CLAIMS

1. An encryption method comprising:

Generating a Functional Neural Network or KeyNode (KN) of a system by programming a chain of multiple nodes also called Artificial Mirror Neurons (AMN) based on captured information of reaction time and emotional response to tasks; one task per mode, racing the nodes in the Functional Neural Network or KeyNode (KN) and their valued output services as an encryption device or cipher for a time of use;

Classifying the KeyNodes where each KeyNode comprises multiple nodes held in a chain which can be generated successively, elongated, and can also evolve by retraction of nodes or decay after a certain number of nodes has been generated;

Encoding KeyNodes based on infinite proposition captured by interactive cognitive computing, forming an irregular tetragon matrix, and allowing dynamic encoding of unusual forms of cryptography;

Generating a boundary of the KeyNode, where a minimum size is defined as A, the maximum size is defined as B, and a time of maximum size of the KeyNode is defined as $C = B - (A - 1)$;

Generating one-time-password or passphrase for decryption at the user's time of use based on a sum of intrinsic values of the nodes in a racing network at this time and adopting the generated password for authentication or adopting the generated passphrase to generate a one-time-pair of keys for encrypted communication to form a one-time-pair of keys for encrypted communication;

Transmitting a one-time encryption key (public key) over any unsecured channel to selected senders for encrypting a plaintext to become a ciphertext;

Decrypting the ciphertext of senders at receiver's end using a one-time decryption key (private key) and the passphrase to regenerate a plaintext.

2. The method as claimed in claim 1, wherein, the steps of programing Artificial Mirror Neuron or node, further comprising capturing reaction and emotional response where the response time is the time taken by the user to complete a cognitive task such as answering a question or making a choice or even taking a selfie between input and output, which determines the length of the motor core radius; and the emotional response is the emotional response to the cognitive task captured around 500ms from the input time, which determines the length of the sensory core radius; and determining the velocity which is inversely to the period of time calculated since user's birthdate (time since birth) and expressed in Linux time in milliseconds; and deriving the angular velocity ω, defined as a pseudovector representation of how the angular position of the radius of

both motor and sensory cores changes with time, from velocity and the angle θ of both motor core and sensory core; the steps of racing the chain of nodes in the Functional Neural Network or KeyNode (KN), further comprising programming multiple nodes, and the chain of nodes in the Functional Neural Network or KeyNode (KN) serves as an encryption device or a cipher; KeyNodes comprise one first signup KeyNode or KN1 which contains at least 8 nodes;

the steps of generating a passphrase, further comprising generating passphrase for use at time 2 (T2), on the base of each node angle θ changes constantly between time 1 (T1) and time 2 (T2), and a sum of angle θ of all nodes also called sum of angle θ2 at time 2 depends on the position of the radiuses or the value of the velocities of the nodes at time 2; and

the method further comprising defining each node by motor core radius r1 and sensory core radius r2, and respectively angle θ1 and angle θ2, and respectively velocity v1 and velocity v2, and respectively angular velocity ω1 and angular velocity ω2 at the time T1 and T2, and the time of each node creation in minute or in millisecond depending on the case and each new node has a specific time of creation T1+n that will refer to the initial nodes time of creation when made at signup or T1; and

when reaching the time T2 or time of applying passphrase, creating a new node by generation-elongation, and summing up motor core and sensory core radius theta angles or angular velocities for all racing nodes in the KN arriving at T2, and adopting passphrase for authentication.

3. The method as claimed in claim 2,

using the angle θ for one time passphrase:

$$\text{One-time passphrase at time T} \quad = \frac{1}{\mathrm{N_T}} \sum_{i=0}^{N_T} (\mathrm{v_{mi}T})\%(2\pi \mathrm{r_{1i}}) \; + \; \frac{1}{\mathrm{N_T}} \sum_{i=0}^{N_T} (\mathrm{v_{Si}T})\%(2\pi \mathrm{r_{2i}})$$

using the angular velocities ω for one time passphrase is:

$$\text{One-time Passphrase at time T} \quad = \frac{1}{\mathrm{N_T}} \sum_{i=0}^{N_T} \frac{\mathrm{v_{mi}}}{2\pi \mathrm{r_{1i}}} \; + \; \frac{1}{\mathrm{N_T}} \sum_{i=0}^{N_T} \frac{\mathrm{v_{Si}}}{2\pi \mathrm{r_{2i}}}$$

where v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN and N_T is the number of nodes in FNN at time T.

4. The method as claimed in claim 2, wherein for the motor cores, the following steps are provided:

$$\text{individual angle theta from M of each node at time T} \quad = (\mathrm{v_m T})\%(2\pi \mathrm{r_1})$$

$$\text{sum of angles theta from M of the whole FNN at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{mi}T)\%(2\pi r_{1i})$$

where v_m is the velocity of the motor core in an AMN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_1 is the radius of the motor core in an AMN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T.

wherein for the sensory cores, the following steps are provided:

$$\text{individual angle theta from S of each node at time T} \quad = (v_s T)\%(2\pi r_2)$$

$$\text{sum of angles theta from S of the whole FNN at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{si}T)\%(2\pi r_{2i})$$

where v_s is the velocity of the sensory core in an AMN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_2 is the radius of the sensory core in an AMN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T.

wherein for the Angular Velocity per core, the following steps are provided:

$$\text{individual angular velocity from M of each node at time T} \quad = \frac{v_m}{2\pi r_1}$$

$$\text{sum of angular velocities from M of the whole FNN at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{mi}}{2\pi r_{1i}}$$

where v_m i the velocity of the motor core in an AMN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_1 is the radius of the motor core in an AMN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T;

Similarly, for the sensory cores, the following steps are provided:

$$\text{individual angular velocity from S of each node at time T} \quad = \frac{v_s}{2\pi r_2}$$

$$\text{sum of angular velocities from S of the whole FNN at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{si}}{2\pi r_{2i}}$$

where v_s is the velocity of the sensory core in an AMN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_2 is the radius of the sensory core in an AMN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T.

5. The method as claimed in claim 1, wherein the steps of encoding KeyNodes based on infinite proposition captured by interactive cognitive computing, there are four encoding schemes such as Question answer-types (Qa), Binary visual-choices (Bv), binary Iconic-choices (Ic) and Ternary visual-choices (Tv) to form the irregular tetragon matrix; generating a new node in the KeyNode according to the output from one of these four types, producing a cipher combining both the reactional nature and the timeliness of nodes and refreshing the FNN with a new node.

6. The method as claimed in claim 2, wherein one time passphrase is fed into Rivest–Shamir–Adleman

("RSA") or elliptic-curve cryptography ("ECC") encryption algorithms to generate a pair of one-time encryption key and decryption key.

7. The method as claimed in claim 6. if theta angles are used for one-time encryption key and decryption key, then

One-time encryption key and decryption key at time T for RSA

$$= \mathrm{RSA}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{mi}T})\%(2\pi\mathrm{r_{1i}}) + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{Si}T})\%(2\pi\mathrm{r_{2i}})\right)$$

or

One-time encryption key and decryption key at time T for ECC

$$= \mathrm{ECC}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{mi}T})\%(2\pi\mathrm{r_{1i}}) + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{Si}T})\%(2\pi\mathrm{r_{2i}})\right)$$

if angular velocities are used for one-time encryption key and decryption key, then

One-time encryption key and decryption key at time T for RSA $= \mathrm{RSA}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{mi}}}{2\pi\mathrm{r_{1i}}} + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{Si}}}{2\pi\mathrm{r_{2i}}}\right)$

And/or

One-time encryption key and decryption key at time T for ECC $= \mathrm{ECC}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{mi}}}{2\pi\mathrm{r_{1i}}} + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{Si}}}{2\pi\mathrm{r_{2i}}}\right)$

where RSA( ) is the RSA algorithmic function for generating encryption key and decryption key with the passphrase, and ECC( ) is the ECC algorithmic function for generating encryption key and decryption key with the passphrase, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN and N_T is the number of nodes in the FNN at time T.

8. The method as claimed in claim 7, wherein using the motor cores or the sensory cores respectively for generating a private key compatible with ECC:

$$Concatenate\ [\ (v_{m1}T)\%(2\pi r_{11}), (v_{m2}T)\%(2\pi r_{12}), \dots\ , (v_{m8}T)\%(2\pi r_{18})\ , \frac{1}{8}\sum_{i=1}^{8}(v_{mi}T)\%(2\pi r_{1i})]$$

or

$$Concatenate\ [\ (v_{s1}T)\%(2\pi r_{21}), (v_{s2}T)\%(2\pi r_{22}), \dots\ , (v_{s8}T)\%(2\pi r_{28})\ , \frac{1}{8}\sum_{i=1}^{8}(v_{si}T)\%(2\pi r_{2i})]$$

where vm are the velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN; or

Combining the motor cores with the sensory cores for generating a private key compatible with ECC:

$$Concatenate\ [\ (v_{s1}T)\%(2\pi r_{21}), (v_{m1}T)\%(2\pi r_{11}), (v_{s2}T)\%(2\pi r_{22}), (v_{s3}T)\%(2\pi r_{23}), (v_{m2}T)\%(2\pi r_{12}),$$

$$(v_{s4}T)\%(2\pi r_{24}), (v_{m3}T)\%(2\pi r_{13}), (v_{m4}T)\%(2\pi r_{14})\ , \frac{1}{16}\sum_{i=1}^{8}\{(v_{si}T)\%(2\pi r_{2i}) + (v_{mi}T)\%(2\pi r_{1i})\}\ ]$$

where vm are the velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN; and

combining the motor cores with the sensory cores for generating a private key compatible with RSA:

$$Concatenate\ [\ (v_{m1}T)\%(2\pi r_{11}), \dots, (v_{m24}T)\%(2\pi r_{1\,24}), (v_{s1}T)\%(2\pi r_{21}), \dots, (v_{s24}T)\%(2\pi r_{2\,24})]$$

where vm are the velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN; or

combining the motor cores with the sensory cores and a random generator for generating a private key compatible with RSA:

$$Concatenate\ [\ (v_{m1}T)\%(2\pi r_{11}), \dots, (v_{m8}T)\%(2\pi r_{18}), \frac{1}{8}\sum_{i=1}^{8}(v_{mi}T)\%(2\pi r_{1i}),$$

$$(v_{s1}T)\%(2\pi r_{21}), \ldots, (v_{s8}T)\%(2\pi r_{28}), \frac{1}{8}\sum_{i=1}^{8}(v_{si}T)\%(2\pi r_{2i}),\ random_generator(1920\ bits)]$$

where the "random_generator(bits)" is a function for generating 1920 bits of digits randomly, vm are the velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN.

9. An encryption system comprising:

KeyNodes Generation module, where the Functional Neural Networks or KeyNodes (KN) of a system are generated by programming a chain of multiple nodes also called Artificial Mirror Neurons (AMN) based on captured information of reaction time and emotional response to tasks, one task per node; the nodes in the Functional Neural Network or KeyNode (KN) are racing, and their valued output serves as encryption device or cipher for a time of use;

KeyNodes classification module, where each KeyNode comprises multiple nodes held in a chain which can be generated successively, elongated, and can also evolve by retraction of nodes or decay after a certain number of nodes has been generated;

KeyNodes encoding module, where KeyNodes are encoded based on infinite proposition captured by interactive cognitive computing, forming an irregular tetragon matrix, and allowing dynamic encoding of unusual forms of cryptography;

Boundary generation module, where a boundary of the KeyNode is generated, and where a minimum size is defined as A, the maximum size is defined as B, and a time of maximum size of the KeyNode is defined as C = B – (A – 1);

Pair of keys generation module, where the one-time-password or passphrase is generated for decryption at the user's time of use based on a sum of intrinsic values of the nodes in a racing network at this time, and adopting the generated password for authentication or adopting the generated passphrase to generate a one-time-pair of keys for encrypted communication to form a one-time-pair of keys for encrypted communication;

Key transmission module, where a one-time encryption key (public key) is transmitted over any unsecured channel to selected senders for encrypting a plaintext to become a ciphertext;

Decryption module, where the ciphertext of senders at receiver's end is decrypted using a one-time decryption key (private key) and the passphrase to regenerate a plaintext.

10. The system as claimed in claim 9, wherein, in the KeyNode Generation module, further comprising reaction and emotional response capturing where the response time is the time taken by the user to complete a cognitive task such as answering a question or making a choice or even taking a selfie between input and output, which determines the length of the motor core radius; and the emotional response is the emotional response to the cognitive task captured around 500ms from the input time, which determines the length of the sensory core radius; the velocity which is inversely to the period of time calculated since user's birthdate (time since birth) and expressed in Linux time in milliseconds is determined; and the angular velocity ω from velocity and the angle θ of both motor core and sensory core are derived;

further comprising programming multiple nodes, and the chain of nodes in the Functional Neural Network or KeyNode (KN) serves as an encryption device or a cipher; KeyNodes comprise one first signup KeyNode or KN1 which contains at least 8 nodes;

further comprising passphrase for use at time 2 (T2) is generated, on the base of each node angle θ changes constantly between time 1 (T1) and time 2 (T2), and a sum of angle θ of all nodes also called sum of angle θ2 at time 2 depends on the position of the radiuses or the value of the velocities of the nodes at time 2; and

the system further comprising a definition module, where each node by motor core radius r1 and sensory core radius r2 is defined by, respectively angle θ1 and angle θ2, respectively velocity v1 and velocity v2, and respectively angular velocity ω1 and angular velocity ω2 at the time T1 and T2, and the time of each node creation in minute and each new node has a specific time of creation T1+n that will refer to the initial nodes time of creation when made at signup or T1; and

when the time T2 or time of applying passphrase is reached, a new node by generation-elongation is created, and motor core and sensory core radius theta angles or angular velocities for all racing nodes in the KN arriving at T2 is summed up and passphrase for authentication is adopted.

11. The system as claimed in claim 10, if the angle θ for one time passphrase is used:

$$\text{one time Passphrase at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{mi}T)\%(2\pi r_{1i}) \; + \; \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{Si}T)\%(2\pi r_{2i})$$

if the angular velocities ω for one time passphrase is used:

$$\text{one time Passphrase at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{mi}}{2\pi r_{1i}} \; + \; \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{Si}}{2\pi r_{2i}}$$

where v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor

core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN and N_T is the number of nodes in FNN at time T.

12. The system as claimed in claim 10, wherein for the motor cores:

$$\text{individual angle theta from M of each node at time T} = (v_m T)\%(2\pi r_1)$$

$$\text{sum of angles theta from M of the whole FNN at time T} = \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{mi} T)\%(2\pi r_{1i})$$

where v_m is the velocity of the motor core in an AMN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_1 is the radius of the motor core in an AMN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T.

wherein for the sensory cores:

$$\text{individual angle theta from S of each node at time T} = (v_s T)\%(2\pi r_2)$$

$$\text{sum of angles theta from S of the whole FNN at time T} = \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{si} T)\%(2\pi r_{2i})$$

where v_s is the velocity of the sensory core in an AMN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_2 is the radius of the sensory core in an AMN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T.

wherein for the Angular Velocity per core:

$$\text{individual angular velocity from M of each node at time T} = \frac{v_m}{2\pi r_1}$$

$$\text{sum of angular velocities from M of the whole FNN at time T} = \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{mi}}{2\pi r_{1i}}$$

where v_m is the velocity of the motor core in an AMN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_1 is the radius of the motor core in an AMN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T;

Similarly, for the sensory cores:

$$\text{individual angular velocity from S of each node at time T} = \frac{v_s}{2\pi r_2}$$

$$\text{sum of angular velocities from S of the whole FNN at time T} = \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{si}}{2\pi r_{2i}}$$

where v_s is the velocity of the sensory core in an AMN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_2 is the radius of the sensory core in an AMN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN, and N_T is the number of nodes in FNN at time T.

13. The system as claimed in claim 9, wherein, in the KeyNodes encoding module, KeyNodes are encoded based on infinite proposition captured by interactive cognitive computing, there are four encoding schemes such as Question answer-types (Qa), Binary visual-choices (Bv), binary Iconic-choices (Ic) and Ternary

visual-choices (Tv) to form the irregular tetragon matrix; generating a new node in the KeyNode according to the output from one of these four types, producing a cipher combining both the reactional nature and the timeliness of nodes and refreshing the FNN with a new node.

14. The system as claimed in claim 10, wherein one time passphrase is fed into Rivest–Shamir–Adleman ("RSA") or elliptic-curve cryptography ("ECC") encryption algorithms to generate a pair of one-time encryption key and decryption key.

15. The system as claimed in claim 14, wherein, if theta angles are used for one-time encryption key and decryption key, then:

One-time encryption key and decryption key at time T for RSA

$$= \mathrm{RSA}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{mi}T})\%\left(2\pi\mathrm{r_{1i}}\right) + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{Si}T})\%\left(2\pi\mathrm{r_{2i}}\right)\right)$$

or

One-time encryption key and decryption key at time T for ECC

$$= \mathrm{ECC}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{mi}T})\%(2\pi\mathrm{r_{1i}}) + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{Si}T})\%(2\pi\mathrm{r_{2i}})\right)$$

If angular velocities are used for one-time encryption key and decryption key, then:

One-time encryption key and decryption key at time T for RSA $= \mathrm{RSA}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{mi}}}{2\pi\mathrm{r_{1i}}} + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{Si}}}{2\pi\mathrm{r_{2i}}}\right)$

And/or

One-time encryption key and decryption key at time T for ECC $= \mathrm{ECC}\left(\frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{mi}}}{2\pi\mathrm{r_{1i}}} + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{Si}}}{2\pi\mathrm{r_{2i}}}\right)$

where RSA( ) is the RSA function for generating encryption key and decryption key with the passphrase, and ECC( ) is the ECC function for generating encryption key and decryption key with the passphrase, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN and N_T is the number of nodes in the FNN at time T.

16. The system as claimed in claim 15, wherein, the motor cores or the sensory cores are used respectively for generating a private key compatible with ECC:

$$Concatenate\ [\ (v_{m1}T)\%(2\pi r_{11}), (v_{m2}T)\%(2\pi r_{12}), \dots\ , (v_{m8}T)\%(2\pi r_{18})\ , \frac{1}{8}\sum_{i=1}^{8}(v_{mi}T)\%(2\pi r_{1i})]$$

or

$$Concatenate\ [\ (v_{s1}T)\%(2\pi r_{21}), (v_{s2}T)\%(2\pi r_{22}), \dots\ , (v_{s8}T)\%(2\pi r_{28})\ , \frac{1}{8}\sum_{i=1}^{8}(v_{si}T)\%(2\pi r_{2i})]$$

where vm are the velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN; or

Combining the motor cores with the sensory cores for generating a private key compatible with ECC:

$$Concatenate\ [\ (v_{s1}T)\%(2\pi r_{21}), (v_{m1}T)\%(2\pi r_{11}), (v_{s2}T)\%(2\pi r_{22}), (v_{s3}T)\%(2\pi r_{23}), (v_{m2}T)\%(2\pi r_{12}),$$

$$(v_{s4}T)\%(2\pi r_{24}), (v_{m3}T)\%(2\pi r_{13}), (v_{m4}T)\%(2\pi r_{14})\ , \frac{1}{16}\sum_{i=1}^{8}\{(v_{si}T)\%(2\pi r_{2i}) + (v_{mi}T)\%(2\pi r_{1i})\}\ ]$$

where vm are the velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN; and

combining the motor cores with the sensory cores for generating a private key compatible with RSA:

$$Concatenate\ [\ (v_{m1}T)\%(2\pi r_{11}), \dots, (v_{m24}T)\%(2\pi r_{1\,24}), (v_{s1}T)\%(2\pi r_{21}), \dots, (v_{s24}T)\%(2\pi r_{2\,24})]$$

where vm are the velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN; or

combining the motor cores with the sensory cores and a random generator for generating a private key compatible with RSA:

$$Concatenate\ [\ (v_{m1}T)\%(2\pi r_{11}), \dots, (v_{m8}T)\%(2\pi r_{18}), \frac{1}{8}\sum_{i=1}^{8}(v_{mi}T)\%(2\pi r_{1i}),$$

$$(v_{s1}T)\%(2\pi r_{21}), \dots, (v_{s8}T)\%(2\pi r_{28}), \tfrac{1}{8}\sum_{i=1}^{8}(v_{si}T)\%(2\pi r_{2i}),\ \ random_generator(1920\ bits)]$$

where the “random_generator(bits)” is a function for generating 1920 bits of digits randomly, vm are the

velocities of the motor core of the nodes in the FNN, r1 are the radiuses of the motor core of the nodes in the FNN, vs are the velocities of the sensory core of the nodes in the FNN, r2 are the radiuses of the sensory core of the nodes in the FNN, v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN.

17. An encryption method comprising:

Generating a Functional Neural Network or KeyNode (KN) of a system by programming a chain of multiple nodes also called Artificial Mirror Neurons (AMN) based on captured information of reaction time and emotional response to a task; racing the nodes in the Functional Neural Network or KeyNode (KN) as an encryption device or cipher for a time of use;

Classifying KeyNodes of multiple nodes where each KeyNode comprises multiple nodes held in a chain which can be generated successively, elongated, and can also evolve by retraction of nodes or decay after a certain number of nodes has been generated;

Encoding KeyNodes based on infinite proposition captured by interactive cognitive computing, forming an irregular tetragon matrix, and allowing dynamic encoding of unusual form of cryptography;

Generating the boundary of the KeyNode, where a minimum size is defined as A, the maximum size is defined as B, and a time of maximum size of the KeyNode is defined as C = B – (A – 1);

Generating one-time-private key for decryption at the user's time of use based on the concatenation of intrinsic values of the nodes in a racing network at this time and adopting the generated private key to generate a one-time public key to form a one-time-pair of keys for encrypted communication;

Transmitting the one-time public key over any unsecured channel to selected senders for encrypting a plaintext to become a ciphertext;

Decrypting the ciphertext of senders at receiver's end using the one-time key to regenerate plaintext data; or using the one-time key for symmetric key encryption of user's personal data for secure storage.

18. The method as claimed in claim 17, wherein the public key and private key are the same by using symmetric key encryption.

19. The method as claimed in claim 17, wherein symmetric key encryption is Advanced Encryption Standard ("AES").

20. The method as claimed in claim 19, wherein AES comprises AES-128, AES-192, and AES-256.

21. The method as claimed in claim 20, if theta angles are used in summation for one-time key

$$\text{One-time key at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{mi}T)\%\left(2\pi r_{1i}\right) \; + \; \frac{1}{N_T}\sum_{i=0}^{N_T}(v_{Si}T)\%\left(2\pi r_{2i}\right)$$

If angular velocities are used in summation for one-time key

$$\text{One-time key at time T} \quad = \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{mi}}{2\pi r_{1i}} \; + \; \frac{1}{N_T}\sum_{i=0}^{N_T}\frac{v_{Si}}{2\pi r_{2i}}$$

where v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN and N_T is the number of nodes in the at time T.

22. The method as claimed in claim 19, if theta angles are used in concatenation for one-time key, then:

$$\text{One-time key at time T} \quad =$$

$$Concatenate\ [\ (v_{m0}T)\%\left(2\pi r_{10}\right), \dots, \left(v_{mN_T}T\right)\%\left(2\pi r_{1N_T}\right), \dots, (v_{s0}T)\%\left(2\pi r_{20}\right), \dots, \left(v_{sN_T}T\right)\%\left(2\pi r_{2N_T}\right)]$$

If angular velocities are used in concatenation for one-time key, then:

$$\text{One-time key at time T} = Concatenate\ [\ \frac{v_{m0}}{2\pi r_{10}}, \dots, \frac{v_{mN_T}}{2\pi r_{1N_T}}, \dots, \frac{v_{S0}}{2\pi r_{20}}, \dots, \frac{v_{SN_T}}{2\pi r_{2N_T}}]$$

23. The method as claimed in claim 20, wherein AES-128 using a 128 bit key, AES-192 using a 192 bit key, and AES 256 using a 256 bit key respectively to encrypt and decrypt a block of messages, wherein there are 10 rounds of processing 128 bit keys, 12 rounds of processing 192 bit key, 14 rounds of processing for 256-bit keys, respectively.

24. The method as claimed in claim 21, wherein AES-128 comprises the following steps:

Breaking the message from the sender to the receiver into 128-bits' blocks;

Organizing the bytes in a given block as a 4×4 matrix;

Passing the block through the following sub-steps for a number of times depending on the size of the cipher key, i.e.10 times for AES-128:

Substituting the bytes by mapping each element in the matrix to the corresponding byte in the Rijndael S-box;

Shifting rows by rotating each element x bytes to the left, whereas x is the index of the row;

Mixing columns by multiplying every column by a predefined matrix according to the following algorithm:

if any term is greater than 2 to the power of 8, dividing the polynomial by the Galois irreducible polynomial as

$$x^8+x^4+x^3+x+1$$

Adding a round key by performing a bitwise XOR operation between the columns of the matrix and the Round Key; although in the first round, the Round Key which is de facto the one-time-key generated by KeyNode, is the first 128 bits of the cipher key;

Inputting the preceding matrix to the next round and repeating the sub-steps, minimum 10 times for an AES-128;

Excluding the final step of mixing columns for the final round.

25. An encryption system comprising:

KeyNode Generation module, a Functional Neural Network or KeyNode (KN) of a system by programming a chain of multiple nodes also called Artificial Mirror Neurons (AMN) based on captured information of reaction time and emotional response to a task are generated; the nodes in the Functional Neural Network or KeyNode (KN) are racing, and their valued output serves as encryption device or cipher for a time of use;

KeyNodes classification module, where each KeyNode comprises multiple nodes held in a chain which can be generated successively, elongated, and can also evolve by retraction of nodes or decay after a certain number of nodes has been generated;

KeyNodes encoding module, KeyNodes based on infinite proposition captured by interactive cognitive computing, forming an irregular tetragon matrix, and allowing dynamic encoding of unusual form of cryptography are encoded;

Boundary generation module, the boundary of the KeyNode, where a minimum size is defined as A, the maximum size is defined as B, and a time of maximum size of the KeyNode is defined as C = B – (A – 1) is generated;

Pair of keys generation module, one-time-private key for decryption at the user's time of use based on

the concatenation of intrinsic values of the nodes in a racing network at this time and adopting the generated private key to generate a one-time public key to form a one-time-pair of keys for encrypted communication are generated;

Key transmission module, the one-time public key over any unsecured channel to selected senders for encrypting a plaintext to become a ciphertext and using symmetric key encryption for encryption of user's personal data for secure storage is transmitted;

Decryption module, the ciphertext of senders at receiver's end using the one-time private key to regenerate plaintext data is decrypted or the one-time key for symmetric key encryption of user's personal data for secure storage is used.

26. The system as claimed in claim 25, wherein the public key and private key are the same by using symmetric key encryption.

27. The system as claimed in claim 25, wherein symmetric key encryption is Advanced Encryption Standard ("AES").

28. The system as claimed in claim 27, wherein AES comprises AES-128, AES-192, and AES-256.

29. The system as claimed in claim 28, if theta angles are used in summation for one-time key, then:

$$\text{One-time key at time T} \quad = \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{mi}T})\%\left(2\pi\mathrm{r_{1i}}\right) \; + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}(\mathrm{v_{Si}T})\%\left(2\pi\mathrm{r_{2i}}\right)$$

If angular velocities are used in summation for one-time key, then:

$$\text{One-time key at time T} \quad = \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{mi}}}{2\pi\mathrm{r_{1i}}} + \frac{1}{\mathrm{N_T}}\sum_{i=0}^{N_T}\frac{\mathrm{v_{Si}}}{2\pi\mathrm{r_{2i}}}$$

where v_{mi} is the velocities of the motor core of the *i*-th node in the FNN, r_{1i} is the radius of the motor core of the *i*-th node in the FNN, v_{si} is the velocities of the sensory core of the *i*-th node in the FNN, r_{2i} is the radius of the sensory core of the *i*-th node in the FNN and N_T is the number of nodes in the at time T.

30. The system as claimed in claim 27, if theta angles are used in concatenation for one-time key, then:

$$\text{One-time key at time T} \quad =$$

$Concatenate\ [\ (v_{m0}T)\%(2\pi r_{10}), \dots, (v_{mN_T}T)\%(2\pi r_{1N_T}), \dots, (v_{s0}T)\%(2\pi r_{20}), \dots, (v_{sN_T}T)\%(2\pi r_{2N_T})]$

If angular velocities are used in concatenation for one-time key, then:

One-time key at time T = $Concatenate\ [\frac{v_{m0}}{2\pi r_{10}}, \dots, \frac{v_{mN_T}}{2\pi r_{1N_T}}, \dots, \frac{v_{s0}}{2\pi r_{20}}, \dots, \frac{v_{sN_T}}{2\pi r_{2N_T}}]$

31. The system as claimed in claim 28, wherein AES-128 using a 128 bit key, AES-192 using a 192 bit key, and AES 256 using a 256 bit key respectively to encrypt and decrypt a block of messages, wherein there are 10 rounds of processing 128 bit keys, 12 rounds of processing 192 bit key, 14 rounds of processing for 256-bit keys, respectively.

32. The system as claimed in claim 29, wherein AES-128 comprises the following modules:

Message breaking module, the message from the sender to the receiver into 128-bits' blocks is broken;

Bytes organization module, the bytes in a given block are then organized as a 4×4 matrix;

Block pass module, the block through the following sub-steps for a number of times depending on the size of the cipher key, i.e. 10 times for AES-128:

Substitution module, the bytes by mapping each element in the matrix to the corresponding byte in the Rijndael S-box are substituted;

Row shifting module, rows are shifted by rotating each element x bytes to the left, whereas x is the index of the row;

Column Mix module, columns by multiplying every column by a predefined matrix according to the following algorithm are mixed:

if any term is greater than 2 to the power of 8, dividing the polynomial by the Galois irreducible polynomial as

$$x^8+x^4+x^3+x+1$$

Key Addition module, a round key is added by performing a bitwise XOR operation between the columns of the matrix and the Round Key; although in the first round, the Round Key which is de facto the one-time-key generated by KeyNode is the first 128 bits of the cipher key;

Repeat module, the preceding matrix is input to the next round and the sub-steps, minimum 10 times for an AES-128 are repeated; and Excluding the final step of mixing columns for the final round.