manner of the fourth aspect, in an eighteenth possible implementation manner of the fourth aspect, each of the proving nodes is further configured to calculate the second value according to the following equation:

$$y_i = r_i + s_i e_i \bmod q, \text{ or}$$

$$y_i = r_i - s_i e_i \bmod q$$

wherein the verifying node is further configured to calculate the second verification value according to the following equation:

$$y = \sum_{i=1}^u y_i \bmod q$$

wherein y_i is the second value calculated by the proving node P_i , r_i is the random value selected by the proving node P_i , s_i is the private key of the proving node P_i , e_i is the random number generated for the proving node P_i , y is the second verification value, u is the number of the proving nodes in the network, and q is the prime order of the cyclic group G_q .

With reference to the eighteenth possible implementation manner of the fourth aspect, in a nineteenth possible implementation manner of the fourth aspect, the verifying node is further configured to determine whether the authentication of the plurality of proving nodes is successful based on whether

the first verification value, the second verification value and the random numbers generated for the plurality of proving nodes in the network satisfy the predetermined authentication condition indicated in the following equation:

$$g^y = x \prod_{i=1}^u v_i^{e_i}, \text{ if } y_i \text{ is computed by } y_i = r_i + s_i e_i \bmod q;$$

$$5 \quad x = g^y \prod_{i=1}^u v_i^{e_i}, \text{ if } y_i \text{ is computed by } y_i = r_i - s_i e_i \bmod q,$$

wherein x is the first verification value, y is the second verification value, v_i is the public key of the proving node P_i .

With reference to the twelfth or eighteenth possible implementation manner of the fourth aspect, in a twentieth possible implementation manner of
10 the fourth aspect, the verifying node is further configured to

if the authentication of the plurality of proving nodes is unsuccessful, classify the plurality proving nodes in the network into a plurality of subsets of proving nodes, and authenticate whether each subset of the proving nodes satisfy a predetermined authentication condition; or

15 perform the existing Fiat-Shamir method to determine whether each of the proving nodes in the network satisfy a predetermined authentication condition.

With reference to the fourth aspect, or any one of the first to twentieth possible implementation manner of the fourth aspect, the verifying node is further configured to send an authentication request message (AR-message) to
20 the each proving node directly or through a parent node directly connected to the each proving node in the network;

wherein each of the proving nodes is further configured to calculate the first value after receiving the authentication request message.

According to a fifth aspect of the invention, an apparatus for performing collective authentication in a communication network including a plurality of
5 proving nodes, is provided. The apparatus is configured to

calculate a first verification value based on a plurality of first values, wherein each of the first values is individually calculated by one of the plurality of proving nodes in the network based on a predetermined public system parameter and a random value selected by the proving node;

10 calculate a second verification value based on a plurality of second values, wherein each of the second value is individually calculated by one of the plurality of proving nodes in the network based on a random number generated for the proving node, the random value selected by the proving node, the private key of the proving node, and the predetermined public system parameter; and

15 determine whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the second verification value and the random number generated for each of the plurality of proving nodes in the network satisfy a predetermined authentication condition.

According to a sixth aspect of the invention, a proving node for performing
20 collective authentication in a communication network including a verifying node and a plurality of proving nodes, is provided. The proving node is configured to

calculate a first value based on a predetermined public system parameter and a random value selected by the proving node and sending the calculated first value to a parent node directly connected to the proving node in the network;

calculate a second value based on a random number generated for the
5 proving node, the random value selected by the proving node, the private key of the proving node and the predetermined public system parameter, and sending the calculated second value to the parent node directly connected to the proving node in the network.

Brief Description of the Drawings

10 The invention will be described in detail with reference to the accompanying drawings, in which:

Figure 1 is a flow chart illustrating a method for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto according to a first embodiment of
15 the invention;

Figure 2 illustrates the step of calculating a first value by each proving node and sending the first value to its parent node in the network according to one example of the first embodiment in Figure 1;

Figure 3 illustrates the step of generating a random vector by the verifying
20 node and sending the random vector to the proving nodes in the network according to one example of the first embodiment in Figure 1;

Figure 4 illustrates the step of calculating a second value by each proving node and sending the second value to its parent node in the network according to one example of the first embodiment in Figure 1;

Figure 5 is a flow chart illustrating the process of failed authentication according to one embodiment of the invention;

Figure 6 is a flow chart illustrating a method for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto according to a second embodiment of the invention.

10 Detailed Description of Embodiments of the Invention

In the following description, numerous specific details are set forth in order to provide a thorough understanding of various illustrative embodiments of the invention. It will be understood, however, to one skilled in the art, that embodiments of the invention may be practiced without some or all of these specific details. It is understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to limit the scope of the invention. In the drawings, like reference numerals refer to same or similar functionalities or features throughout the several views.

Embodiments of the invention provide a method for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto, the method at least comprises the following steps:

each proving node in the network calculates a first value for the each proving node based on a predetermined public system parameter and a random value selected by the each proving node;

the verifying node calculates a first verification value based on a plurality
5 of first values calculated by the plurality of proving nodes in the network;

a random number is generated for the each proving node in the network;

each proving node in the network calculates a second value for the each proving node based on the random number generated for the each proving node, the random value selected by the each proving node, the private key of
10 the each proving node, and the predetermined public system parameter;

the verifying node calculates a second verification value based on a plurality of second values calculated by the plurality of proving nodes in the network; and determines whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the second verification
15 value and the random numbers generated for the plurality of proving nodes in the network satisfy a predetermined authentication condition.

Figure 1 is a flow chart illustrating a method for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto according to a first embodiment of
20 the invention. In this embodiment, the communication network includes $u + 1$ nodes/devices, which are a verifying node/device V and u proving nodes/devices $P_1, \dots, P_u$. The $u + 1$ nodes/devices in the communication network

form a spanning tree W . The verifying node V is the root node in the spanning tree W and all of the nodes/devices in the spanning tree W know their parent nodes/devices and their child nodes/devices together with their indices, if any.

In Internet of Things(IoT) / Machine to Machine (M2M) scenarios, the
 5 verifying node/device V could be a server, gateway, router or suitable entity for authenticating a number of IoT/M2M proving nodes/devices/entities P_i . In wireless sensor networks (WSNs), the proving nodes/devices P_i may be a plurality of sensors and the verifying node/device may be a base station.

In this embodiment, the private key and public key of each proving node
 10 P_i is calculated based on the predetermined public system parameter and the identity of the each proving node P_i . Specifically, the private key and the public key of the each proving node P_i satisfy the Equation (1) or (2), i.e. the public or private key of each proving node P_i may be calculated according to the following Equation (1) or (2):

$$15 \quad v_{i,j} = \frac{1}{s_{i,j}^e} \bmod n; \quad i = 1, \dots, u; j = 1, \dots, k \quad (1)$$

$$v_{i,j} = s_{i,j}^c \bmod n; \quad i = 1, \dots, u; j = 1, \dots, k \quad (2)$$

wherein $v_i = (v_{i,1}, \dots, v_{i,k})$ is the public key of the proving node P_i , $s_i = (s_{i,1}, \dots, s_{i,k})$ is the private key of the proving node P_i , n is the predetermined public system parameter called public modulus which is a product of a plurality of
 20 prime numbers, wherein some of the prime numbers may be the same, or at least two of the prime numbers are different, u is the number of the proving

nodes in the network; k is a pre-set integer no less than 1; c is a pre-set integer no less than 2.

Here, the predetermined public system parameter n , the private or public key of each proving node may be generated in the same way as in the existing
5 Fiat-Shamir scheme/method, and n is shared by all the nodes in the network. The authentication process will be described hereinafter with reference to the Figures 2 to 4.

In **block** 101, optionally, the proving nodes are triggered to perform an authentication process with the verifying node V .

10 In this embodiment, the authentication process may be triggered by an authentication request message (AR-message) from the verifying node V , if the network has a plurality of layers/levels of proving nodes connected to the verifying node V . The verifying node V may first send an authentication request message (AR-message) to all of the first level proving nodes directly connected
15 to the verifying node to trigger the authentication process. Then, each of the first level proving nodes respectively sends the AR-message to its child nodes.

Referring to Figure 2, in one example of the embodiment, the proving nodes directly connected to the verifying node V include proving node P_4 and proving node P_7 . The proving node P_4 has three child nodes P_1 , P_2 and P_3 , the
20 proving node P_7 has two child nodes P_5 and P_6 . In this example, the verifying node V may first send the AR-message to P_4 and P_7 , then P_4 sends the AR-

message to its child nodes P_1 - P_2 ; P_7 sends the AR-message to its child nodes P_5 and P_6

The AR-message may contain a commitment to e for guaranteeing the protocol's zero-knowledge property even against dishonest verifying node. A
5 commitment to e is a value that hides e but at the same time the verifying node V cannot change the value of e when the verifying node V needs to open this commitment. Here, e is a challenge selected by the verifying node V , which will be specified below.

In other embodiments of the invention, the authentication process may be
10 triggered by a predetermined timing when the proving nodes perform collective or swarm authentication with the verifying node V automatically.

In **block 102**, each of the proving nodes in the network calculates a first value based on a predetermined public system parameter and a random value selected by the each proving node and sends the calculated first value to its
15 parent node.

In one example of this embodiment, each of the proving nodes may calculate the first value according to the following equation (3):

$$x_i = r_i^c \bmod n \quad (3)$$

Wherein x_i is the first value calculated by the proving node P_i , n is the
20 predetermined public modulus, r_i is the random value selected by the proving node P_i , c is the pre-set integer no less than 2. Referring to Figure 2, in this example, the first values respectively calculated by the proving nodes P_1 , P_2 and

P_3 are sent to their parent proving node P_4 ; the first values respectively calculated by the proving nodes P_5 and P_6 are sent to their parent proving node P_7 .

In **block 103**, each of the parent proving nodes in the network calculates
5 a first unit value based on the first values of its own and of its child nodes and sends the calculated first unit value to its parent proving node if any, or the verifying node V . The parent proving nodes include any proving node which has at least one child proving node in the communication network.

These recursive procedures are performed based on the number of
10 layers/levels of the spanning tree W in the network. In the example shown in Figure 2, the network has two layers of proving nodes, accordingly the recursive procedures are performed in two steps.

In the example shown in Figure 2, the parent proving nodes in the network include proving nodes P_4 and P_7 . The proving node P_4 calculates a first unit value
15 based on its own first value and the first values of its child nodes P_1 - P_3 . Specifically, the proving node P_4 calculates the first unit value by multiplying its own first value with all of the first values from the proving nodes P_1 - P_3 and sends the calculated first unit value to the verifying node V . Similarly, the proving node P_7 calculates a first unit value based on its own first value and the first values of
20 its child nodes P_5 and P_6 . Specifically, the proving node P_7 calculates the first unit value by multiplying its own first value with the first values from the proving nodes P_5 and P_6 and sends the calculated first unit value to the verifying node V .

In **block** 104, the verifying node V computes a first verification value based on at least one first unit value respectively calculated by at least one proving node directly connected to the verifying node V .

In the example shown in Figure 2, the first verification value is calculated
5 according to the following Equation (4):

$$x = \prod_{i=1}^u x_i \bmod n \quad (4)$$

wherein x is the first verification value, x_i is the first value calculated by the proving node P_i , n is the predetermined public modulus value, u is the number of the proving nodes in the network. In the example shown in Figure 2, the network
10 has 7 proving nodes, $u = 7$.

Alternatively, each of the parent proving nodes may only send the first values of its own and its child nodes to the verifying node V , and then the verifying node V will calculate the first verification value based on the Equation (4).

15 In **block** 105, a random number is generated for each proving node in the communication network.

In embodiments of the invention, the random number for each proving node in the network may be generated by the verifying node and sent to each proving node in the network. Alternatively, the random number for each proving
20 node may be generated by the proving node based on a root random value generated and shared by the verifying node.

As shown in Figure 3, in one embodiment of the invention, a random vector $e = (e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ is generated by the verifying node V , and sent as an authentication challenge (AC-message) to the proving nodes P_4 and P_7 directly connected to the verifying node. Then, the proving node P_4 sends the
5 random vector e to each of its child nodes P_1 , P_2 and P_3 ; the proving node P_7 sends the random vector e to each of its child devices P_5 and P_6 .

It should be noted that the purpose of this step is to inform or provide each of the proving nodes in the network a random number generated for it. Various ways may be adopted to realize this purpose. For example, the whole
10 random vector e may be shared to each of the proving nodes in the network and each of the proving nodes in the network identifies from the received random vector a corresponding random number generated for itself. Alternatively, for each of the proving nodes in the network, only the relevant part of the random vector is shared, e.g. if the proving node is a parent proving node in the network,
15 then only random number generated for itself and its child proving nodes may be sent to the proving node, or if the proving node has no child node in the network, i.e. the proving node is a leave node, then only the corresponding random number generated for this proving node is sent to it.

In **block 106**, each of the proving nodes in the network calculates a
20 second value based on the random value selected by the each proving node, the private key of the each proving node, and the random number generated for the each proving node and sends the calculated second value to its parent node in the network.

In one embodiment of the invention, each of the proving nodes calculates the second value according to Equation (5)

$$y_i = r_i \prod_{j=1}^k s_{i,j}^{e_{i,j}} \bmod n \quad (5)$$

wherein n is the predetermined public modulus, r_i is the random value
 5 selected by the proving node P_i , $s_i = (s_{i,1}, \dots, s_{i,k})$ is the private key of the proving node P_i , k is the pre-set integer $k \geq 1$, $e_i = (e_{i,1}, \dots, e_{i,k})$ is the random number generated for the proving node P_i .

In **block 107**, each of the parent proving nodes in the network calculates a second unit value based on the second values of its own and its child devices
 10 and sends the calculated second unit value to its parent node if any, or the verifying node V .

These recursive procedures are performed based on the number of layers/levels of the spanning tree W in the network. In the example shown in Figure 2, the network has two layers of proving nodes, accordingly the recursive
 15 procedures are performed in two steps.

In **block 108**, the verifying node calculates a second verification value based on at least one second unit value wherein each of the at least one second unit value is respectively calculated by at least one proving node directly connected to the verifying node in the network, i.e. by at least one child node of
 20 the verifying node in the network.

In one example of this embodiment, the verifying node calculates the second verification value according to the following equation (6):

$$y = \prod_{i=1}^u y_i \bmod n; \quad (6)$$

wherein y is the second verification value, y_i is the second value
 5 calculated by the proving node P_i , n is the predetermined public modulus value,
 u is the number of the proving nodes in the network.

In **block 109**, the verifying node V determines whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the second verification value and the random number generated for each
 10 proving node satisfy a predetermined authentication condition.

Specifically, if the first verification value, the second verification value and the random number generated for each proving node satisfy the predetermined authentication condition, the verifying node determines that the authentication of the plurality of proving nodes is successful, i.e. all of the proving nodes are
 15 authenticated successfully.

In one example of this embodiment, the verifying node determines whether authentication of the plurality of proving nodes is successful according to the following Equation (7) or (8), i.e. the predetermined authentication condition:

$$x = y^c \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = \frac{1}{s_{i,j}^c} \bmod n \quad (7)$$

$$y^c = x \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = s_{i,j}^c \bmod n \quad (8)$$

wherein x is the first verification value, y is the second verification value, c is the pre-set integer $c \geq 2$, $v_i = (v_{i,1}, \dots, v_{i,k})$ is the the public key of the proving node P_i .

If the verifying node V determines that all of the proving nodes are
5 authenticated successfully, the verifying node V may output a set of authenticated proving nodes (AN).

It should be noted that in order to enhance security and/or reduce the size of each proving node's private key, the above authentication process may be run or performed more than one time.

10 For a predefined security parameter λ , which is at least 20. To ensure this security level, the following constraints on parameters are suggested:

- The authentication protocol should be run or performed t times so that $t \geq \lambda/k$, where k denotes the number of each proving node's private keys. This number is reasonably close to one when c is close to λ ;

15 - To achieve a marginal 20 bit security level for online authentication, one can require that $tk \geq \lambda \geq 20$, which means that an attacker can impersonate a node to go through the authentication by guessing the correct challenge vector e with success rate about once in one million tries;

- The predetermined public system parameter, i.e. public modulus value
20 n should be at least 512 bits.

It is to be appreciated by a skilled person that each proving node's public key may be certified by another entity via any suitable means, e.g., issued by a trusted party (usually called Certification Authority) as in a public or proprietary Public Key Infrastructure (PKI), pre-installed by the manufacturer, configured by the owner, stored in a key directory which is accessible to all legitimate nodes and users. In order to achieve mutual authentication, the verifying device V may also authenticate itself to the proving nodes by using its private key to sign a message which is newly generated.

The total number of operations required to authenticate the network depends on the exact topology at hand, but can safely be bounded by the following conditions:

- Number of modular squarings is $(2tu + 1)$
- Number of modular multiplications is less than $(2u + 2uk)t$.

On average, each proving node only performs a constant number of operations. Finally, only $O(d)$ (namely, a number linear in d) messages are sent, where d is the degree of W which is the order of $\log u$, so that only a logarithmic number of messages are sent during authentication.

Memory is a scarce resource for many lightweight IoT devices, so in some scenarios it may be a challenge for each node P_i to store the private key $s_i = (s_{i,1}, \dots, s_{i,k})$, which is up to $k|n|$ bits. To reduce memory requirements, each $s_{i,j}$ can be selected as a short random number, e.g. 80 bits. Moreover, a node P_i can even select a random seed b_i and then derive each of its private key values

$s_{i,j}$ by evaluating of a cryptographically strong pseudo random function G with respect to the random seed b_i and other information, e.g., index j . For example, each P_i can set $s_{i,j} = G(b_i, j)$. In practice, G can be a cryptographic hash function. Once $s_{i,j}$ is generated from the random seed b_i , each corresponding
5 public key value v_i is still calculated by Equation (1).

Note that as the drawback of this variant, each time of authentication a proving node needs to recover its private key vector from the random seed b_i by evaluating function G for k times.

In this embodiment of the invention, the uk -bit challenge $e = (e_1, \dots, e_u)$
10 is sent throughout the network to all individual nodes. One way to reduce the length of e without compromising security is the following:

- the verifying node V selects a short value e , e.g. 80-bit and sends it to the nodes.

- Each proving node P_i computes the challenges e_i for itself by evaluating
15 the value of a cryptographically strong pseudo random function H with respect to the input including e and its index or identity. For example, we can set $e_i = H(e, i)$. In practice, H can be a cryptographic hash function.

- The verifying device V also computes challenges e_i the same way, and uses these challenge values to check the validity of (x, y) according to Equation
20 (7) or (8).

This variant does not impact security, assuming an ideal pseudo random function H , and it can be used in conjunction with the other improvements described below.

Similar to the existing Fiat-Shamir scheme, the authentication process disclosed in embodiments of the invention can be transformed into an identity-based scheme by assuming that there is a trusted party T who knows the prime factors of the public modulus n and is responsible to extract the private key $s_{i,j}$ for each proving node.

In detail, T can first derive the public key of each proving node P_i from P_i 's identity information ID_i and index j , and other possible information, by using a cryptographically strong pseudo random function F . For example, by defining the public key $v_{i,j} = F(ID_i, j)$, T can extract the private key $s_{i,j}$ from $v_{i,j}$ according to Equation (1) or Equation(2), by using its secret knowledge of modulus n 's prime factors. Note that in this variant, some $v_{i,j}$ derived from using function F may need to be discarded, though such a treatment will not affect a verifying device to recover a node P_i 's public key.

In order to further reduce computational cost of the authentication process, in one embodiment of the invention, the following is performed:

- Each proving node P_i chooses an alphabet of m words $w_1, \dots, w_m$, in which a word is a 32-bit value, and computes once and for the lookup table of all pairwise products $p_{i,j} = w_i w_j$. Note that each entry $p_{i,j}$ is 64 bit long.

- The values $s_{i,j}$'s are generated by randomly sampling from this alphabet m' times. Namely, $s_{i,j}$ is built by concatenating m' words (bit patterns) taken from the alphabet only.
- So, the values $s_{i,j}$'s, which are each $32m'$ -bit integers, can take $m^{m'}$ possible values.

To compute the response y_i , each proving node P_i has to calculate a number of values $s_{i,j}s_{i,l}$, which can be calculated by looking up the table consisting of all values $p_{i,j}$. This provides a substantial speed-up over the naive approach.

For instance, if $m = m' = 32$, then each s_i is a 1024-bit number chosen amongst $32^{32} = 2^{160}$ possible values.

The size of the lookup table is moderate, for the example given, all we need to store is $32 \times (32 + 1)/2 = 528$ values due to its symmetry. Because each entry is 64 bits, the lookup table size boils-down to 4224 bytes only.

Moreover, this size can be further reduced if these m words are selected according to some specific rules, like the first smallest m words.

The idea is that computational cost can be reduced if some products are pre-computed and stored, only to assemble them online during the authentication process: the values of $s_{i,a,b}$ defined by $s_{i,a,b} = s_{i,a}s_{i,b} \bmod n$, for any $1 \leq a, b \leq k$, are stored in a lookup table. The use of combined values of $s_{i,a,b}$ in the evaluation of value y_i results in three possible scenarios for each:

1. Both $s_{i,a}$ and $s_{i,b}$ appear in y_i , the probability of this occurring is $1/4$, in which case one additional multiplication must be performed;
2. Neither of $s_{i,a}$ and $s_{i,b}$ appears in y_i , the probability of this occurring is $1/4$, in which case no action is performed;
- 5 3. Either of $s_{i,a}$ or $s_{i,b}$, but not both, appears in y_i , this happens with probability $1/2$, in which case one single multiplication is required.

As a result the expected number of multiplications is reduced by 25%, to wit $(3/4)2^{k-1}$ where k is the size of e .

The method can be extended to work in a window of size $\gamma > 1$. For
 10 instance, with $\gamma = 3$, the following values for each $0 \leq l < k/3$ are pre-computed:

$$s_{i,3l+1,3l+2} = s_{i,3l+1}s_{i,3l+2} \bmod n,$$

$$s_{i,3l+2,3l+3} = s_{i,3l+2}s_{i,3l+3} \bmod n,$$

$$s_{i,3l+1,3l+3} = s_{i,3l+1}s_{i,3l+3} \bmod n,$$

$$s_{i,3l+1,3l+2,3l+3} = s_{i,3l+1}s_{i,3l+2}s_{i,3l+3} \bmod n.$$

15 Following the same analysis as above, the expected number of multiplications during the challenge-response phase is $(7/24)2^k$. The price to pay is that larger values of γ require more precomputing and memory. More precisely, by writing $\mu = 2^k \bmod \gamma$ and using the notation $[q]$ to denote the integer part of the number q we have the following trade-offs, where L is the number of
 20 components of s_i :

- The expected number of multiplications is $2^k \left(\frac{2^{\gamma-1}}{2^\gamma} \left(\frac{2^k}{\gamma} - 1 \right) - \frac{2^\mu - 1}{2^\mu} \right)$;

- The number of pre-multiplications is
- $L - 1 + (2^\gamma - \gamma - 1) \left\lfloor \frac{2^k}{\gamma} \right\rfloor + (2^\mu - \mu - 1);$
- The number of stored values in the lookup table is $(2^\gamma - 1) \left\lfloor \frac{2^k}{\gamma} \right\rfloor + (2^\mu - 1).$

5 The authentication process may be adapted to better fit operational constraints: in the context of IoT for instance communication outcoming from nodes is a very costly operation. It is a further goal of the invention to describe variants that aim at reducing the amount of information sent, the size of memory and/or the amount of computation by individual nodes, while maintaining
10 security.

To facilitate abnormal process, the indexes of proving nodes and verifying node may be transferred explicitly or implicitly with messages among the nodes in the network.

15 If the first verification value, the second verification value and the random number generated for each proving node do not satisfy the predetermined authentication condition, then the authentication process fails and the verifying node V may perform the process of failed authentication as described below.

20 The above authentication process may fail due to different reasons, which can be accidentally or maliciously introduced at any step. As a consequence of the algorithm's distributed nature that has been described above, a single defective node suffices for authentication failure. According to one embodiment of the invention, the verifying node may have several optional

process of failed authentication to perform including giving up/terminating the authentication procedures, running/performing the authentication process again with all of the proving nodes, identifying the subset of proving nodes which failed authentication, or performing the existing Fiat-Shamir authentication process
5 with each proving node in the network individually. This process of failed authentication is illustrated in FIG. 5, and described as follows.

In **block 501**, the verifying node V determines whether the first verification value x , the second verification value y and the random number for each proving node satisfy the predetermined authentication condition indicated in Equation (7)
10 or (8), if yes, the flow sequence proceeds to block 502; if no, the flow sequence proceeds to block 503.

In **block 502**, the verifying node V adds all the proving nodes $P_1, \dots, P_u$ into the set AN (Authenticated Nodes), which was initialized as empty. This means that all of the proving nodes in the network are authenticated
15 successfully, and the authentication process is completed normally. Otherwise, at least one node failed for the authentication. V may choose different methods for the process of failed authentication.

In **block 503**, the verifying node V determines whether to give up/terminate the authentication process. If yes, the flow proceeds to block 504; if
20 no, the flow sequence proceeds to block 505.

In **block 504**, all the proving nodes are not authenticated and the AN is unchanged.

In **block 505**, the verifying node V determines whether to perform the authentication process again. If yes, the flow sequence proceeds to the block 501 to determine whether the first verification value x , the second verification value y and the random number for each proving node satisfy the predetermined authentication condition indicated in equation (7) or (8); if no, the flow sequence
5 proceeds to block 506.

In **block 506**, if the verifying node V chooses to classify the proving nodes in the network into a plurality of subsets and perform the authentication process on each subset of the proving nodes separately, the flow sequence proceeds to
10 block 507; otherwise, the flow sequence proceeds to block 511.

In **block 507 to block 510**, the verifying node V determines whether the first verification value x and the second verification value y calculated for the current subset of the proving nodes satisfies the predetermined authentication condition indicated in equation (7) or (8). If yes, the verifying node V adds the
15 current subset of proving nodes to the set AN ; if no, the set AN is unchanged. If the current subset of proving nodes is the last subset, the set AN is output, if the current subset of proving nodes is not the last subset, the flow sequence proceeds to the block 507.

In **block 511**, if the verifying node V chooses to authenticate the proving
20 nodes in the network separately using the existing Fiat-Shamir method, the flow sequence proceeds to block 512.

In **block 512-block 515**, if the verifying node V determines that the current proving node is authenticated, the verifying node V adds the current proving node to the set AN; if not, the set AN is unchanged. If the current proving node is the last proving node, the verifying node V outputs the set AN; if the
5 current proving node is not the last proving node, the flow sequence proceeds to the block 512 to authenticate the next proving node.

It should be noted that the embodiment shown in Figure 5 is only used for illustrating the various optional process of failed authentication. It is not used to limit the scope of the invention. In particular, the order of the steps in blocks 503,
10 505, 506, and 511 may be swapped in other embodiments of the invention as they are independent.

In this embodiment, given the output AN, all the proving nodes not in AN are classified as unsuccessfully authenticated nodes.

In order to authenticate each subset of proving nodes, it is assumed that
15 the verifying node V knows or is informed by each first level parent node, i.e. the nodes directly connected to the verifying node V about the indexes of all the proving nodes in such a subset. Moreover, if necessary, for a subset of proving nodes who has failed in the authentication process, each first level parent proving node can similarly find out which child proving node of the first level
20 parent proving node is responsible for the failure, though here the first level parent proving nodes are assumed to know the public keys of their child proving nodes. The child proving node of the first level parent proving node may be a leave node, i.e. the proving node has no child node in the network, or a parent

node which has at least one child node in the network. Even more, this process can go further with respect to lower parent proving nodes gradually to exactly identify all of the individual proving nodes who has failed in the authentication one by one.

5 In order to enable the existing Fiat-Shamir scheme as a back-up authentication, there is no real additional burden in implementing this solution. This is due to the fact that all the proving nodes already embark the hardware and software required for Fiat-Shamir computations, and can use the same public system parameters, e.g. the public modulus.

10 The basic ideas for the above collective/swarm authentication process can be expended to some other identification schemes which consist of commitment, challenge, and response. The following shows how the invention are applicable to Schnorr identification scheme, (C. P. Schnorr, Efficient identification and signatures for smart cards, in G. Brassard, ed. Advances in
15 Cryptology – Crypto'89, pp. 239–252, Springer-Verlag, 1990. Lecture Notes in Computer Science 435), such that a verifying device in a network can efficiently authenticate a plurality of devices or nodes collectively.

Similar to the swarm authentication based Shamir-Fiat algorithm, the invention discussed above can be straightforwardly extended to distributed Schnorr
20 identification scheme, where a verifying device V can collectively authenticate u nodes $P_1, \dots, P_u$ within a network in an efficient way.

Figure 6 is a flow chart illustrating the method for performing collective authentication in a communication network including a verifying node and a plurality of proving nodes according to another embodiment of the invention. In this embodiment, the public or private key for each proving node is calculated
5 according to the following Equation (9):

$$v_i = g^{s_i} \quad (9)$$

wherein v_i is the public key of the proving node P_i , s_i is the private key of the proving node P_i , g is the generator of a cyclic group G_q , which has a prime order q . The binary length of the prime order q of the cyclic group G_q is at least 80 bits.

10 The steps similar to the embodiment shown in Figure 1 will not be described in detail here.

In **block 601**, a collective swarm authentication process between the verifying node V and the proving nodes is triggered.

In **block 602**, each of the proving nodes calculates a first value according
15 to the following Equation (10), and sends the calculated first value to its parent node in the network.

$$x_i = g^{r_i} \quad (10)$$

wherein x_i is the first value calculated by the proving node P_i , g is the generator of a cyclic group G_q , r_i is the random value selected by the proving node P_i .

20 In **block 603**, each of the parent proving nodes in the network calculates a first unit value based on the first values of its own and its child nodes and

sends the calculated first unit value to its parent proving node if any, or the verifying node V .

In **block 604**, the verifying node V computes a product of all the first unit values received from its child nodes in the communication network to obtain a
 5 first verification value, as shown in Equation (11):

$$x = \prod_{i=1}^u x_i = \prod_{i=1}^u g^{r_i} \quad (11)$$

wherein x is the first verification value, u is the number of the proving nodes in the network.

In **block 605**, a random number is generated for each proving node in the
 10 network.

In **block 606**, each of the proving nodes in the network calculates a second value according to Equation (12) or (13) and sends the calculated second value to its parent node in the network.

$$y_i = r_i + s_i e_i \bmod q, \text{ or} \quad (12)$$

$$15 \quad y_i = r_i - s_i e_i \bmod q. \quad (13)$$

wherein r_i is the random value selected by the proving node P_i , s_i is the private key of the proving node P_i , e_i is the random number generated for the proving node P_i .

In **block 607**, each of the parent proving nodes in the network calculates
 20 a second unit value based on the second values of its own and its child devices

and send the calculated second unit value to its parent device if any, or the verifying node V .

In **block 608**, the verifying node calculates a second verification value according to the following Equation (14):

$$y = \sum_{i=1}^u y_i \bmod q \quad (14)$$

wherein y is the second verification value, y_i is the second value calculated by the proving node P_i , u is the number of the proving nodes in the network, and q is the prime order of the cyclic group G_q

In **block 609**, the verifying node V determines whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the second verification value and the random number generated for each proving node satisfy a predetermined authentication condition.

The predetermined authentication condition is shown in the following Equation (15) or (16):

$$g^y = x \prod_{i=1}^u v_i^{e_i}, \text{ if } y_i \text{ is generated by Equation (12)} \quad (15)$$

$$x = g^y \prod_{i=1}^u v_i^{e_i}, \text{ if } y_i \text{ is generated by Equation (13)} \quad (16)$$

wherein x is the first verification value, y is the second verification value, v_i is the public key of the proving node P_i , and e_i is the random number generated for the proving node P_i .

The process of failed authentication aforementioned and illustrated in Figure 5 can be adopted in the second embodiment. It will not be described in detail here.

It is to be noted that the underlying cyclic group G_q of prime order q can be
5 selected as a sub-group of Z_p , i.e., the multiplicative group of integers modular p , where p is a large prime with at least 512 bits and q is a divisor of $(p - 1)$.

It is to be noted that the underlying cyclic group G_q of prime order q can be denoted additively, rather than multiplicatively. Especially, G_q can be selected as a cyclic group of points defined on an elliptic curve.

10 It is to be noted that to guarantee security, the binary length of q , which is the prime order of the underlying cyclic group G_q , is at least 80.

It is to be noted that each node's public key v_i can be certified by another entity via any suitable means, e.g., issued by a trusted party (usually called Certification Authority) as in a public or proprietary PKI (Public Key
15 Infrastructure), pre-installed by the manufacturer, configured by the owner, stored in a key directory which is accessible to all legitimate nodes and users.

It is to be noted that v_i can be even defined as an identity based public key derived from node P_i 's identity (and other parameters), by assuming that a system administrator (usually called as a private key generator) signs each
20 node's identity to generate a signature, which is treated as the private key for the node. In particular, such a signature can be generated by using Schnorr signature scheme, or other Diffie-Hellman key based signature schemes, as

studied by Mihir Bellare et al. (Security Proofs for Identity-Based Identification and Signature Schemes, in the proceedings of EUROCRYPT 2004, pp. 268-286, Lecture Notes in Computer Science 3027, Springer, 2004.)

It is to be noted that to achieve mutual authentication, the verifying node V
5 may also authenticate itself to the proving nodes $P_1, \dots, P_u$ by using its private key to sign a message which is newly generated.

According to embodiments of the invention described above, the authentication process disclosed in the invention can be used to perform swarm authentication of a plurality of proving nodes simultaneously in a communication
10 network. Furthermore, each proving node in the communication network only transmits information to its closest neighbors, i.e. its parent node/child node(s), to be verified by the verifying node, therefore the transmitting energy to the verifying node is significantly decreased.

It is to be understood that the embodiments and features described above
15 should be considered exemplary and not restrictive. For example, the above-described embodiments may be used in combination with each other. Many other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention. The scope of the invention should, therefore, be determined with reference to the appended claims, along
20 with the full scope of equivalents to which such claims are entitled. Furthermore, certain terminology has been used for the purposes of descriptive clarity, and not to limit the disclosed embodiments of the invention.

Claims

1. A method for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto, the method comprising:

5 calculating, by each proving node in the network, a first value for the each proving node based on a predetermined public system parameter and a random value selected by the each proving node;

 sending, by the each proving node in the network, the calculated first value to its parent node;

10 calculating, by the verifying node, a first verification value based on a product of the plurality of first values calculated by the plurality of proving nodes in the network;

 generating a random number for the each proving node in the network;

 calculating, by the each proving node in the network, a second value for the
15 each proving node based on the random number generated for the each proving node, the random value selected by the each proving node, a private key of the each proving node, and the predetermined public system parameter;

 sending, by the each proving node in the network, the calculated second value to its parent node;

calculating, by the verifying node, a second verification value based on a product of the plurality of second values calculated by the plurality of proving nodes in the network; and

determining, by the verifying node, whether authentication of the plurality of proving nodes is successful based on whether the first verification value, the
5 second verification value and the random numbers generated for the plurality of proving nodes in the network satisfy a predetermined authentication condition.

2. The method according to claim 1, further comprising:

if one of the proving nodes in the network is directly connected to at least
10 one child node, sending, by the proving node in the network, a first unit value to its parent node , wherein the first unit value is calculated based on a product of the first values calculated by both the proving node and the at least one child node of the proving node;

wherein the step of calculating a first verification value based on a product
15 of the plurality of first values calculated by the plurality of proving nodes, further comprises:

calculating, by the verifying node, the first verification value based on at least one first unit value, wherein each of the at least one first unit value is respectively calculated by at least one proving node directly connected to the
20 verifying node in the network.

3. The method according to claim 1 or 2, further comprising:

if one of the proving nodes in the network is directly connected to at least one child node, sending, by the proving node in the network, a second unit value to its parent node, wherein the second unit value is calculated based on a product of the second values calculated by both the proving node and the at least one child
5 node of the proving node;

wherein the step of calculating a second verification value based on a product of the plurality of second values calculated by the plurality of proving nodes, further comprises:

calculating, by the verifying node, the second verification value based on at
10 least one second unit value, wherein each of the at least one second unit value is respectively calculated by at least one proving node directly connected to the verifying node in the network.

4. The method according to any one of claim 1 to claim 3, wherein the step of generating a random number for the each proving node in the network, further
15 comprises:

generating, by the verifying node, a random number for the each proving node in the network; and

sending the generated random number for the each proving node to the each proving node directly or through a parent node of the each proving node in
20 the network.

5. The method according to any one of claim 1 to claim 3, wherein the step of generating a random number for the each proving node in the network, further comprises:

generating, by the verifying node, a root random value and sharing the
5 generated root random value in the network;

generating, by the each proving node, a random number for the each proving node based on the root random value and a pre-determined function.

6. The method according to any preceding claim, further comprising:

calculating, by the each proving node or the verifying node or a trusted
10 party, a public key or the private key of the each proving node.

7. The method according to claim 6, wherein the public key or the private key of the each proving node is calculated based on the predetermined public system parameter and an identity of the each proving node.

8. The method according to claim 7, wherein the predetermined system parameter
15 is a public modulus which is a product of a plurality of prime numbers, wherein at least two of the prime numbers are different.

9. The method according to claim 7 or claim 8, wherein the public key or private key for the each proving node is calculated according to the following Equation:

$$v_{i,j} = \frac{1}{s_{i,j}^c} \bmod n, \text{ or } v_{i,j} = s_{i,j}^c \bmod n; \quad i = 1, \dots, u; j = 1, \dots, k$$

wherein $v_i=(v_{i,1}, \dots, v_{i,k})$ is the public key of the proving node P_i , $s_i=(s_{i,1}, \dots, s_{i,k})$ is the private key of the proving node P_i , n is the predetermined public system parameter called public modulus, u is the number of the proving nodes in the network, k is a pre-set integer no less than 1, and c is a pre-set integer no less than 2.

10. The method according to claim 9, wherein the first value is calculated by the each proving node in the network according to the following equation:

$$x_i = r_i^c \bmod n$$

wherein x_i is the first value calculated by the proving node P_i , n is the predetermined public modulus value, r_i is the random value selected by the proving node P_i , c is the pre-set integer no less than 2.

11. The method according to claim 10, wherein the first verification value is calculated according to the following equation:

$$x = \prod_{i=1}^u x_i \bmod n$$

wherein x is the first verification value, x_i is the first value calculated by the proving node P_i , n is the predetermined system parameter called public modulus, u is the number of the proving nodes in the network.

12. The method according to claim 11, wherein the second verification value is calculated according to the following equation:

$$y = \prod_{i=1}^u y_i \bmod n;$$

wherein y is the second verification value, y_i is the second value calculated by the proving node P_i , n is the predetermined public modulus value, and u is the number of the proving nodes in the network, wherein the second value y_i is calculated according to the following equation:

$$y_i = r_i \prod_{j=1}^k s_{i,j}^{e_{i,j}} \bmod n$$

wherein n is the predetermined public modulus value, r_i is the random value selected by the proving node P_i , $s_i=(s_{i,1}, \dots, s_{i,k})$ is the private key of the proving node P_i , k is the pre-set integer $k \geq 1$, and $e_i=(e_{i,1}, \dots, e_{i,k})$ is the random number generated for the proving node P_i , wherein $e_{i,j}$ is the j -th bit of e_i .

13. The method according to claim 12, wherein the step of determining whether authentication of the plurality of proving nodes is successful, further comprises:

determining the authentication of the plurality of proving nodes is successful if the following equation is satisfied:

$$x = y^c \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = \frac{1}{s_{i,j}^c} \bmod n$$

$$y^c = x \prod_{i=1}^u \prod_{j=1}^k v_{i,j}^{e_{i,j}} \bmod n, \text{ if } v_{i,j} = s_{i,j}^c \bmod n$$

wherein x is the first verification value, y is the second verification value, c is the pre-set integer $c \geq 2$, $v_i=(v_{i,1}, \dots, v_{i,k})$ is the public key of the proving node P_i , and $e_i=(e_{i,1}, \dots, e_{i,k})$ is the random number generated for the proving node P_i .

14. A method for performing collective authentication in a communication network having a verifying node and a plurality of proving nodes connected thereto, the method comprising:

calculating, by the verifying node, a first verification value based on a product of a plurality of first values calculated by the plurality of proving nodes in the network, wherein each proving node in the network calculates a first value for the each proving node based on a predetermined public system parameter and a
5 random value selected by the each proving node, and sends the calculated first value to its parent node;

calculating, by the verifying node, a second verification value based on a product of a plurality of second values calculated by the plurality of proving nodes in the network, wherein each proving node in the network calculates a second
10 value for the each proving node based on a random number generated for the each proving node, the random value selected by the each proving node, a private key of the each proving node, and the predetermined public system parameter, and sends the calculated second value to its parent node; and

determining, by the verifying node, whether authentication of the plurality of
15 proving nodes is successful based on whether the first verification value, the second verification value and the random numbers generated for the plurality of proving nodes in the network satisfy a predetermined authentication condition.

15. The method according to claim 14, wherein the step of calculating the first verification value based on a product of a plurality of first values calculated by the
20 plurality of proving nodes in the network, further comprises:

calculating, by the verifying node, the first verification value based on at least one first unit value respectively received from at least one proving node

directly connected to the verifying node, wherein each of the at least one first unit value is calculated based on a product of the first value calculated by a proving node directly connected to the verifying node and at least one first value respectively calculated by at least one child node of the proving node directly
5 connected to the verifying node.

16. The method according to claim 14 or claim 15, wherein the step of calculating the second verification value based on a product of a plurality of second values calculated by the plurality of proving nodes in the network, further comprising:

calculating, by the verifying node, the second verification value based on
10 at least one second unit value respectively received from at least one proving node directly connected to the verifying node, wherein each of the at least one second unit value is calculated based on a product of the second value calculated by a proving node directly connected to the verifying node and at least one second value respectively calculated by at least one child node of the
15 proving node directly connected to the verifying node.

17. The method according to any one of claim 14 to claim 16, further comprising:

generating, by the verification node, a random number for each proving node in the network and sending the generated random number to the each proving node in the network.

20 18. The method according to any one of claim 14 to claim 16, further comprising:

generating, by the verifying node, a root random value and sharing the generated root random value in the network;

wherein the random number for each proving node in the network is generated by the each proving node based on the root random value and a pre-determined function.

19. A method for performing collective authentication in a communication
5 network having a verifying node and a plurality of proving nodes connected thereto, the method comprising:

calculating, by each proving node in the network, a first value based on a predetermined public system parameter and a random value selected by the each proving node and sending the calculated first value to its parent node directly
10 connected to the each proving node, wherein a product of the first values calculated by the plurality of the proving nodes in the network is to be used to calculate a first verification value for performing collective authentication of the plurality of proving nodes;

calculating, by the each proving node in the network, a second value based
15 on a random number generated for the each proving node, the random value selected by the each proving node, a private key of the each proving node, and the predetermined public system parameter, and sending the calculated second value to its parent node directly connected to the each proving node, wherein a product of the second values calculated by the plurality of the proving nodes in the
20 network is to be used to calculate a second verification value for performing collective authentication of the plurality of proving nodes .

20. The method according to claim 19, further comprising: if one of the proving nodes in the network is directly connected to at least one child node,

5 sending, by the proving node in the network, a first unit value to its parent node, wherein the first unit value is calculated based on a product of the first values calculated by both the proving node and the at least one child node of the proving node; and

 sending, by the proving node in the network, a second unit value to its parent node, wherein the second unit value is calculated based on a product of the second values calculated by both the proving node and the at least one child
10 node of the proving node.

21. The method according to claim 19 or claim 20, further comprising:

 receiving, by the each proving node in the network, a root random value from the verifying node, and

 generating, by the each proving node, the random number for calculating the
15 second value for the each proving node based on the received root random value and a pre-determined function.