US 20230254122A1

(19) **United States**

(12) **Patent Application Publication**

Islamov

(10) **Pub. No.: US 2023/0254122 A1**

(43) **Pub. Date: Aug. 10, 2023**

(54) **SECRET MATERIAL EXCHANGE AND AUTHENTICATION CRYPTOGRAPHY OPERATIONS**

(71) Applicant: **Winkk, Inc.**, Menlo Park, CA (US)

(72) Inventor: **Rustam Islamov**, Fremont, CA (US)

(21) Appl. No.: **18/136,187**

(22) Filed: **Apr. 18, 2023**

Related U.S. Application Data

(62) Division of application No. 17/040,949, filed on Sep. 23, 2020, now Pat. No. 11,637,694, filed as application No. PCT/US2019/041871 on Jul. 15, 2019.

(60) Provisional application No. 62/698,644, filed on Jul. 16, 2018.

(30) **Foreign Application Priority Data**

Jul. 15, 2019 (WO) PCT/US2019/041871

Publication Classification

(51) **Int. Cl.**

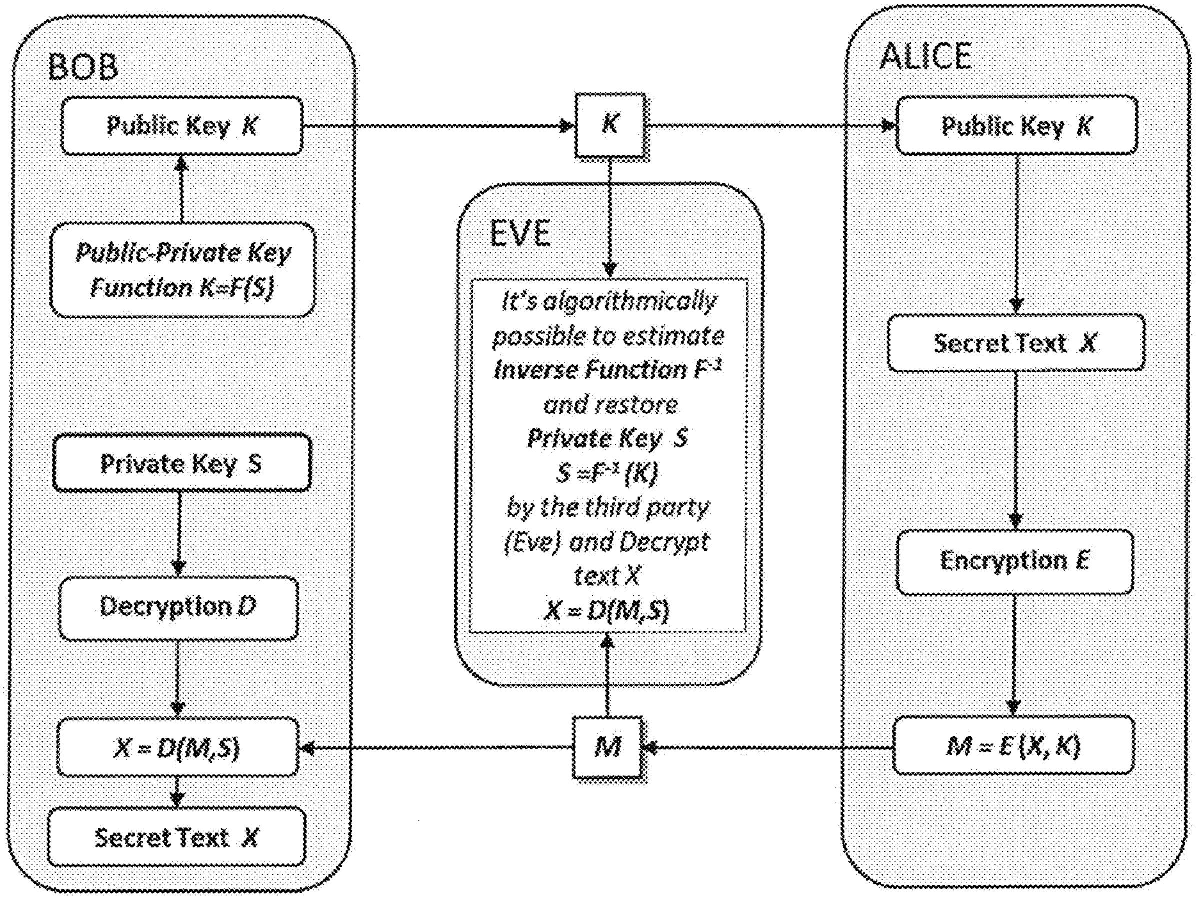
H04L 9/08	(2006.01)
H04L 9/14	(2006.01)
H04L 9/30	(2006.01)

(52) **U.S. Cl.**

CPC ***H04L 9/0819*** (2013.01); ***H04L 9/085*** (2013.01); ***H04L 9/0869*** (2013.01); ***H04L 9/14*** (2013.01); ***H04L 9/3093*** (2013.01)

(57) **ABSTRACT**

Aspects of associative cryptography key operations are described. In one embodiment, a first cryptographic function is applied to secret data to produce a first encrypted result. The first encrypted result is transmitted by a first device to a second device. The second device applies a second cryptographic function to the first encrypted result to produce a second encrypted result. At this point, the secret data has been encrypted by two different cryptographic functions, each of them being sufficient to secure the secret data from others. The two different cryptographic function can be inversed or removed, in any order, to reveal the secret data. Thus, the first device can apply a first inverse cryptographic function to the second encrypted result to produce a first result, and the second device can apply a second inverse cryptographic function to the first result to decrypt the secret data.


BOB
Public Key K
Public-Private Key Function K=F(S)
Private Key S
Decryption D
X = D(M,S)
Secret Text X
K
EVE
It's algorithmically possible to estimate Inverse Function F⁻¹ and restore Private Key S S =F⁻¹(K) by the third party (Eve) and Decrypt text X X = D(M,S)
M
ALICE
Public Key K
Secret Text X
Encryption E
M = E(X, K)

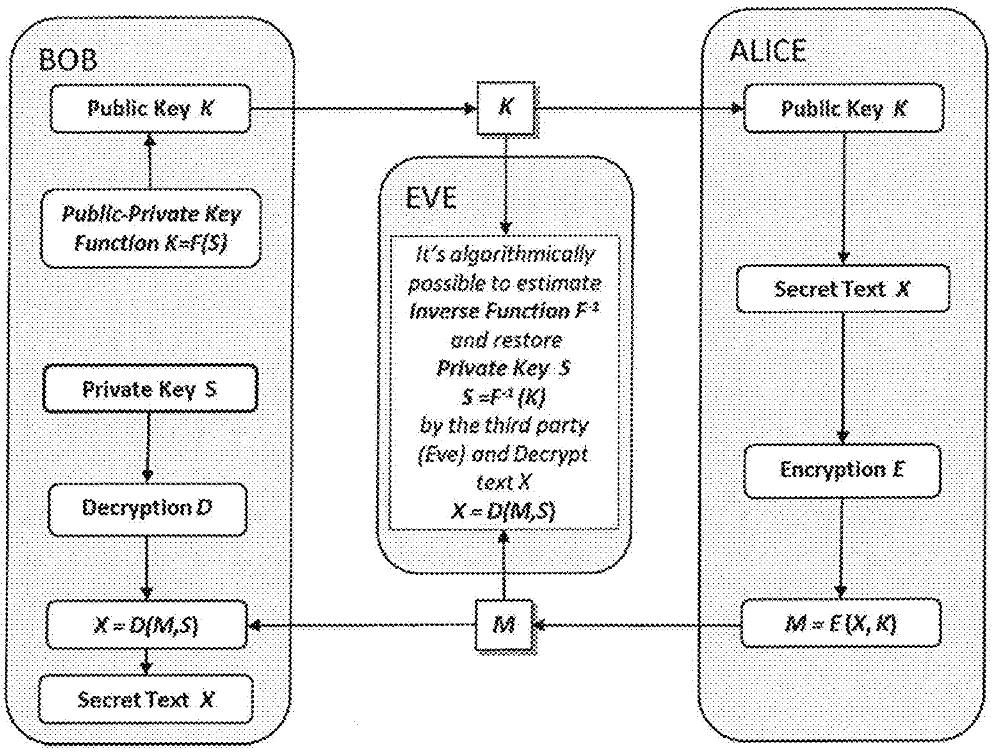

BOB
Public Key K
Public-Private Key Function $K=F(S)$
Private Key S
Decryption D
$X = D(M,S)$
Secret Text X
K
EVE
It's algorithmically possible to estimate Inverse Function F^{-1} and restore Private Key S $S = F^{-1}(K)$ by the third party (Eve) and Decrypt text X $X = D(M,S)$
M
ALICE
Public Key K
Secret Text X
Encryption E
$M = E(X, K)$

FIG. 1

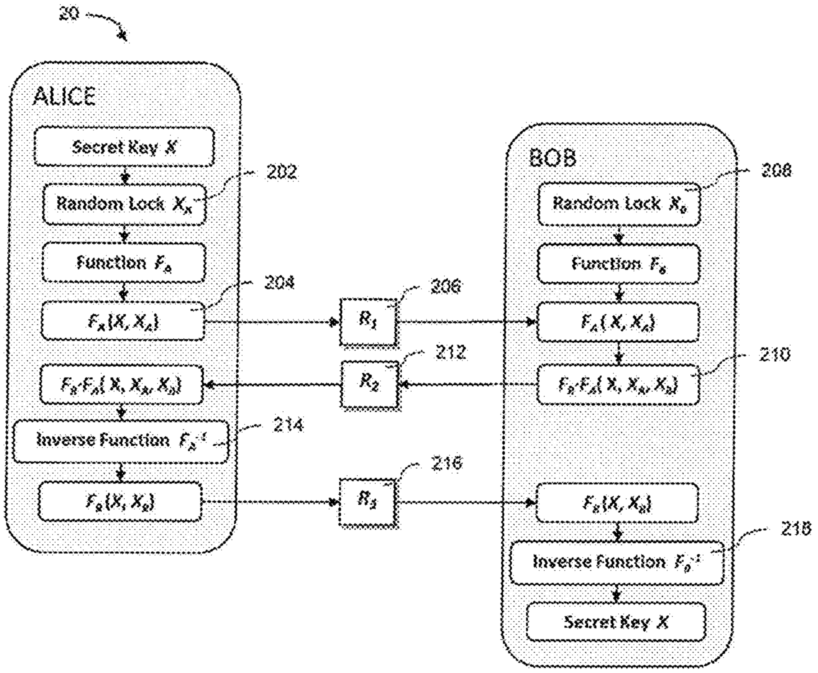

20
ALICE
Secret Key X
Random Lock X_X
202
Function F_A
$F_X(X, X_A)$
204
R_1
206
BOB
Random Lock X_B
208
Function F_B
$F_A(X, X_A)$
$F_B \cdot F_A(X, X_A, X_B)$
210
R_2
212
$F_B \cdot F_A(X, X_A, X_B)$
Inverse Function F_A^{-1}
214
$F_B(X, X_B)$
R_3
216
$F_B(X, X_B)$
Inverse Function F_B^{-1}
218
Secret Key X

FIG. 2

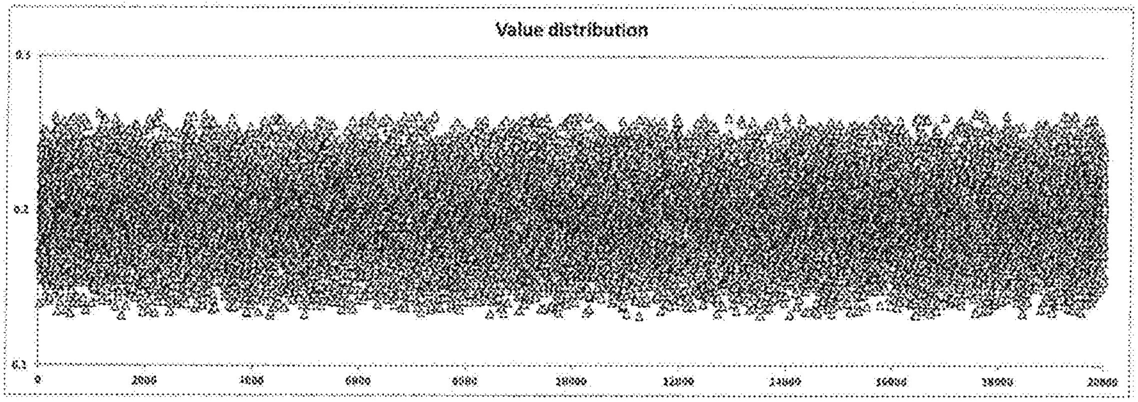

Value distribution
0.3
0.2
0.1
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000

FIG. 3A

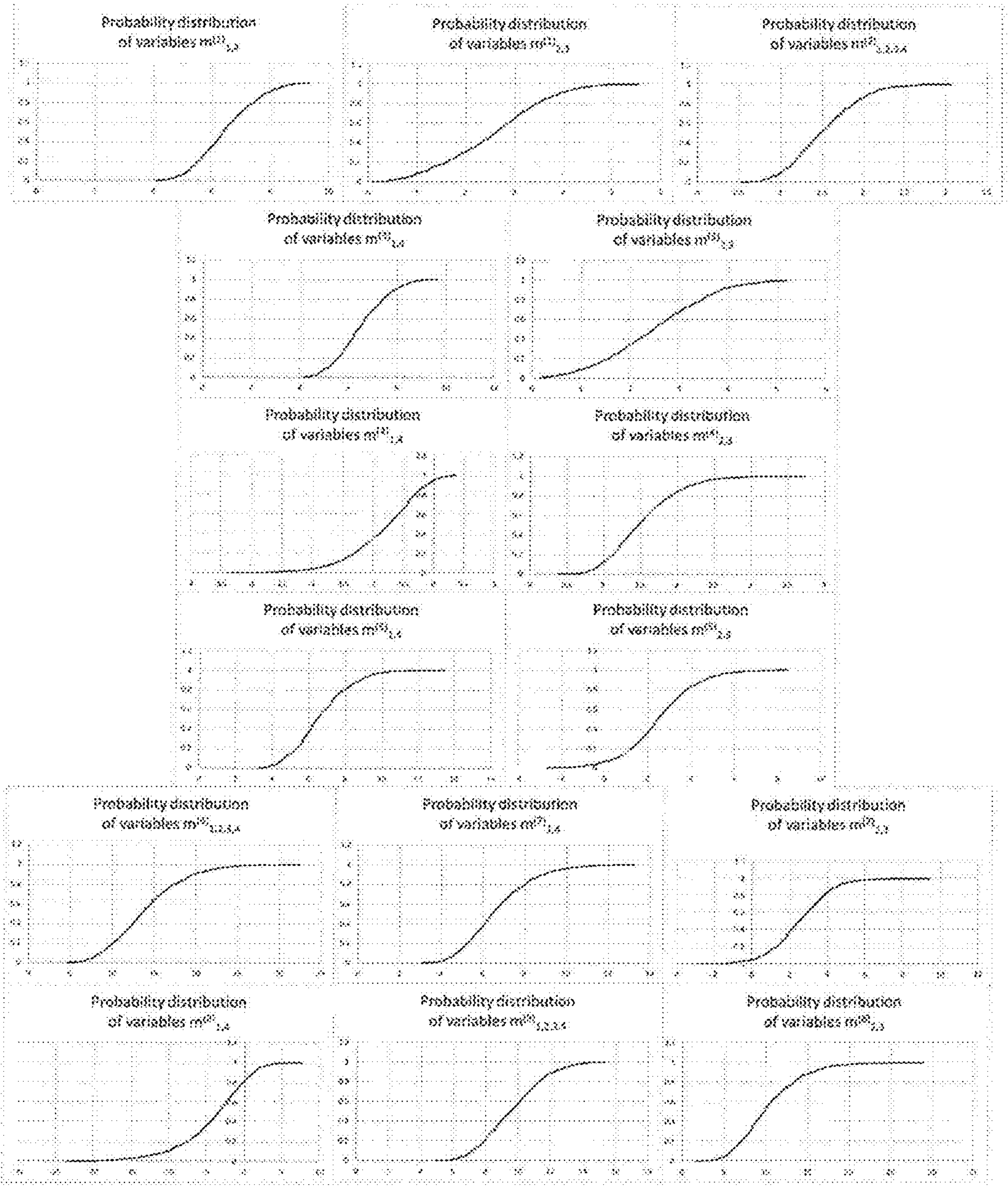
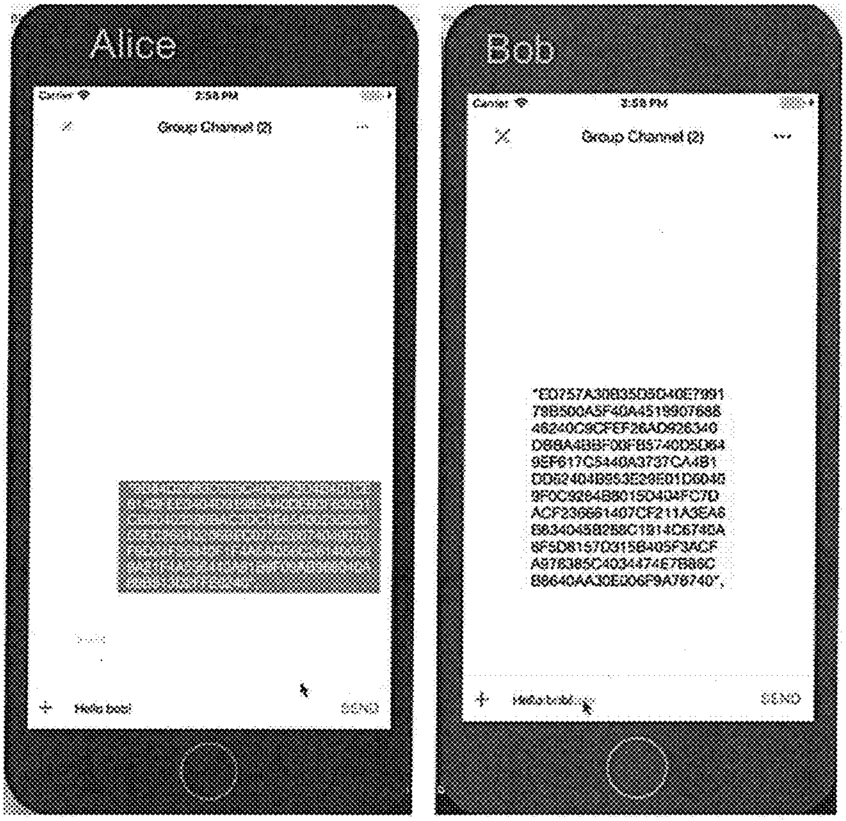

Probability distribution of variables $m^{(1)}_{1,4}$
Probability distribution of variables $m^{(1)}_{2,3}$
Probability distribution of variables $m^{(2)}_{1,2,3,4}$
Probability distribution of variables $m^{(3)}_{1,4}$
Probability distribution of variables $m^{(3)}_{2,3}$
Probability distribution of variables $m^{(4)}_{1,4}$
Probability distribution of variables $m^{(4)}_{2,3}$
Probability distribution of variables $m^{(5)}_{1,4}$
Probability distribution of variables $m^{(5)}_{2,3}$
Probability distribution of variables $m^{(6)}_{1,2,3,4}$
Probability distribution of variables $m^{(7)}_{1,4}$
Probability distribution of variables $m^{(7)}_{2,3}$
Probability distribution of variables $m^{(8)}_{1,4}$
Probability distribution of variables $m^{(8)}_{1,2,3,4}$
Probability distribution of variables $m^{(8)}_{2,3}$

FIG. 3B



"E0757A30B35D5C40E7991
79B500A5F40A4519907688
46240C9CFEF26AD926340
D88A4B8F00FB5740D5D84
9EF617C5440A3737CA4B1
DD5240B953E28E01D6040
9F0C9264B8015D404FC7D
ACF23B661407CF211A3EA6
B834045B258C1814C6740A
6F5D8157D315B405F3ACF
A978385C4034474E7B86C
B8640AA30E006F9A78740",

FIG. 4

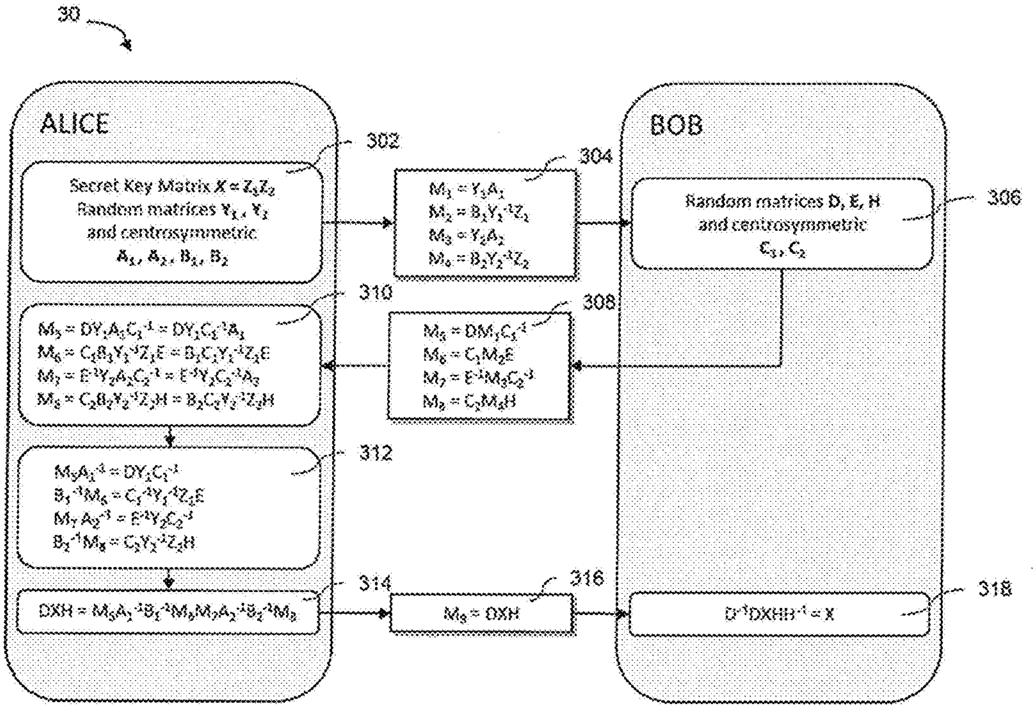

30
ALICE
BOB
302
Secret Key Matrix $X = Z_1Z_2$
Random matrices Y_1, Y_2
and centrosymmetric
A_1, A_2, B_1, B_2
304
$M_1 = Y_1A_1$
$M_2 = B_1Y_1^{-1}Z_1$
$M_3 = Y_2A_2$
$M_4 = B_2Y_2^{-1}Z_2$
306
Random matrices D, E, H
and centrosymmetric
C_1, C_2
308
$M_5 = DM_1C_1^{-1}$
$M_6 = C_1M_2E$
$M_7 = E^{-1}M_3C_2^{-1}$
$M_8 = C_2M_4H$
310
$M_5 = DY_1A_1C_1^{-1} = DY_1C_1^{-1}A_1$
$M_6 = C_1B_1Y_1^{-1}Z_1E = B_1C_1Y_1^{-1}Z_1E$
$M_7 = E^{-1}Y_2A_2C_2^{-1} = E^{-1}Y_2C_2^{-1}A_2$
$M_8 = C_2B_2Y_2^{-1}Z_2H = B_2C_2Y_2^{-1}Z_2H$
312
$M_5A_1^{-1} = DY_1C_1^{-1}$
$B_1^{-1}M_6 = C_1^{-1}Y_1^{-1}Z_1E$
$M_7A_2^{-1} = E^{-1}Y_2C_2^{-1}$
$B_2^{-1}M_8 = C_2Y_2^{-1}Z_2H$
314
$DXH = M_5A_1^{-1}B_1^{-1}M_6M_7A_2^{-1}B_2^{-1}M_8$
316
$M_9 = DXH$
318
$D^{-1}DXHH^{-1} = X$

FIG. 5

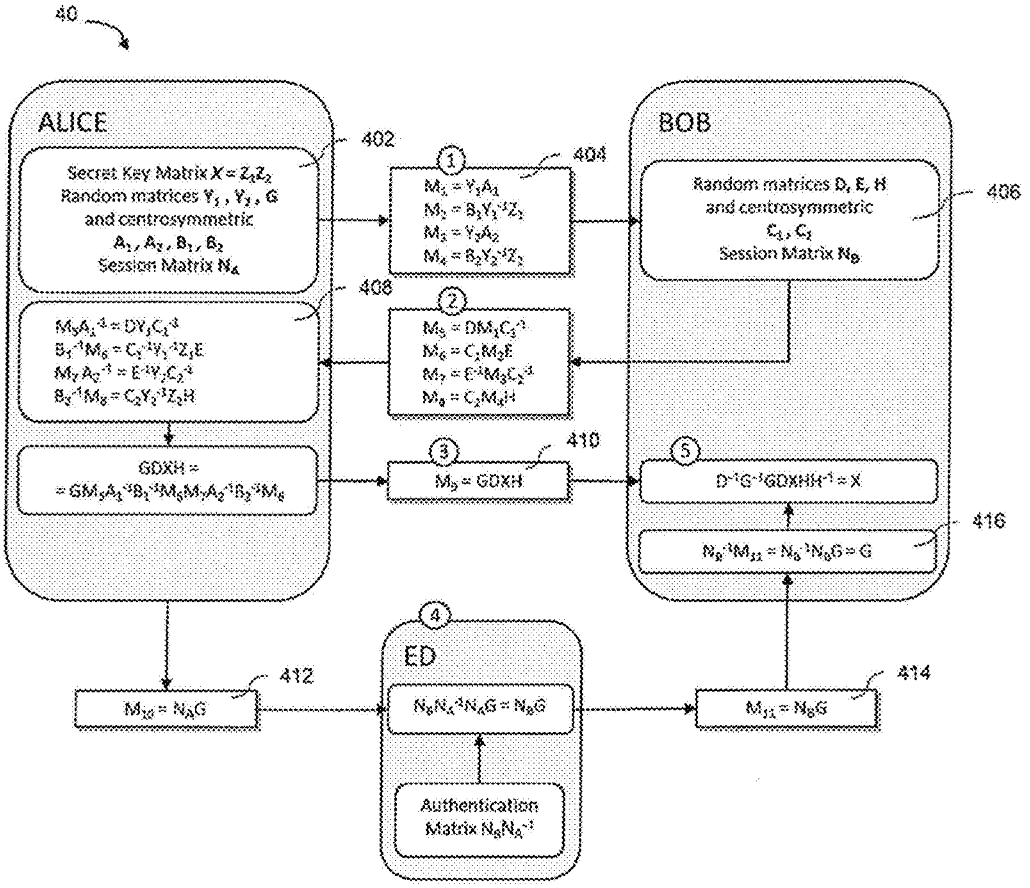

40
ALICE
402
Secret Key Matrix $X = Z_1Z_2$
Random matrices Y_1, Y_2, G
and centrosymmetric
A_1, A_2, B_1, B_2
Session Matrix N_A
1
404
$M_1 = Y_1A_1$
$M_2 = B_1Y_1^{-1}Z_1$
$M_3 = Y_2A_2$
$M_4 = B_2Y_2^{-1}Z_2$
BOB
406
Random matrices D, E, H
and centrosymmetric
C_1, C_2
Session Matrix N_B
408
$M_5A_1^{-1} = DY_1C_1^{-1}$
$B_1^{-1}M_6 = C_1^{-1}Y_1^{-1}Z_1E$
$M_7A_2^{-1} = E^{-1}Y_2C_2^{-1}$
$B_2^{-1}M_8 = C_2Y_2^{-1}Z_2H$
2
$M_5 = DM_1C_1^{-1}$
$M_6 = C_1M_2E$
$M_7 = E^{-1}M_3C_2^{-1}$
$M_8 = C_2M_4H$
$GDXH =$
$= GM_5A_1^{-1}B_1^{-1}M_6M_7A_2^{-1}B_2^{-1}M_8$
3
410
$M_9 = GDXH$
5
$D^{-1}G^{-1}GDXHH^{-1} = X$
416
$N_B^{-1}M_{11} = N_B^{-1}N_BG = G$
4
ED
412
$M_{10} = N_AG$
$N_BN_A^{-1}N_AG = N_BG$
Authentication Matrix $N_BN_A^{-1}$
414
$M_{11} = N_BG$

FIG. 6

SECRET MATERIAL EXCHANGE AND AUTHENTICATION CRYPTOGRAPHY OPERATIONS

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims priority to PCT Application No. PCT/US2019/041871, filed on Jul. 15, 2019, and titled "SECRET MATERIAL EXCHANGE AND AUTHENTICATION CRYPTOGRAPHY OPERATIONS," which is hereby incorporated by reference in its entirety for all purposes.

BACKGROUND

[0002] Cryptography is related to the study of protocols, techniques, and approaches that prevent third parties from accessing, reading, and/or interpreting secret data. Cryptography can be applied to various processes in information security, such as data integrity and encryption, confidentiality, authentication, verification, and non-repudiation. Thus, cryptography has several applications in various fields, including data encryption and privacy, computer network communications and transaction processing, and computing system security and integrity.

[0003] Modern cryptography often relies upon computational hardness in mathematical theory. In other words, it might be theoretically possible to break certain cryptographic systems, but the time required to do so makes such cryptographic-defeating processes intractable. Typically, computationally-secure cryptography processes are preferable to those which are easier to defeat. At the same time, however, computationally-secure cryptography processes might be more computationally-intensive to implement and, thus, more time consuming and costly. In that context, although some cryptographic processes, such as a one time pad, cannot be broken or defeated even with unlimited computing power, those schemes are more difficult to implement than a good, theoretically-breakable but computationally secure approach. As such, modern computing devices may exchange secret data using cryptographic processes having security problems (e.g., the processes are susceptible to brute force attack). At the same time, those cryptographic processes may be resource intensive (e.g., the processes are computationally-intensive to implement).

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Many aspects of the present disclosure can be better understood with reference to the following drawings. The components in the drawings are not necessarily to scale, with emphasis instead being placed upon clearly illustrating the principles of the disclosure. Moreover, in the drawings, like reference numerals designate corresponding parts throughout the several views.

[0005] FIG. **1** illustrates a process of secret text transfer using asymmetric keys.

[0006] FIG. **2** illustrates a representative process of secret key transfer using cryptography processes according to various embodiments described herein.

[0007] FIG. **3**A illustrates an example distribution function of variables resulting from the white noise associative cryptography key operations according to various embodiments described herein.

[0008] FIG. **3**B illustrates example probability distribution functions of variables resulting from the white noise associative cryptography key operations according to various embodiments described herein.

[0009] FIG. **4** illustrates example user interfaces of a program to perform cryptography key operations according to various embodiments described herein.

[0010] FIG. **5** illustrates a more particular example of a secret key transfer process according to the concepts described herein.

[0011] FIG. **6** illustrates an example of a secret key transfer process using authentication according to the concepts described herein.

DETAILED DESCRIPTION

[0012] As noted above, cryptography is related to the study of protocols, techniques, and approaches that prevent third parties from accessing, reading, and/or interpreting secret data. In the context of cryptography, the Rivest-Shamir-Adleman (RSA) cryptosystem, elliptic curve cryptography (ECC) cryptosystem, and other asymmetrical (and symmetrical) methods of secure key exchange have security problems. Those cryptosystems are based on complexity and can, theoretically, be decrypted.

[0013] In contrast to the RSA, ECC, and other cryptosystems, the cryptographic processes described herein is more immune to cryptanalysis and permits the sharing of secret data, such as symmetric keys and other secret data, over public networks. The cryptographic system can also be used for authentication. No known methods of traditional or quantum computing can be used to circumvent the cryptographic approaches described herein. The cryptographic system described herein was developed to achieve a number of goals including (1) securely exchanging cryptographic keys over public networks, (2) information ciphering, authentication, and (4) encryption for public networks that is secure against standard and quantum computing.

[0014] In the context described herein, white noise can be defined as (or can include) a sequence of independent random variables (e.g., discrete numbers) with a uniform probability distribution. Polynomial white noise can be defined as (or can include) a sequence of polynomial function values composed by independent random variables (e.g., discrete numbers) with a uniform probability distribution.

[0015] No known algorithm can decrypt the operations described herein due, at least in part, to the use of white noise randomization. The unknown independent variables appear to third parties as random white noise and, thus, there is no correlation between those variables and any information being transferred. As one example, the key exchange method or process described herein can be shown as an exchange of matrices with a corresponding number of different unknown independent variables and visible values. The number of unknown independent variables always exceeds the number of visible independent values in any combination of subsets of matrices. Further, the number of unknown variables exceeds the number of publically visible polynomial functions. Additionally, no inverse polynomial functions can be determined without information about the secret key—even if the plain text of the secret key is known by a third party.

[0016] Turning to the drawings, FIG. **1** illustrates a process of secret text transfer using asymmetric keys. In the example shown in FIG. **1**, Alice wishes to communicate secret text to Bob over a public network, such as the Internet, and Eve is the eavesdropper. To communicate the secret text, which can be a symmetric key or any other secret information, Alice and Bob use asymmetric cryptography. Asymmetric cryptography relies upon a key pair including a public key that can be disseminated to third parties (e.g., Alice) and a private key which is kept private (e.g., by Bob). In an asymmetric cryptography system, any person can encrypt a message using the public key, and that encrypted message can only be decrypted using the private key. The strength of asymmetric cryptography relies on the degree of difficulty (e.g., computational impracticality) for a private key to be determined from its associated public key. Asymmetric cryptography also depends on keeping the private key private.

[0017] Referring back to FIG. **1**, Alice obtains a copy of a public key from Bob (or any other source). Alice encrypts the secret text using the public key to produce the encrypted secret text and communicates it to Bob over the public network. Bob then decrypts the encrypted secret text using the private key to obtain the secret key. Over the public network, Eve can only see the encrypted secret text. Even if Eve obtains a copy of the encrypted secret text and the public key used to create it, Eve cannot obtain the secret text from the encrypted secret text using the public key. Instead, only the private key, which is securely held and protected by Bob, can be used to decrypt the encrypted secret text to obtain the secret text from Alice.

[0018] There are drawbacks and limitations to using asymmetric cryptography. For example, it is algorithmically possible to estimate (or determine) the private key in a key pair from the publicly available public key. Additionally, asymmetric key pairs are relatively difficult and time consuming to create, typically depending upon the identification of large prime numbers. Further, asymmetric cryptography can be vulnerable in that it may produce the same predictable encrypted output when the same secret text is encrypted.

[0019] To be distinguished from other cryptographic systems, various cryptography processes or operations are described herein. In one embodiment, a first cryptographic function is applied to secret data. The first cryptographic function operates as a type of cryptographic key and encrypts or ciphers the secret data to produce a first encrypted result. The first encrypted result can be securely transmitted by a first device to a second device. The second device then applies a second cryptographic function to the first encrypted result. Similar to the first cryptographic function, the second cryptographic function operates as a cryptographic key and further (or doubly) encrypts or ciphers the first encrypted result to produce a second (or doubly) encrypted result. At this point, the secret data has been encrypted by two different cryptographic functions, each of them being sufficient to secure the secret data. The two different cryptographic functions can then be inversed or removed, in any order, to reveal the secret data.

[0020] Turning to the embodiments, FIG. **2** illustrates a representative process **20** of secret key transfer using cryptography processes according to various embodiments described herein. The process described below can be performed by any suitable computing device(s) including a processor and memory, without limitation. In the example shown in FIG. **2**, Alice wants to securely pass the secret key Xto Bob over a public network. To do so, Alice should first encrypt the secret key X before sending it to Bob.

[0021] To encrypt the secret key X, Alice holds a first cryptographic function F_A. In various embodiments, the cryptographic function F_A can be embodied as any suitable mathematical function having an inverse which cannot be determined without knowledge of a certain set of parameters of the mathematical function. In one embodiment, the function F_A can be embodied as a polynomial function or multivariate polynomial function defined in part by one or more variables, combinations of variables, combinations of variables at various powers, and coefficients. To undo or unlock (e.g., decrypt) the effect of the cryptographic function F_A, Alice also holds a first inverse cryptographic function $F^{-1}A$.

[0022] To start, at step **202**, the process **20** includes Alice generating, with a first computing device, a first random lock X_A. The first random lock X_A can be embodied as an array or vector of random scalar integers, for example, or another suitable organized structure of random numbers. In the process **20**, the first random lock X_A can operate as a type of initialization vector upon which the cryptographic function F_A is applied in combination with the secret key X. For example, the first random lock X_A helps to randomize the application of the cryptographic function F_A creating, in effect, a new random cryptographic function F_A for each different random lock X_A. In that context, the first random lock X_4 helps to achieve semantic security, so that repeated usage of the cryptographic function F_A with the same operand does not produce the same ciphered result and does not allow an attacker to infer any information.

[0023] At step **204**, the process **20** includes Alice applying, with the first computing device, the first cryptographic function F_A to a combination of the secret key X and the first random lock X_A to produce a first encrypted result R_1. Here, Alice's secret key X, which can include letters, numbers, American Standard Code for Information Interchange (ASCII) characters, etc., is ciphered with random numbers (i.e., the first random lock X_A) using the cryptographic operation or function F_A. The cryptographic function F_A can be embodied as any suitable mathematical function, such as a polynomial or multivariate polynomial function. For example, the cryptographic function F_A can be embodied as a polynomial function $F(CX^k)$ of kth order written as:

$$F(CX^k) = \sum_{i_1=1}^{k} \dots \sum_{i_2=1}^{k} C\textcircled{?} X_{i_1} X_{i_2} \dots X_{i_2}, \tag{1}$$

$\textcircled{?}$ indicates text missing or illegible when filed

where $C_{i \ldots k}$ are coefficients of the polynomial function $F(CX^k)$, and $X_{i \ldots k}$ are combinations of the operand X, which can include a combination of a random lock and secret data.

[0024] Thus, at step **204**, Alice's secret key X, which may include letters, numbers, American Standard Code for Information Interchange (ASCII) characters, etc., are ciphered with random numbers based on the first random lock X_A and the first cryptographic function F_A. As an example, a distribution function of the variables in the results R_1, R_2, and R_3

is shown in FIG. **3**A, and probability distribution functions of the variables in the results R_1, R_2, and R_3 is shown in FIG. **3**B.

[0025] The structure of the polynomial function $F(CX^k)$ and the coefficients can be known to others (although they generally are not) from the formalization of the algorithm. However, even if the structure of the polynomial function F and values of the coefficients C, k are known to a third party, the third party still cannot decrypt the transferred information.

[0026] At step **206**, the process **20** includes Alice transmitting, with the first computing device, the first encrypted result R_1 to Bob's second computing device. At step **208**, the process **20** includes Bob generating, with the second computing device, a second random lock X_B. Similar to the first random lock X_A, the second random lock X_B can be embodied as an array or vector of random scalar integers, for example, or another suitable organized structure of random numbers. In the process **20**, the second random lock X_S can also operate as a type of initialization vector for the cryptographic function F_B. For example, the second random lock X_B helps to randomize the application of Bob's cryptographic function F_B creating, in effect, a new random cryptographic function F_B for each different random lock $_{XB}$. In that context, the second random lock X_B helps to achieve semantic security, so that repeated usage of the cryptographic function F_B with the same operand does not produce the same ciphered result and does not allow an attacker to infer any information.

[0027] At step **210**, the process includes Bob applying, with the second computing device, Bob's cryptographic function F_B to a combination of the first encrypted result R_1 and the second random lock X_B to produce a second encrypted result R_2. Here, the first encrypted result R_1 (e.g., $F_A(X,X_A)$) is (doubly) ciphered with random numbers (i.e., the second random lock X_B) using the cryptographic operation or function F_B. The cryptographic function F_B can be embodied as any suitable mathematical function, such as a polynomial or multivariate polynomial function. For example, the cryptographic function F_B can be embodied as a polynomial function $F(CX^k)$ of kth order according to that shown above in Equation (1).

[0028] At this point, Alice's secret key X has been encrypted or ciphered by two different cryptographic functions F_A and F_B, each of them being sufficient to secure the secret key X from others. The two different cryptographic functions can then be inversed or removed, in any order, to reveal the secret key X. In other words, to decrypt the secret key X from the second encrypted result R_2 (i.e., to undo the effects of the cryptographic functions F_A and Fa) it is possible to either apply the inverse $F^{-1}{}_A$ function to F_A or the inverse $F^{-1}{}_B$ function to F_B first. Thus, according to one aspect of associative cryptography key operations described herein, the order in which the second encrypted result R_2 is applied to the inverse cryptographic functions $F^{-1}{}_A$ and $F^{-1}{}_B$. does not impact the results of the decryption of secret key X from the second encrypted result R_2. Further, any number of cryptographic functions to $F_1 \ldots F_N$ can be applied to encrypt secret data in any order to produce an encrypted result R_N, and that encrypted result R_N can be decrypted in any order using the inverse cryptographic functions $F^{-1}{}_1 \ldots F^{-1}{}_N$.

[0029] At step **212**, the process **20** includes Bob transmitting, with the second computing device, the second encrypted result R_2 to the first computing device. At step **214**, the process **20** includes Alice applying, with the first computing device, the first inverse cryptographic function $F^{-1}{}_A$ to the second encrypted result R_2 to produce the result R_3. The first inverse cryptographic function $F^{-1}{}_A$ unlocks or removes the effect of both the first random lock X_A and the first cryptographic function F_A. Thus, the result R_3 is what remains of the second encrypted result R_2 after the effect of the first random lock X_A and the first cryptographic function F_A are undone or unlocked (e.g., $F_B(X,X_B)$). Thus the result R_3 is still encrypted, but only by Bob's second random lock X_B and the second cryptographic function F_B, and the result R_3 can be securely transmitted over the public network.

[0030] At step **216**, the process **20** includes Alice transmitting, with the first computing device, the result R_3 to the second computing device. Finally, at step **218**, the process **20** includes Bob applying, with the second computing device, the second inverse cryptographic function $F^{-1}{}_B$ to the result R_3 to arrive at the secret key X.

[0031] At the end of the process **20**, the secret key X has been securely communicated from Alice to Bob. In contrast to the asymmetric key process described above with reference to FIG. **1**, key pairs are not used in the process **20**.

[0032] The general idea embodied in the process **20** is based on certain features of the publically unknown vectors X and the publically available (potentially visible) vectors R. Particularly, the number of variables "n" of the vectors X $\{x_1, \ldots, x_n\}$ is always more than the number of variables "m" of the vectors $R=\{r_1, \ldots, r_m\}$, i.e., n>m. Thus, there are no known algorithms which give a definite decryption solution of the secret key X, based only on visible values of the vectors R in the public networks. From this point of view, the method is cryptanalysis resistant. To obtain the only solution $x_1, \ldots, x_n$ from the values $r_1, \ldots, r_m$ of the polynomial functions F_A and F_B, the third party (e.g., outsider Eve) should have additional information about the structure of the random vectors X_A and X_B, which are available for Alice and Bob only. For instance, from $x_1+x_2+x_3=r_1$, it is not possible for a third party to arrive at a single solution for x_1 with only the value of the variable r_1 being publically visible, because the additional information about the values of the variables x_2+x_3 are not known.

[0033] A comparison of the features of asymmetrical methods and the method described herein is give in Table 1 below.

TABLE 1

Features	Public-Private Key Asymmetrical (RSA, ECC)	PWN Three Pass Method
Numbers	Prime Numbers	Any Random Numbers
Time to Develop New Key	Relatively More Costly	Negligible
Processing Time	Relatively More Costly	Negligible
Inverse Function From Public Key	Relatively Complex	Inverse Function Does Not Exist
Third Party Defeat	Possible	Never
Public Network Output For Constant Input	Constant, predictable	Random, unpredictable

[0034] An example of the use of the method described herein is provided below. Using the method, plain text (as a letter or ASCII code of 256 numbers) is represented in ciphered text by three corresponding random numbers r_1, r_2

and r_3 which are calculated by a random generator. Table 2 shows an example of how the plain text "This is a plain text" appears in ciphered numbers.

TABLE 2

Plain text	Ciphered text		
text	r_1	r_2	r_3
T	0.001251	0.563585	0.003585
h	0.193304	0.808741	0.158307
i	0.585009	0.479873	0.28051
s	0.350291	0.895962	0.313555
	0.82284	0.746605	0.614412
i	0.174108	0.858943	0.151801
s	0.710501	0.513535	0.363394
	0.303995	0.014985	0.006167
a	0.091403	0.364452	0.035009
	0.147313	0.165899	0.02575
p	0.988525	0.445692	0.438709
1	0.119083	0.004669	0.001204 i
a	0.00891 1	0.37788	0.005292
i	0.531663	0.571184	0.303183
n	0.601764	0.607166	0.363988
	0.166234	0.663045	0.113037
t	0.450789	0.352123	0.159469
e	0.057039	0.607685	0.037377
x	0.783319	0.802606	0.623152
t	0.519883	0.30195	0.157851

[0035] Uniform distribution is called "white noise" due to its informative features. For the letter 'A' (ASCII code 65), as one example, the random numbers may appear over the public net as $r_1=0.001251$, $r_2=0.563585$, $r_3=0.560746$ or $r_1=0.585009$, $r_2=0.479873$, $r_3=0.105796$ and every time the random variables r_1, r_2, r_3 will be unpredictable. The correlation function between any two variables x and y is estimated as follows:

$$corr(x, y) = \frac{\sum(x-\overline{x})(y-\overline{y})}{\sqrt{\sum(x-\overline{x})^2\sum(y-\overline{y})^2}}. \tag{2}$$

The results of correlation function evaluation for pairs (r_1, r_2), (r_2, r_3) and (r_1, r_3) are given in Table 3 below.

TABLE 3

corr (r_1, r_2)	corr (r_2, r_3)	corr (r_1, r_3)
−0.013927	−0.002873	−0.010771

[0036] The correlation is negligibly small, which means that ciphered information is encapsulated into white noise and is not analyzable by a third party. There are no known algorithms to decrypt the ciphered information without the encryption key.

[0037] In the approaches described herein, there are neither restrictions nor requirements on the encryption key number and length. All keys are equal in terms of crypt analysis resistance. Additionally, there are no correlations between the plain text and the ciphered random numbers (r_1, r_2, r_3), as the combinations of them are unpredictable. There are no known algorithms which can decrypt ciphered random numbers (r_1, r_2, r_3) into plain text without the key. There are no known algorithms which can recalculate the encryption key using visible ciphered random numbers (r_1, r_2, r_3) and visible plain text. There is no need for rotation of encryption keys if a physical, completely unpredictable random number generator is used. The series repetition period of real random numbers (r_1, r_2, r_3) is infinite.

[0038] Computational time needed to encrypt and decrypt data by the method described herein is significantly smaller than commonly used algorithms. Since the method uses polynomial functions, the transaction of numbers (or ASCII) should be controlled by calculation procedures. The analysis of 25,600,000 transactions demonstrates that the final error of the secret key value estimate does not exceed 0.001%. This means that, for example, the transaction of the letter 'A,' which is represented by the integer number 65 (ASCII), after all transformations from client to server could be calculated to be a number about 64.9999 (and depends in part on the random generator variables during the transaction).

[0039] A comparison of the features of a standard symmetrical method and the method described herein are given in Table 4 below.

TABLE 4

Features	Symmetrical FIPS Pub 197	WNT One Pass Transaction (in combination with Three Pass Transaction)
Encryption Key Rotation	Must Have	Not Needed
Processing time	Costly	Negligible
Security resistance and key length	Strong relation	No Relation
Hack	Costly	Never (Potentially Impossible)
Public net output for constant input (without key rotation)	Constant, Predictable	Random, Unpredictable

[0040] A computer program was developed to implement the method described herein. As shown in FIG. **4**, Alice securely sends her secret text "Hello bob" to Bob using the three pass transaction. In FIG. **4**, random values appear to a third party during the three pass transaction (specially shown in the blue box).

[0041] Among other benefits, the processes described herein can be used to achieve unbreakable (or nearly unbreakable) encryption over wireless, wired, and public networks, and against quantum computing attacks. It requires relatively little processing power for encrypting and decrypting and, thus, can be used for rapid verification and transactions. A practically limitless number of new keys can be generated on the fly. Thus, the keys can be changed on every transaction. Encryption and decryption can also occur on individual devices due to the high speed of encryption and low processing requirements. Further, there is no single point of compromise because every individual party has their own key. If a key is compromised, it is the one compromised and can be renewed or replaced.

[0042] An outline of various problems encountered and solutions that can be provided by the cryptographic systems described herein are given in Table 5 below.

TABLE 5

Problem	Solution
Establishing a secure and reliable ID for all transactions	Digital ID system in the cloud for processing Ids ID system only used for registration and verification Information unhackable
Having a secure payment system that eliminates fraud	Payment system using ID Email, internet banking, wireless transaction
Cryptocurrency that is secure and stable	Absolutely secure, stable, and based on verifiable IDs
Fast enough and secure trading system for cryptocurrencies	Rapid trading and verification Trading exchanges connected to Exchange
Mobile Payments	Integrity over wireless signals and public net Transactions cannot be defrauded via screening or copying
Key Management System	Cloud key management service ID system to outsource all key management responsibilities
People forget passwords and passwords are a weak point in security	Pass eliminates the use of passwords using ID center

[0043] FIG. **5** illustrates a more particular example of a secret key transfer process **30** according to the concepts described herein. While an example using square matrices of a certain size is provided below, the concepts described herein can be extended to use with square matrices of any size. Further, although the example below is presented in certain cases as steps between "Alice" and "Bob," the process is conducted by computing systems or devices.

[0044] At the outset, consider the key to be exchanged, K, as a sequence of m bytes, each including one of the ASCII codes from 0 to 255, as follows:

$$K=\{k_1,k_2, \ldots ,k_m\},0\leq k_i\leq 255.$$

[0045] For example, the key string "ABCD" can be presented as ASCII codes K={65, 66, 67, 68}. A sequence of real numbers X can then be defined as a transformation of the key numbers (i.e., k_1, k_2, k_m), which are integers, into real ones, as follows:

$$X=\Phi(K),\Phi:N^m\rightarrow R^m \text{ and}$$

$$X=\{x_1,x_2, \ldots ,x_m\},x_i\in R.$$

[0046] The sequence of real numbers is put into set of second order square matrices, as follows:

$$X=\begin{vmatrix} x_1 & x_2 \\ x_3 & x_4 \end{vmatrix} \ldots \begin{vmatrix} x_{m-3} & x_{m-2} \\ x_{m-1} & x_m \end{vmatrix}$$

[0047] If the number of real key numbers is not multiple of four, the last matrix is not fully filled in. In this case, the rest of the matrix members can be generated and added as any random numbers without influencing the algorithm.

[0048] Now, assume that Alice wants to pass the secret key K to Bob. For simplicity, however, consider one square matrix X, as follows:

$$X=\begin{vmatrix} x_1 & x_2 \\ x_3 & x_4 \end{vmatrix}$$

[0049] The matrix X decomposes into two singular matrices Z_1 and Z_2

$$X=Z_1Z_2,$$

$$Z_1=\begin{vmatrix} Z_1 & Z_2 \\ Z_3 & \frac{Z_2Z_3}{Z_3} \end{vmatrix}, \text{ and}$$

$$Z_2=\begin{vmatrix} Z_4 & Z_5 \\ Z_6 & \frac{Z_3Z_5}{Z_6} \end{vmatrix}$$

[0050] At step **302**, the process includes forming the matrix X as a singular matrix using a number of the real key numbers of the secret key K based on the following relationship $x_4=x_2x_3/x_1$. In that case, the inverse of matrix X, or X^{-1}, does not exist (see properties of singular matrices and matrix determinants in APPENDIX). In that case, the matrix X represents a portion of the secret key K, $\{k_1, k_2, k_3\}$.

[0051] As part of a first pass transaction, at step **302**, the process further includes generating a uniformly distributed random matrices Y_1, Y_2 and inverse matrices Y_1^{-1}, Y_2^{-1}, as follows:

$$Y_1=\begin{vmatrix} y_1 & y_2 \\ y_3 & y_4 \end{vmatrix},$$

$$Y_1^{-1}=\frac{\begin{vmatrix} y_4 & -y_2 \\ -y_3 & y_1 \end{vmatrix}}{y_1y_4-y_2y_3}, y_i\in R, y_1y_4\neq y_2y_3,$$

$$Y_2=\begin{vmatrix} y_5 & y_6 \\ y_7 & y_8 \end{vmatrix}, \text{ and}$$

$$Y_2^{-1}=\frac{\begin{vmatrix} y_8 & -y_6 \\ -y_7 & y_5 \end{vmatrix}}{y_5y_8-y_6y_7}, y_i\in R, y_5y_8\neq y_6y_7.$$

[0052] At step **302**, the process also includes generating uniformly distributed random centrosymmetric A_1, A_2, B_1, B_2, and inverse A_1^{-1}, A_2^{-1}, B_1^{-1}, B_2^{-1} matrices as follows:

$$A_1=\begin{vmatrix} a_1 & a_2 \\ a_2 & a_1 \end{vmatrix},$$

$$A_2=\begin{vmatrix} a_3 & a_4 \\ a_4 & a_3 \end{vmatrix},$$

$$A_1^{-1}=\frac{\begin{vmatrix} a_1 & -a_2 \\ -a_2 & a_1 \end{vmatrix}}{a_1^2-a_2^2},$$

$$A_2^{-1}=\frac{\begin{vmatrix} a_3 & a_4 \\ a_4 & a_3 \end{vmatrix}}{a_3^2-a_4^2}, a_i\in R, a_1^2\neq a_2^2, a_3^2\neq a_4^2,$$

$$B_1=\begin{vmatrix} b_1 & b_2 \\ b_2 & b_1 \end{vmatrix},$$

$$B_2=\begin{vmatrix} b_3 & b_4 \\ b_4 & b_3 \end{vmatrix},$$

$$B_1^{-1}=\frac{\begin{vmatrix} b_1 & -b_2 \\ -b_2 & b_1 \end{vmatrix}}{b_1^2-b_2^2},$$

-continued

$$B_2^{-1}=\begin{vmatrix} b_3 & -b_4 \\ -b_4 & b_3 \end{vmatrix}, b_i \in R, b_1^2 \neq b_2^2, b_3^2 \neq b_4^2.$$

Centrosymmetric square matrices A and B are always of the form AB=BA.

[0053] At step **304**, the process includes Alice generating and sending matrices M_1 and M_2 to Bob, as follows:

$$M_1=\begin{vmatrix} m_1^{(1)} & m_2^{(1)} \\ m_3^{(1)} & m_4^{(1)} \end{vmatrix},$$

$$M_2=\begin{vmatrix} m_1^{(2)} & m_2^{(2)} \\ m_3^{(2)} & m_4^{(2)} \end{vmatrix},$$

$$M_3=\begin{vmatrix} m_1^{(3)} & m_2^{(3)} \\ m_3^{(3)} & m_4^{(3)} \end{vmatrix}, \text{ and}$$

$$M_4=\begin{vmatrix} m_1^{(4)} & m_2^{(4)} \\ m_3^{(4)} & m_4^{(4)} \end{vmatrix},$$

[0054] which are generated according to the following calculations:

$$M_1=Y_1A_1, \tag{3}$$

$$M_2=B_1Y_1^{-1}Z_1, \tag{4}$$

$$M_3=Y_2A_2, \text{ and} \tag{5}$$

$$M_4=B_2Y_2^{-1}Z_2, \tag{6}$$

[0055] Thus, at step **304**, Alice sends to Bob fourteen publicly visible values ($m_1^{(1)}$, $m_2^{(1)}$, $m_3^{(1)}$, $m_4^{(1)}$, $m_1^{(2)}$, $m_2^{(2)}$, $m_3^{(2)}$, $m_1^{(3)}$, $m_2^{(3)}$, $m_3^{(3)}$, $m_4^{(3)}$, $m_1^{(4)}$, $m_2^{(4)}$, $m_3^{(4)}$) of matrices M_1, M_2, M_3 and M_4 that are calculated from twenty-two independent unknown (for the third party) variables (a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, z_1, z_2, z_3, z_4, z_5, z_6) known by Alice only, as follows:

$$m_1^{(1)}=a_1y_1+a_2y_2,$$

$$m_2^{(1)}=a_2y_1+a_1y_2,$$

$$m_3^{(1)}=a_1y_3+a_2y_4,$$

$$m_4^{(1)}=a_2y_3+a_1y_4,$$

$$m_1^{(2)}=\frac{b_1(x_1y_4-x_3y_2)+b_2(x_3y_1-x_1y_3)}{y_1y_4-y_2y_3},$$

$$m_2^{(2)}=\frac{b_1\left(x_2y_4-y_2x_2x_3/x_1\right)+b_2\left(y_1x_2x_3/x_1-x_2y_3\right)}{y_1y_4-y_2y_3},$$

$$m_3^{(2)}=\frac{b_2(x_1y_4-x_3y_2)+b_1(x_3y_1-x_1y_3)}{y_1y_4-y_2y_3}$$

$$m_1^{(3)}=a_3y_5+a_4y_6,$$

$$m_2^{(3)}=a_4y_5+a_3y_6,$$

$$m_3^{(3)}=a_3y_7+a_4y_8,$$

$$m_4^{(3)}=a_4y_7+a_3y_8,$$

$$m_1^{(4)}=\frac{b_3(x_4y_8-x_6y_6)+b_4(x_6y_5-x_4y_7)}{y_5y_4-y_6y_8},$$

-continued

$$m_2^{(4)}=\frac{b_3\left(x_5y_8-y_6\frac{x_5x_6}{x_4}\right)+b_4\left(y_5\frac{x_5x_6}{x_4}-x_5y_7\right)}{y_5y_8-y_6y_7}, \text{ and}$$

$$m_3^{(4)}=\frac{b_4(x_4y_8-x_6y_6)+b_3(x_6y_5-x_xy_7)}{y_5y_8-y_6y_7}.$$

[0056] The variable $m_4^{(2)}$ and $m_4^{(4)}$ of the singular matrices M_2 and M_2 are used as $m_4^{(2)}=m_2^{(2)}m_3^{(2)}/m_1^{(2)}$ and $m_4^{(2)}=m_2^{(2)}m_3^{(2)}/m_1^{(2)}$.

[0057] As a second pass transaction, at step **306**, the process includes Bob receiving the M_1 and M_2 matrices from Alice. At step **306**, the process includes generating uniformly distributed random centrosymmetric matrices C_1, C_2 and inverse C_1^{-1}, C_2^{-1} matrices, as follows:

$$C_1=\begin{vmatrix} c_1 & c_2 \\ c_2 & c_1 \end{vmatrix},$$

$$C_2=\begin{vmatrix} c_3 & c_4 \\ c_4 & c_3 \end{vmatrix},$$

$$C_1^{-1}=\frac{\begin{vmatrix} c_1 & -c_2 \\ -c_2 & c_1 \end{vmatrix}}{c_1^2-c_2^2}, c_i \in R, c_1^2 \neq c_2^2, \text{ and}$$

$$C_2^{-1}=\frac{\begin{vmatrix} c_3 & -c_4 \\ -c_4 & c_3 \end{vmatrix}}{c_3^2-c_4^2}, c_i \in R, c_3^2 \neq c_4^2.$$

[0058] The process at step **306** also includes generating uniformly distributed random matrices D and H, as follows:

$$D=\begin{vmatrix} d_1 & d_2 \\ d_3 & d_4 \end{vmatrix}, \text{ and}$$

$$H=\begin{vmatrix} h_1 & h_2 \\ h_3 & h_4 \end{vmatrix}, d_i, h_i \in R, d_1d_4 \neq d_2d_3, h_1h_4 \neq h_2h_3.$$

[0059] The process at step **306** also includes generating corresponding inverse matrices D^{-1} and H^{-1}, as follows:

$$D^{-1}=\frac{\begin{vmatrix} d_1 & d_2 \\ d_3 & d_4 \end{vmatrix}}{d_1d_4-d_2d_3}, \text{ and}$$

$$H^{-1}=\frac{\begin{vmatrix} h_1 & h_2 \\ h_3 & h_4 \end{vmatrix}}{h_1h_4-h_2h_3}.$$

[0060] The process at step **306** also includes generating the matrices M_5, M_6, M_7 and M_8, as follows:

$$M_5=\begin{vmatrix} m_1^{(5)} & m_2^{(5)} \\ m_3^{(5)} & m_4^{(5)} \end{vmatrix},$$

$$M_6=\begin{vmatrix} m_1^{(6)} & m_2^{(6)} \\ m_3^{(6)} & m_4^{(6)} \end{vmatrix},$$

-continued

$$M_7 = \begin{vmatrix} m_1^{(7)} & m_2^{(7)} \\ m_3^{(7)} & m_4^{(7)} \end{vmatrix}, \text{ and}$$

$$M_8 = \begin{vmatrix} m_1^{(8)} & m_2^{(8)} \\ m_3^{(8)} & m_4^{(8)} \end{vmatrix},$$

as a result of the following calculations:

$$M_5=DM_1C_1^{-1}=D_1Y_1A_1C_1^{-1}, \tag{7}$$

$$M_6=C_1M_2E=C_1B_1Y_1^{-1}Z_1E, \tag{8}$$

$$M_7=E^{-1}M_3C_2^{-1}=E^{-1}YA_2C_2^{-1}, \text{ and} \tag{9}$$

$$M_8=C_2M_4H=C_2B_2Y_2^{-1}Z_2H, \tag{10}$$

[0061] At step **308**, the process includes Bob sending to Alice fourteen publicly visible values ($m_1^{(5)}$, $m_2^{(5)}$, $m_3^{(5)}$, $m_4^{(5)}$, $m_1^{(6)}$, $m_2^{(6)}$, $m_3^{(6)}$, $m_1^{(7)}$, $m_2^{(7)}$, $m_3^{(7)}$, $m_4^{(7)}$, $m_1^{(8)}$, $m_2^{(8)}$, $m_3^{(8)}$) of matrices M_3 and M_4 that are calculated from sixteen independent unknown (for the third party) variables (c_1, c_2, c_3, c_4, d_1, d_2, d_3, d_4, e_1, e_2, e_3, e_4, h_1, h_2, h_3, h_4) which are known by Bob only, as follows:

$$m_1^{(5)} = \frac{c_1(d\textcircled{?}m\textcircled{?} + d_2m\textcircled{?}) - c_2(d\textcircled{?}m\textcircled{?} + d\textcircled{?}m_4^{(1)})}{c\textcircled{?} - c\textcircled{?}},$$

$$m_2^{(5)} = \frac{c_1(d\textcircled{?}m\textcircled{?} + d_2m\textcircled{?}) - c_2(d\textcircled{?}m\textcircled{?} + d\textcircled{?}m_4^{(1)})}{c\textcircled{?} - c\textcircled{?}},$$

$$m_3^{(5)} = \frac{c_1(d\textcircled{?}m\textcircled{?} + d_2m\textcircled{?}) - c_2(d\textcircled{?}m\textcircled{?} + d\textcircled{?}m_4^{(1)})}{c\textcircled{?} - c\textcircled{?}},$$

$$m_4^{(5)} = \frac{c_1(d\textcircled{?}m\textcircled{?} + d_2m\textcircled{?}) - c_2(d\textcircled{?}m\textcircled{?} + d\textcircled{?}m_4^{(1)})}{c\textcircled{?} - c\textcircled{?}},$$

$$m_1^{(6)} = (c_1m_1^{(2)} + c_2m_3^{(2)})e_1 + (c_1m_2^{(2)} + c_2m_4^{(2)})e_3,$$

$$m_2^{(6)} = (c_1m_1^{(2)} + c_2m_3^{(2)})e_2 + (c_3m_2^{(2)} + c_2m_4^{(2)})e_4,$$

$$m_3^{(6)} = (c_2m_1^{(2)} + c_1m_3^{(2)})e_1 + (c_2m_2^{(2)} + c_1m_4^{(2)})e_3,$$

$$m_4^{(6)} = m_2^{(6)}m_3^{(6)}/m_1^{(6)},$$

$$m_1^{(7)} = \frac{c\textcircled{?}(c\textcircled{?}m_1^{(3)} - c\textcircled{?}m\textcircled{?}) - c_4(c\textcircled{?}m_2^{(3)} - c\textcircled{?}m\textcircled{?})}{(c_3^2 - c_4^2)(e_1e_4 - e_2e_3)},$$

$$m_2^{(7)} = \frac{c\textcircled{?}(c\textcircled{?}m_2^{(3)} - c\textcircled{?}m\textcircled{?}) - c_4(c\textcircled{?}m\textcircled{?} - c\textcircled{?}m\textcircled{?})}{(c_3^2 - c_4^2)(e_1e_4 - e_2e_3)},$$

$$m_3^{(7)} = \frac{c\textcircled{?}(c\textcircled{?}m_3^{(3)} - c\textcircled{?}m\textcircled{?}) - c_4(c\textcircled{?}m\textcircled{?} - c\textcircled{?}m\textcircled{?})}{(c_3^2 - c_4^2)(e_1e_4 - e_2e_3)},$$

$$m_4^{(7)} = \frac{c\textcircled{?}(c\textcircled{?}m_4^{(3)} - c\textcircled{?}m\textcircled{?}) - c_4(c\textcircled{?}m\textcircled{?} - c\textcircled{?}m\textcircled{?})}{(c_3^2 - c_4^2)(e_1e_4 - e_2e_3)},$$

$$m_1^{(8)} = (c_3m_1^{(4)} + c_4m_3^{(4)})h_1 + (c_3m_2^{(4)} + c_4m_4^{(4)})h_3,$$

$$m_2^{(8)} = (c_3m_1^{(4)} + c_4m_3^{(4)})h_2 + (c_3m_2^{(4)} + c_4m_4^{(4)})h_4,$$

$$m_3^{(8)} = (c_4m_1^{(4)} + c_3m_3^{(4)})h_1 + (c_4m_2^{(4)} + c_3m_4^{(4)})h_3,$$

$\textcircled{?}$ indicates text missing or illegible when filed

and

[0062] As a third pass transaction, at step **310**, the process includes Alice receiving from Bob the matrices M_5, M_6, M_7 and M_8 as follows:

$$M_5=DY_1A_1C_1^{-1},$$

$$M_6=C_1B_1Y_1^{-1}Z_1E,$$

$$M_7=E^{-1}Y_2A_2C_2^{-1}, \text{ and}$$

$$M_8=CBY^{-1}XH.$$

[0063] Note that centrosymmetric matrices satisfy the following conditions:

$$AC^{-1}=C^{-1}A \text{ and}$$

$$CB=BC,$$

meaning that the matrices M_5, M_6, M_7, and M_8 can be transformed into:

$$M_5=DY_1A_1C_1^{-1}=DY_1C_1^{-1}A_1,$$

$$M_6=C_1B_1Y_1^{-1}Z_1E=B_1C_1Y_1^{-1}Z_1E,$$

$$M_7=E^{-1}Y_2A_2C_2^{-1}=E^{-1}Y_2C_2^{-1}A_2, \text{ and}$$

$$M_8=C_2B_2Y_2^{-1}Z_2H=B_2C_2Y_2^{-1}Z_2H.$$

[0064] Thus, at step **312**, the process includes multiplying the matrices M_5, M_6, M_7 and M_8 with the known inverse matrices A_1^{-1}, A_2^{-1}, B_1^{-1} and B_2^{-1}, respectively, as follows:

$$M_5A_1^{-1}=DY_1C_1^{-1}A_1A_1^{-1}=DY_1C_1^{-1},$$

$$B_1^{-1}M_6=B_1^{-1}B_1C_1Y_1^{-1}Z_1E=C_1Y_1^{-1}Z_1E,$$

$$M_7A_2^{-1}=E^{-1}Y_2C_2^{-1}A_2A_2^{-1}=E^{-1}Y_2C_2^{-1}, \text{ and}$$

$$B_2^{-1}M_8=B_2^{-1}B_2C_2Y_2^{-1}Z_2H=C_2Y_2^{-1}Z_2H.$$

[0065] Further, at step **314**, the process includes multiplying the results of those together to arrive at the matrix M_5, as follows:

$$M_9 = M_5A_1^{-1}B_1^{-1}M_6M_7A_2^{-1}B_2^{-1}M_8,$$

$$M_9 = DY_1C_1^{-1}C_1Y_1^{-1}Z_1EE^{-1}Y_2C_2^{-1}C_2Y_2^{-1}Z_2H = DZ_1Z_2H, \text{ such that}$$

$$M_9 = DXH, \text{ and} \tag{11}$$

$$M_9 = \begin{vmatrix} m_1^{(9)} & m_2^{(9)} \\ m_3^{(9)} & m_4^{(9)} \end{vmatrix}.$$

[0066] At step **316**, the process includes Alice sending the following three publicly visible values to Bob ($m_1^{(9)}$, $m_2^{(9)}$, $m_3^{(9)}$), as follows:

$$m_1^{(9)}=(d_1x_1+d_2x_3)h_1+(d_1x_2+d_2x_4)h_3,$$

$$m_2^{(9)}=(d_1x_1+d_2x_3)h_2+(d_1x_2+d_2x_4)h_4,$$

$$m_3^{(9)}=(d_3x_1+d_4x_3)h_1+(d_3x_2+d_4x_4)h_3, \text{ and}$$

$$m_4^{(9)}=m_3^{(9)}m_2^{(9)}/m_1^{(9)}.$$

[0067] Thus, as part of the final key restoration at step **316**, Bob receives the matrix M9 from Alice, as follows:

$$M_9=DXH.$$

[0068] At step **318**, the process includes Bob restoring the key X from Alice by using inverse matrices D^{-1} and H^{-1}, which are known to Bob, and the matrix M_5, as follows:

$$D^{-1}M_9H^{-1}=D^{-1}DXHH^{-1}=X.$$

[0069] As shown in Table 6 below, the entire scheme of the key exchange process can be performed using an exchange of matrices with a corresponding number of different unknown independent variables (underlined in Table 6) and visible (by the third party) values (bolded in Table 6). This scheme demonstrates that the number of unknown independent variables always exceeds the number of visible independent values in any combination of subsets of matrices.

[0070] This means that the system of nonlinear equations is an indeterminate system. There are no algorithms for the third party to obtain unknown independent variables including the secret key X using the visible independent values.

TABLE 6

			Variables	Independent Variables		Values	
1	Alice	Y_1A_1	$A_1[\underline{2}]$, $Y_1[\underline{4}]$	$\underline{22}$	$M_1[4]$	**4**	**14**
		$B_1Y_1^{-1}Z_1$	$B_1[\underline{2}]$, $Z_1[\underline{3}]$		$M_2[4]$	**3**	
		Y_2A_2	$A_2[\underline{2}]$, $Y_2[\underline{4}]$		$M_3[4]$	**4**	
		$B_2Y_2^{-1}Z_2$	$B_2[\underline{2}]$, $Z_2[\underline{3}]$		$M_4[4]$	**3**	
2	Bob	$DY_1A_1C_1^{-1}$	D [4], $C_1[2]$	$\underline{16}$	$M_5[4]$	**4**	**14**
		$C_1B_1Y_1^{-1}$ Z_1E	E [4]		$M_6[4]$	**3**	
		$B^{-1}Y_2A_2C_2^{-1}$			$M_7[4]$	**4**	
		$C_2B_2Y_2^{-}1$ Z_2H	$C_2[2]$, H [4]		$M_8[4]$	**3**	
3	Alice	DXH			$M_9[3]$	**3**	**3**
	Total			$\underline{38}$		**31**	

[0071] The direct restoration of the matrix X (using formula transformations of Eqs. 3-11 is also impossible. Note that the matrix X is singular. It leads to several features, which are used to perform the key exchange algorithm resistant against the third party decryption (see APPENDIX):

[0072] The matrices M_2, M_4, M_6, M_8, and M_9

$$M_2=B_1Y_1^{-1}Z_1,$$

$$M_4=B_2Y_2^{-1}Z_2,$$

$$M_6=C_1B_1Y_1^{-1}Z_1E,$$

$$M_8=C_2B_2Y_2^{-1}Z_2H, \text{ and}$$

$$M_9=DZ_1Z_2H$$

are also singular (due to the matrices Z_1 and Z_2 being singular).

[0073] Thus, the equation $M_5L_1M_6M_7L_2M_8=M_9$ (from the Eqs. 7-10) can not be resolved in regards to centrosymmetric matrices $L_1=A_1^{-1}B_1^{-1}$ and $L_2=A_2^{-1}B_2^{-1}$ by the third party as far as the matrix M_9 is singular so, the direct calculation $X=M_1L_1M_2M_3L_2M_4$ is not possible.

[0074] The concepts described herein can be used for other cryptographic operations, such as key exchanging using authentication. FIG. **6** illustrates an example secret material or key exchanging process using authentication according to the concepts described herein.

[0075] As shown in FIG. **6**, Alice wants to pass the secret key K to Bob. They use Ed as an independent party for authentication. In the transaction, the square singular matrix

$$X=\begin{vmatrix} x_1 & x_2 \\ x_3 & x_4 \end{vmatrix}$$

is used to represent the key $K=\{k_1, k_2, k_3\}$, where $x_4=x_2x_3/x_1$.

[0076] It is assumed that Alice and Bob both have passed the authentication procedure and both have got corresponding session numbers N_1^A, N_2^A and N_1^B, N_2^B from Ed according to the concepts described above.

[0077] Alice and Bob form centrosymmetric matrices N^A and N^B correspondently, as follows:

$$N_A=\begin{vmatrix} N_1^A & N_2^A \\ N_2^A & N_1^A \end{vmatrix} \text{ and } N_{AB}=\begin{vmatrix} N_1^B & N_2^B \\ N_2^B & N_1^B \end{vmatrix}.$$

[0078] As part of a first pass transaction, at step **402**, the process **40** includes Alice generating uniformly distributed random matrices Y_1, Y_2 and inverse matrices Y_1^{-1}, Y_2^{-1}, as follows:

$$Y_1=\begin{vmatrix} y_1 & y_2 \\ y_3 & y_4 \end{vmatrix},$$

$$Y_1^{-1}=\frac{\begin{vmatrix} y_4 & -y_2 \\ -y_3 & y_1 \end{vmatrix}}{y_1y_4-y_2y_3},\ y_i \in R,\ y_1y_4 \neq y_2y_3,$$

$$Y_2=\begin{vmatrix} y_5 & y_6 \\ y_7 & y_8 \end{vmatrix}, \text{ and}$$

$$Y_2^{-1}=\frac{\begin{vmatrix} y_5 & -y_6 \\ -y_7 & y_8 \end{vmatrix}}{y_5y_8-y_6y_7},\ y_i \in R,\ y_5y_8 \neq y_6y_7.$$

[0079] Alice also generates uniformly distributed random centrosymmetric matrices A and B, as follows:

$$A_1=\begin{vmatrix} a_1 & a_2 \\ a_2 & a_1 \end{vmatrix},$$

$$A_2=\begin{vmatrix} a_3 & a_4 \\ a_4 & a_3 \end{vmatrix},$$

$$A_1^{-1}=\frac{\begin{vmatrix} a_1 & -a_2 \\ -a_2 & a_1 \end{vmatrix}}{a_1^2-a_2^2},$$

-continued

$$A_2^{-1} = \frac{\begin{vmatrix} a_3 & a_4 \\ a_4 & a_3 \end{vmatrix}}{a_3^2 - a_4^2}, a_i \in R, a_1^2 \neq a_2^2, a_3^2 \neq a_4^2,$$

$$B_1 = \begin{vmatrix} b_1 & b_2 \\ b_2 & b_1 \end{vmatrix},$$

$$B_2 = \begin{vmatrix} b_3 & b_4 \\ b_4 & b_3 \end{vmatrix},$$

$$B_2^{-1} = \begin{vmatrix} b_3 & -b_4 \\ -b_4 & b_3 \end{vmatrix}, b_i \in R, b_1^2 \neq b_2^2, b_3^2 \neq b_4^2,$$

[0080] Note that any centrosymmetric square matrices A and B always have the following feature: AB=BA. At step **404**, the process includes Alice sending to Bob results as matrices M_1 and M_2, as follows:

$$M_1 = \begin{vmatrix} m_1^{(1)} & m_2^{(1)} \\ m_3^{(1)} & m_4^{(1)} \end{vmatrix},$$

$$M_2 = \begin{vmatrix} m_1^{(2)} & m_2^{(2)} \\ m_3^{(2)} & m_4^{(2)} \end{vmatrix},$$

$$M_3 = \begin{vmatrix} m_1^{(3)} & m_2^{(3)} \\ m_3^{(3)} & m_4^{(3)} \end{vmatrix}, \text{ and}$$

$$M_4 = \begin{vmatrix} m_1^{(4)} & m_2^{(4)} \\ m_3^{(4)} & m_4^{(4)} \end{vmatrix}.$$

of the following calculations:

$$M_1=Y_1A_1, \tag{1B}$$

$$M_2=B_1Y_1^{-1}Z_1, \tag{2B}$$

$$M_3=Y_2A_2, \text{ and} \tag{3B}$$

$$M_4=B_2Y_2^{-1}Z_2. \tag{4B}$$

[0081] As part of a second pass transaction, at step **406**, Bob receives M_1 and M_2 from Alice. Bob generates uniformly distributed random centrosymmetric matrices C_1, C_2 and inverse C_1^{-1}, C_2^{-1} matrices, as follows:

$$C_1 = \begin{vmatrix} c_1 & c_2 \\ c_2 & c_1 \end{vmatrix},$$

$$C_2 = \begin{vmatrix} c_3 & c_4 \\ c_4 & c_3 \end{vmatrix},$$

$$C_1^{-1} = \frac{\begin{vmatrix} c_1 & -c_2 \\ -c_2 & c_1 \end{vmatrix}}{c_1^2 - c_2^2}, c_i \in R, c_1^2 \neq c_2^2, \text{ and}$$

$$C_2^{-1} = \frac{\begin{vmatrix} c_3 & -c_4 \\ -c_4 & c_3 \end{vmatrix}}{c_3^2 - c_2^2}, c_i \in R, c_3^2 \neq c_4^2.$$

and uniformly distributed random matrices D and H, as follows:

$$D = \begin{vmatrix} d_1 & d_2 \\ d_3 & d_4 \end{vmatrix}, \text{ and}$$

-continued

$$H = \begin{vmatrix} h_1 & h_2 \\ h_3 & h_4 \end{vmatrix}, d_i, h_i \in R, d_1d_4 \neq d_2d_3, h_1h_4 \neq h_2h_3,$$

and correspondent inverse matrices D^{-1} and H^{-1}, as follows:

$$D^{-1} = \frac{\begin{vmatrix} d_1 & d_2 \\ d_3 & d_4 \end{vmatrix}}{d_1d_4 - d_2d_3}, \text{ and}$$

$$H^{-1} = \frac{\begin{vmatrix} h_1 & h_2 \\ h_3 & h_4 \end{vmatrix}}{h_1h_4 - h_2h_3}.$$

[0082] At step **406**, Bob also obtains the matrices M_5, M_6, M_7 and M_8, defined as follows:

$$M_5 = \begin{vmatrix} m_1^{(5)} & m_2^{(5)} \\ m_3^{(5)} & m_4^{(5)} \end{vmatrix},$$

$$M_6 = \begin{vmatrix} m_1^{(6)} & m_2^{(6)} \\ m_3^{(6)} & m_4^{(6)} \end{vmatrix},$$

$$M_7 = \begin{vmatrix} m_1^{(7)} & m_2^{(7)} \\ m_3^{(7)} & m_4^{(7)} \end{vmatrix}, \text{ and}$$

$$M_8 = \begin{vmatrix} m_1^{(8)} & m_2^{(8)} \\ m_3^{(8)} & m_4^{(8)} \end{vmatrix},$$

as a result of the following calculations:

$$M_5=DM_1C_1^{-1}=D_1Y_1A_1C_1^{-1}, \tag{5B}$$

$$M_6=C_1M_2E=C_1B_1Y_1^{-1}Z_1E, \tag{6B}$$

$$M_7=E^{-1}M_3C_2^{-1}=E^{-1}YA_2C_2^{-1}, \text{ and} \tag{7B}$$

$$M_8=C_2M_4H=C_2B_2Y_2^{-1}Z_2H, \tag{8B}$$

[0083] As part of a third pass transaction, at step **408**, the process includes Alice generating a uniformly distributed random matrix G, as follows:

$$G = \begin{vmatrix} g_1 & g_2 \\ g_3 & g_4 \end{vmatrix}, g_i \in R.$$

[0084] Alice receives from Bob the matrices M_5, M_6, M_7 and M_8, as follows:

$$M_5=DY_1A_1C_1^{-1}=DY_1C_1^{-1}A_1,$$

$$M_6=C_1B_1Y_1^{-1}Z_1E=B_1C_1Y_1^{-1}Z_1E,$$

$$M_7=E^{-1}Y_2A_2C_2^{-1}=E^{-1}Y_2C_2^{-1}A_2, \text{ and}$$

$$M_8=C_2B_2Y_2^{-1}Z_2H=B_2C_2Y_2^{-1}Z_2H.$$

[0085] At step **408**, the process also includes Alice multiplying both the matrices M_5, M_6, M_7 and M_8 with the inverse matrices which are known to her, A_1^{-1}, A_2^{-1}, B_1^{-1} and B_2^{-1}, respectively, as follows:

$$B_1^{-1}M_6 = B_1^{-1}B_1C_1Y_1^{-1}Z_1E = C_1Y_1^{-1}Z_1E,$$

$$B_1^{-1}M_6 = B_1^{-1}B_1C_1Y_1^{-1}Z_1E = C_1Y_1^{-1}Z_1E,$$

$$M_7A_2^{-1} = E^{-1}Y_2C_2^{-1}A_2A_2^{-1} = E^{-1}Y_2C_2^{-1}, \text{ and}$$

$$B_2^{-1}M_8 = B_2^{-1}B_2C_2Y_2^{-1}Z_2H = C_2Y_2^{-1}Z_2H,$$

$$M_5 = GM_5A_1^{-1}B_1^{-1}M_6M_7A_2^{-1}B_2^{-1}M_8 = GDY_1C_1^{-1}C_1Y_1^{-1}Z_1EE^{-1}Y_2C_2^{-1}C_2Y_2^{-1}Z_2H, \text{ such}$$

$$M_9 = GDXH \text{ where,} \qquad (9B)$$

$$M_9 = \begin{vmatrix} m_1^{(9)} & m_2^{(9)} \\ m_3^{(9)} & m_4^{(9)} \end{vmatrix}.$$

[0086] At step **410**, the process includes Alice sending three publicly visible values to Bob, including ($m_1^{(9)}$, $m_2^{(9)}$, $m_3^{(9)}$). The matrix M_9 is singular and $m_4^{(9)}=m_3^{(9)}m_2^{(9)}/m_1^{(9)}$. At step **412**, Alice also sends four publicly visible values to Ed ($m_1^{(6)}$,$m_2^{(6)}$,$m_3^{(6)}$,$m_4^{(6)}$) of the matrix M_{10}, defined as:

$$M_{10} = \begin{vmatrix} m_1^{(10)} & m_2^{(10)} \\ m_3^{(10)} & m_4^{(10)} \end{vmatrix},$$

as a result of the following calculations:

$$M_{10}-N^AG. \qquad (9B)$$

[0087] For authentication, Ed receives the matrix M6 from Alice. At step **414**, Ed sends to Bob the matrix M_{11} using the inverse matrix $(N^A)^{-1}$ and the matrix N^B as follows:

$$M_{11}=N^B(N^A)^{-1}N^AG=N^BG,$$

$$M_{11}=N^BG. \qquad (10B).$$

[0088] As part of the final key restoration, at step **416**, the process includes Bob receiving the matrix M_{11} from Ed and obtaining the matrix G using the inverse matrix $(N^B)^{-1}$, as follows:

$$G=(N^B)^{-1}M_{11}=(N^B)^{-1}N^BG.$$

[0089] Bob also receives the matrix M_9 from Alice at step **410**. Using inverse matrices G^{-1}, D^{-1}, and H^{-1}, which are known to Bob, he can restore the key X from the received matrix M_5 as follows:

$$D^{-1}G^{-1}M_9H^{-1}=D^{-1}G^{-1}GDXH\ H^{-1}=X.$$

[0090] The embodiments described herein can be implemented by either a method or process or as a system or device. The method can be performed using any suitable computing device, and the system can be embodied as any suitable computing device. The computing device can include at least one processing system, for example, having one or more processors and memories electrically and communicatively coupled together using a local interface. The local interface can be embodied as a data bus with an accompanying address/control bus or other addressing, control, and/or command lines.

[0091] In various embodiments, the memory can store data and software or executable code components executable by the processor. For example, the memory can store executable-code components associated with cryptographic operations for execution by the processor. The software or executable-code components can be developed using or embodied in various programming languages, such as, for example, C, C++, C#, Objective C, JAVA®, JAVASCRIPT®, Perl, PHP, VISUAL BASIC®, PYTHON®, RUBY, FLASH®, or other programming languages.

[0092] The embodiments can rely, in part, on executable instructions or instructions for execution by the computing device. The terms "executable" or "for execution" refer to software forms that can ultimately be run or executed by a processor, whether in source, object, machine, or other form. Examples of executable programs include, for example, a compiled program that can be translated into a machine code format and loaded into a random access portion of memory and executed by a processor, source code that can be expressed in an object code format and loaded into a random access portion of the memory and executed by the processor, or source code that can be interpreted by another executable program to generate instructions in a random access portion of the memory and executed by the processor, etc.

[0093] An executable program can be stored in any portion or component of the memory including, for example, a random access memory (RAM), read-only memory (ROM), magnetic or other hard disk drive, solid-state, semiconductor, or similar drive, universal serial bus (USB) flash drive, memory card, optical disc (e.g., compact disc (CD)) or digital versatile disc (DVD)), floppy disk, magnetic tape, or other memory component.

[0094] Although the process diagram shown in FIGS. **2** and **5** illustrate a certain order, it is understood that the order can differ from that which is depicted. For example, an order of execution of two or more blocks can be scrambled relative to the order shown. Also, two or more blocks shown in succession can be executed concurrently or with partial concurrence. Further, in some embodiments, one or more of the blocks can be skipped or omitted. In addition, any number of counters, state variables, warning semaphores, or messages might be added to the logical flow described herein, for purposes of enhanced utility, accounting, performance measurement, or providing troubleshooting aids, etc. It is understood that all such variations are within the scope of the present disclosure.

[0095] Also, any algorithm, method, process, or logic described herein that are embodied, at least in part, by software or executable-code components, can be embodied or stored in any tangible or non-transitory computer-readable medium or device for execution by an instruction execution system such as a general purpose processor. In this sense, the logic can be embodied as, for example, software or executable-code components that can be fetched from the computer-readable medium and executed by the instruction execution system. Thus, the instruction execution system can be directed by execution of the instructions to perform certain processes such as those illustrated in FIG. **2**. In the context of the present disclosure, a "computer-readable medium" can be any tangible medium that can contain, store, or maintain any logic, application, software, or executable-code component described herein for use by or in connection with an instruction execution system.

[0096] The computer-readable medium can include any physical media such as, for example, magnetic, optical, or semiconductor media. More specific examples of suitable computer-readable media include, but are not limited to, magnetic tapes, magnetic floppy diskettes, magnetic hard drives, memory cards, solid-state drives, USB flash drives, or optical discs. Also, the computer-readable medium can include a RAM including, for example, an SRAM, DRAM, or MRAM. In addition, the computer-readable medium can include a ROM, a PROM, an EPROM, an EEPROM, or other similar memory device.

[0097] Disjunctive language, such as the phrase "at least one of X, Y, or Z," unless specifically stated otherwise, is to be understood with the context as used in general to present that an item, term, etc., can be either X, Y, or Z, or any combination thereof (e.g., X, Y, and/or Z). Thus, such disjunctive language is not generally intended to, and should not, imply that certain embodiments require at least one of X, at least one of Y, or at least one of Z to be each present.

[0098] It should be emphasized that the above-described embodiments of the present disclosure are merely possible examples of implementations set forth for a clear understanding of the principles of the disclosure. Many variations and modifications can be made to the above-described embodiment(s) without departing substantially from the spirit and principles of the disclosure. All such modifications and variations are intended to be included herein within the scope of this disclosure and protected by the following claims.

1-20. (canceled)

21. A method of cryptography, comprising:

generating, with a first computing device, a first random lock;

applying, with the first computing device, a cryptographic function to a combination of secret data and the first random lock to produce a first encrypted result; and

transmitting, with the first computing device, the first encrypted result to a second computing device.

22. The method of cryptography according to claim **21**, further comprising:

generating, with the second computing device, a second random lock;

applying, with the second computing device, a second cryptographic function to a combination of the first encrypted result and the second random lock to produce a second encrypted result; and

transmitting, with the second computing device, the second encrypted result to the first computing device.

23. The method of cryptography according to claim **22**, further comprising:

applying, with the first computing device, a first inverse cryptographic function to the second encrypted result to produce a first result;

transmitting, with the first computing device, the first result to the second computing device; and

applying, with the second computing device, a second inverse cryptographic function to the first result to decrypt the secret data at the second computing device.

24. The method of cryptography according to claim **21**, wherein the cryptographic function comprises at least one of a polynomial function or a multivariate polynomial function defined in part by one or more variables and coefficients.

25. The method of cryptography according to claim **21**, further comprising applying, with the computing device, an inverse cryptographic function to the first encrypted result to arrive at the secret data.

26. The method of cryptography according to claim **25**, wherein the inverse cryptographic function comprises at least one of an inverse polynomial function or an inverse multivariate polynomial function.

27. The method of cryptography according to claim **21**, further comprising:

transmitting, with the first computing device, a symmetric key from the first computing device to the second computing device, wherein the transmitting comprises:

encrypting the symmetric key by applying at least one associative cryptography key operation to the symmetric key to create an encrypted symmetric key; and

decrypting the encrypted symmetric key by applying at least one associative cryptography key operation to the encrypted symmetric key to produce the symmetric key.

28. The method of cryptography according to claim **27**, wherein the at least one associative cryptography key operation is based on applying a polynomial function composed by a combination of independent random variables with uniform probability distribution and white noise randomization.

29. A computing device comprising:

a processor; and

a memory for storing an application to be executed by the processor, the application for:

generating, with a first computing device, a first random lock;

applying, with the first computing device, a cryptographic function to a combination of secret data and the first random lock to produce a first encrypted result; and

transmitting, with the first computing device, the first encrypted result to a second computing device.

30. The computing device according to claim **29**, wherein the application is further for:

generating, with the second computing device, a second random lock;

applying, with the second computing device, a second cryptographic function to a combination of the first encrypted result and the second random lock to produce a second encrypted result; and

transmitting, with the second computing device, the second encrypted result to the first computing device.

31. The computing device according to claim **30**, wherein the application is further for:

applying, with the first computing device, a first inverse cryptographic function to the second encrypted result to produce a first result;

transmitting, with the first computing device, the first result to the second computing device; and

applying, with the second computing device, a second inverse cryptographic function to the first result to decrypt the secret data at the second computing device.

32. The computing device according to claim **29**, wherein the cryptographic function comprises at least one of a polynomial function or a multivariate polynomial function defined in part by one or more variables and coefficients.

33. The computing device according to claim **29**, wherein the application is further for applying, with the computing device, an inverse cryptographic function to the first encrypted result to arrive at the secret data.

34. The computing device according to claim **33**, wherein the inverse cryptographic function comprises at least one of an inverse polynomial function or an inverse multivariate polynomial function.

35. The computing device according to claim **29**, wherein the application is further for:

transmitting, with the first computing device, a symmetric key from the first computing device to the second computing device, wherein the transmitting comprises:

encrypting the symmetric key by applying at least one associative cryptography key operation to the symmetric key to create an encrypted symmetric key; and

decrypting the encrypted symmetric key by applying at least one associative cryptography key operation to the encrypted symmetric key to produce the symmetric key.

36. The computing device according to claim **35**, wherein the at least one associative cryptography key operation is based on applying a polynomial function composed by a combination of independent random variables with uniform probability distribution and white noise randomization.

37. A method of cryptography, comprising:

generating, with a first computing device, a first random lock;

applying, with the first computing device, a cryptographic function to a combination of secret data and the first random lock to produce a first encrypted result;

transmitting, with the first computing device, the first encrypted result to a second computing device;

generating, with the second computing device, a second random lock;

applying, with the second computing device, a second cryptographic function to a combination of the first encrypted result and the second random lock to produce a second encrypted result;

transmitting, with the second computing device, the second encrypted result to the first computing device;

applying, with the first computing device, a first inverse cryptographic function to the second encrypted result to produce a first result;

transmitting, with the first computing device, the first result to the second computing device; and

applying, with the second computing device, a second inverse cryptographic function to the first result to decrypt the secret data at the second computing device.

38. The method of cryptography according to claim **37**, wherein the cryptographic function comprises at least one of a polynomial function or a multivariate polynomial function defined in part by one or more variables and coefficients.

39. The method of cryptography according to claim **37**, further comprising applying, with the computing device, an inverse cryptographic function to the first encrypted result to arrive at the secret data.

40. The method of cryptography according to claim **39**, wherein the inverse cryptographic function comprises at least one of an inverse polynomial function or an inverse multivariate polynomial function.

41. The method of cryptography according to claim **37**, further comprising:

transmitting, with the first computing device, a symmetric key from the first computing device to the second computing device, wherein the transmitting comprises:

encrypting the symmetric key by applying at least one associative cryptography key operation to the symmetric key to create an encrypted symmetric key; and

decrypting the encrypted symmetric key by applying at least one associative cryptography key operation to the encrypted symmetric key to produce the symmetric key.

42. The method of cryptography according to claim **42**, wherein the at least one associative cryptography key operation is based on applying a polynomial function composed by a combination of independent random variables with uniform probability distribution and white noise randomization.

* * * * *