

WO 2014/135233 A1



Published:

— *with international search report (Art. 21(3))*

Data Processing Technique

Technical Field

5 The present disclosure generally relates to data processing. In particular, the disclosure relates to a device for intrinsically safe and secure data processing. Without limitation, the disclosure relates to an architecture for such a device and a method of operating such a device.

Background

10 Data processing is a vital function in various areas including technology, private or business communication, and military. Data processing system have rapidly increased in both capability and complexity. As a result of the latter, the operational safety and the security of data or data processing has been compromised. Lacking security can arise due to system failure. A minute failure in one out of millions of hardware components, such as transistors, may induce a widespread malfunction of the system. Also, the number of lines of code designed for operating data processing systems has increased on the order of 100 million lines for modern operating systems and some applications suites. Such complexity almost certainly implies errors, which consequences remain unpredictable to both developers and users.

25 Security, which is also referred to as information security or data security in certain scenarios, relates to the ability of a device to defend information from unidentified and unauthorized access. Herein, the defending of information may include protecting the information. Access may include disclosure and/or modification.

30 Safety, which is also referred to as operational safety or operational reliability in certain scenarios, relates to the ability of a device to perform its functions under predefined conditions for a specified period of time. Herein, safety may include failure safety.

35 Lacking security can also arise due to an external attack of the system. The increased complexity increases the vulnerability of the system, e.g., due to data leakage or weak points of internal mechanisms that can be exploited by the external attack.

- 2 -

One approach is to retain legacy systems that have proven to run stable and secure. The approach, however, eschews rapid technological progress. Furthermore, the approach neglects the diminishing safety due to aging of the system components.

Summary

According to one aspect, a device for data processing is provided. The device comprises at least three functional units, each of which includes a processing component and a memory component.

The memory component may be coupled to the respective processing component.

Each of the functional units may include one or more processing components. The architecture may be an embedded microprocessor architecture. Each of the processing components may include a microprocessor.

The processing component may be adapted to perform instructions stored in the respective memory component. The processing component may include a processor based on Reduced Instruction Set Computing (RISC), e.g. an ARM processor.

The device may further comprise a connecting component adapted to selectively connect subsets of the functional units. Alternatively or in combination, the connecting component may be adapted to selectively disconnect the functional units. For example, all functional units that are currently not connected may be mutually galvanically separated. All interfaces of each of the currently disconnected functional units may be isolated. All lines of the currently disconnected functional unit may be potential-free or grounded. The functional units may be selectively connected and/or disconnected by means of at least one of field-effect transistors (FET) and microelectromechanical systems (MEMS).

The connecting component may be adapted to disconnect a functional unit by means of complete galvanic isolation. External connections of a disconnected functional unit may be short-circuited, e.g., separately for each disconnected functional unit. Alternatively or in addition, the connections may be grounded (which is also referred to as safe galvanic isolation).

- 3 -

The connecting component may include a light transmitter, fiber optics and a light receiver arranged to selectively connect the functional units via the fiber optics. The connecting component may be adapted to selectively disconnect the functional units by powering off at least one of the light transmitter and light receiver. Alternatively or
5 in addition, the connecting component may be adapted to selectively disconnect the functional units by obstructing a light path of the arrangement.

Alternatively or in addition, the connecting component may be adapted to wirelessly connect the functional units. The disconnection may be brought about by passively
10 shielding or actively perturbing the wireless connection.

The selectively connected functional units may be pairwise connected. The connecting component may be adapted to selectively connect functional units in pairs so that each connected functional unit is connected to only one other functional
15 unit.

The connected pairs may be disjoint at any point in time. In other words, no functional unit of a given pair of connected functional units may simultaneously be connected to a third functional unit. From the perspective of a connected functional
20 unit, there may be no more than one communication partner at a time. Each of the functional units may be adapted to activate no more than one network socket or communication port at once.

The connecting component may connect two functional units by means of circuit
25 switching on a physical layer. The circuit-switched connection on the physical layer may be combined with, and/or compatible with, a packet-switched communication protocol, e.g., on a data link layer or an application layer.

Each pair of connected functional units may be connected for one data transfer only.
30 The pair connection may be established at the beginning of the data transfer. The pair connection may be disconnected at the end of the data transfer. The data transfer and the selective connection between a pair of functional units may be uniquely related. One pair of connected functional units may be associated to each data transfer.

35 A pair of connected functional units may include a sending functional unit and a receiving functional unit. The sending functional unit and the receiving functional unit

- 4 -

may define a secured entity for the duration of the data transfer. The entity may be secured against external access during the data transfer by means of the pair connection. In a highly restrictive implementation, the connecting component may be adapted to connect no more than two of the functional units at a time.

5

The connecting component may be implemented by means of a crossbar switch. The pair connectivity may be represented by a triangular matrix (e.g., an upper or lower triangular matrix). The uniqueness of the connectivity may be represented by no more than one connection indicated for each column and for each row in the triangular matrix.

10

Each of the functional units may include at least one communication interface adapted for the selective connection via the connecting component. The interfaces may be connected by the connecting component. The interfaces may include at least one of an Inter-Integrated Circuit (I2C) bus interface, a Universal Serial Bus (USB) interface, a Serial Peripheral Interface (SPI), a Universal Asynchronous Receiver Transmitter (UART) interface or an Ethernet interface. Examples of a wireless connection include members of the set of IEEE 802.11 specifications.

15

20

There may be n functional units in the device. At a given time, p pairs of 2^*p functional units may be connected, wherein $2^*p \leq n$. A state of the pairwise connectivity may be represented by a binary relation, i.e., a two-dimensional relation on the set of functional units, such as the triangular matrix having p non-zero entries. There may be $(n \text{ over } 2^*p) * n! / 2^p$ different connectivity states. In general, a galvanic connectivity state may be represented by one or more disjoint subsets of the functional units. For a number of x_j subsets each having a size j , there may be $(n \text{ over } \sum_j j * x_j) * n! / (j!)^{x_j}$ different connectivity states. Therein, $x_2 = p$ may be the number of connected pairs. The x_j subsets of size j may be represented by x_j non-zero entries in a j -dimensional hypercube having an edge length equal to n . Only a $j!$ -th fraction of the hypercube has to be stored to uniquely represent all connected subsets of size j . Denoting the j axes of the j -dimensional hypercube by a_i , $i=1$ to j , the $j!$ -th fraction of the hypercube may be $1 \leq a_1 < a_2 < \dots < a_{j-1} < a_j \leq n$. For $j=2$, the hypercube is a matrix and the $j!$ -th fraction is half of a matrix, e.g., aforementioned triangular matrix. There may be one hypercube of dimension j for all connected subsets of size j . The entire connectivity state may be represented by a sequence of differently dimensioned hypercubes, i.e.: a matrix (as a 2-dimensional hypercube), a cube (as a 3-dimensional hypercube), 4-dimensional cube, etc. The largest possible

25

30

35

- 5 -

hypercube dimension may be n for the device comprising n functional units. The sequence of differently dimensioned hypercubes may be integrated into the hypercube having the largest dimension. Subdimensional hypercubes within the hypercube having the largest dimension, e.g. boundaries of hypercube, may
5 represent the differently sized connected subsets.

The functional units may be adapted for data transfer only via the connecting component. Each of the functional units may include a protocol stack implementing communication protocols for data transfer. The protocol stack may include a
10 Resources Control Layer (1_e). The Resources Control Layer may be functionally arranged between a Physical Layer (1) and a Data Link Layer (2) of the data transfer. The Resources Control Layer may be adapted to identify a peer functional unit, to verify the authenticity of the peer functional unit, and/or to verify the authorization of the peer functional unit for applying a Data Link Layer
15 communication protocol or for passing the data packet to the Data Link Layer of the protocol stack. Identity, authenticity and authorization of the peer functional unit may be assessed based on data received from the peer functional unit. Alternatively or in combination, identity, authenticity and authorization of the peer application may be assessed in a separate challenge-response communication between the connected
20 functional units.

Alternatively or in addition, the protocol stack may include a Physical Switching Layer (3_e). The Physical Switching Layer may be functionally arranged between a Network Layer (3) and a Transport Layer (4) of the data transfer. For sending a data
25 packet, the Physical Switching Layer may be adapted to include configuration data that defines a mode of operation for the connecting component. The configuration data may be included into a field of the data packet to be transferred.

The connecting component may be adapted to read the configuration data defining a mode of operation for the connecting component from a data packet received for
30 transfer via the connecting component. The connecting component may further be adapted to enter the mode of operation in accordance with the configuration data. The mode of operation may limit or define the number of mutually connected functional units for the transfer of the data packet.

35 Preferable, the peer functional unit may be identified based on a logical unclonable feature, optionally a meromorphic function.

- 6 -

Alternatively or in addition, the protocol stack may include an Object Control Layer (5_e). The Object Control Layer may be functionally arranged between a Session Layer (5) and a Presentation Layer (6). The Object Control Layer may be adapted to identify a peer application, to verify the authenticity of the peer application, and/or to
5 verify the authorization of the peer application providing the transferred data. Identity, authenticity and authorization of the peer application may be assessed based on the data received from the peer application. Alternatively or in combination, identity, authenticity and authorization of the peer application may be assessed in a separate challenge-response communication between the connected functional units.

10 Optionally, the received data packet is not passed on towards the Application Layer, if one or more of identity, authenticity and/or authorization of the peer application do not match one or more of identity, authenticity and/or authorization of the peer functional unit. The matching condition may be stored at the receiving functional unit.
15 Alternatively or in addition, a first signature may be included in, or entangled with, the data packet for identity, authenticity and/or authorization of the peer application. A second signature may be included in, or entangled with, the data packet for identity, authenticity and/or authorization of the peer functional unit. The first signature and the second signature may be entangled. The entanglement may include an exclusive
20 disjunction (XOR).

For sending a data packet, the first signature may be entangled with a data field of the data packet. The data field may be encrypted. The first signature may be a public key used for the encryption.

25 For sending a data packet, the Resources Control Layer may further be adapted to include the second signature. The second signature may be derived from a substantially irreproducible feature of the peer functional unit. The second signature may be included into the data packet to be transmitted.

30 A security anchor may be located within each of the functional units. The security anchor may include a non-volatile memory. The security anchor may store a private key of the respective functional unit. The Object Control Layer may be adapted to retrieve the private key from the security anchor. Access to the security anchor may
35 be physically restricted to the processing unit of the respective functional unit performing the Object Control Layer.

- 7 -

In one implementation, the Object Control Layer may provide the private key to the Presentation Layer, or any other higher layer, for decrypting at least a part of the received data packet using the private key. In another implementation, the Object Control Layer may be adapted to retrieve the private key from the security anchor
5 and to decrypt at least a part of the received data packet using the private key.

At least one of the functional units may be adapted to control the connecting component. A functional unit that at least currently controls the connecting component may be referred to as a supervising functional unit. Different functional
10 units may assume the role of the supervising functional unit. Preferably, there is only one supervising functional unit at a time. Controlling the connecting component may include initiating and/or terminating the connections between the non-supervising functional units.

15 A supervisor program may be encoded into the memory of the supervising functional unit. The supervisor program may define the data processing performed by the device. The supervisor program may be represented and/or stored as a table. A first column of the table may include a sequence of instructions. A second column of the table may include at least one of test data (e.g., to be applied to the processing unit)
20 and reference data (e.g., to be compared with a response to the test data) for testing the processing unit of a functional unit before and/or after performing one or more of the instructions. The test data or reference data may be stored in the table in association to instructions that use or require functionality that is tested.

25 The supervising functional unit may alternate between a monitoring state and a processing state. Alternately or in addition, each of the functional units may be adapted to assume different operational states. The functional units may be switchable between the different operational states, e.g., by the supervising functional unit. The different operational states may include two or more of a
30 disconnected state, a wake-up state, a stand-by state, a wait state, a sleep state, an active state and a power-down state.

The supervising functional unit may execute a single-user multi-tasking operating system. A first task may monitor at least one of the status of the tamper-proof zones
35 and the status of the functional units not controlling the connecting component (i.e., the non-supervising functional units). A second task may execute the supervising

- 8 -

program. The operating system may alternate between the first and second tasks. The operating system may be limited to execute no more than two tasks.

5 In a preferred implementation, no operating system is executed on the supervising functional unit. The processing component and the memory component of the supervising functional unit may be implemented by means of a Field Programmable Gate Array (FPGA). In the case of an implementation using a Field Programmable Gate Array (FPGA), any excess functionality of the FPGA is physically destroyed during production of the device.

10 Alternatively or in combination, the processing component and the memory component of any one of the functional units may be implemented by means of a Complex Programmable Logic Device (CPLD). The memory component of any one of the functional units may include portions implemented by EPROM and/or
15 EEPROM.

The supervising functional unit may alternate between a first logical state machine and a different second logical state machine. The first logical state machine may monitor at least one of the status of the tamper-proof zones and the status of the
20 functional units not controlling the connecting component, e.g., by comparing boundary signals of each of the functional units for tabulated (upper and/or lower) limits. A second logical state machine may execute the supervisor program.

25 The tamper-proof zone (which is also referred to a De-Militarized Zone, DMZ, and is technically different from the DMZ-terminology used in the context of computer networks) requires (a) encapsulation, (b) data split and (c) external monitoring.

30 Another one of the functional units not currently controlling the connecting component may be adapted to take control over the connecting component in response to a failure or an exception occurring at the functional unit currently controlling the connecting component. The supervisor program may be transferred to the memory component of the new supervising functional unit.

35 The supervising functional unit may be adapted to segment the supervisor program into instruction modules. The supervising functional unit may further be adapted to associate at least two of the other functional units to different instruction modules. Only one primary functional unit may be associated to each of the instruction

- 9 -

modules. A secondary functional unit may be associated to at least some of the instruction modules as a backup in case of failure of the associated primary functional unit. The associated functional module is adapted to execute the respective one or more instruction modules. The supervising functional unit triggers the execution.

In a variant of the device, a Real-Time Operating System (RTOS) is executed by the supervising functional unit.

Each of the non-supervising functional units may execute a different operating system. Operating systems executed on the non-supervising functional units may include a Real-Time Operating System (RTOS), a unixoidal operating system, and a single-user single-tasking operating system. For example, the single-tasking core of RISC OS may be executed. The association may depend on the requirements of the instruction module. For example, a time critical instruction module may be associated to a functional module executing a RTOS. One of the non-supervising functional units may be a system functional unit adapted for external data communication. Another one of the non-supervising functional units may be an application functional unit adapted for executing generic applications. The system functional unit may include the single-user single-tasking operating system. The application functional unit may include the unixoidal operating system, e.g., Debian Linux.

The supervising functional unit may communicate with the non-supervising functional units in a push-pull operation. The associated functional units may execute the instruction modules respectively received from the supervising functional unit. Upon reaching an input instruction in the instruction module, the functional unit may signal an input request (also referred to as pull signal) to the supervising functional unit. Upon reaching an output instruction in the instruction module, the functional unit may signal an output request (also referred to as push signal) to the supervising functional unit. Upon termination of the instruction module, the functional unit may signal a status response to supervising functional unit. In case of an incomplete termination, the status response may include an error code or an exception code. The error code or exception code may uniquely indicate an error handler routine. In case of completion of the instruction module, the status response may include a pull signal for a further instruction module.

- 10 -

The supervising functional unit may be adapted to switch from the monitoring state indicating no error for an associated functional unit to the processing state. The supervising functional unit may further be adapted to send the respective instruction module to the associated functional unit.

5

The supervising functional unit may further be adapted to switch from the processing state, when the associated functional unit has completed or terminated the transferred instruction module, to the monitoring state for the associated functional unit.

10

By periodically switching between the monitoring state and the processing state, the boundary signals may be periodically monitored. The boundary monitoring is also referred to as mandatory boundary monitoring, if it is periodically performed, e.g., triggered by a timer causing a non-maskable interrupt that executes the boundary monitoring. The boundary monitoring is also referred to as a boundary scan and differs from a conventional boundary scan in that the boundary monitoring does not use JTAG-functionalities and is not applied to a single component. The boundary monitoring is applied to the functional unit as a whole. No component inside the functional units shall include JTAG-functionality. The boundary monitoring may be limited to the interface of the functional units and/or the regular functionality of the functional units.

15

20

25

30

35

Alternatively or in combination, the supervising functional unit may switch to the monitoring state in relation to one or more non-supervising functional units that have completed or terminated the execution of the respective one or more instruction modules. Monitoring the boundary signals may include a boundary scan of the functional unit having completed or terminated the respective instruction module. The functional unit can be subjected to the boundary scan. The boundary scan applied to the functional unit may differ from conventional chip operational function testing in that the boundary scan is applied to the functional unit as a whole, e.g., irrespective of subunit included in the functional unit and implemented by separate chips. The sufficiency of the boundary scan of the functional unit can be a consequence of the complexity limit defined for each of the functional units. The functional unit is preferably subjected to the boundary scan during the monitoring state, since the functional unit is not performing any instructions of the supervisor program during the monitoring state. The supervising functional unit may return to

the processing state in relation to non-supervising functional units, for which the monitoring of boundary signals indicates no error.

The supervisor functional unit may be adapted to control the connecting component to connect a sending functional unit and a receiving functional unit upon receiving matching requests from the sending functional unit and the receiving functional unit. The sending functional unit is also referred to as an originator. The receiving functional unit is also referred to as a terminator. A peer-to-peer communication may be established between the sending and receiving functional units. On the transport layer, the communication may be stateless, e.g., using the User Datagram Protocol (UDP).

The matching requests may require the reception of both:

- (i) an output request from the sending functional unit, the output request indicating the receiving functional unit as recipient of the data to be transferred, and
- (ii) an input request from the receiving functional unit, the input request indicating the sending functional unit as originator of the data to be transferred.

The supervising functional unit may further be adapted to resend the instruction module previously sent to a first functional unit in response to receiving from the first functional unit an output request indicating a second functional unit as recipient that does upon completion of its instruction module not provide a matching input request indicating the first functional module as originator. Alternatively or in combination, the supervising functional unit may further be adapted to resend an instruction module previously sent to a first functional unit in response to receiving from the first functional unit an input request indicating a second functional unit as originator that does upon completion of its instruction module not provide a matching output request indicating the first functional module as recipient.

By resending the instruction module, an error occurring during the execution of the respective functional module can be resolved. A maximum number of repetitions per functional unit may be defined before the functional module is powered off by the supervising functional unit.

- 12 -

The functional units may be spatially divided from each other. The functional units may be separately arranged next to one another. Each of the functional units may include a dedicated power supply unit. Uniquely associating one power supply unit with one of the functional units can, e.g., improve power failsafe operation and/or side-channel tamper security. The power supply unit may be remote-controlled, e.g., by the supervising functional unit.

The at least three functional units and the connecting component are encapsulated into a demilitarized zone and/or a tamper-proof zone. The demilitarized zone may be secured at least against attacks using data communication means. The tamper-proof zone may be secured at least against physical attacks including intrusion.

Any data exchange between the at least three functional units may be encrypted. E.g., the at least three functional units may exchange data via encrypted channels. The encrypted channels may be switched via the connecting component.

Alternatively or in addition, each of the at least three functional units and the connecting component may be individually encapsulated into a demilitarized zone and/or a tamper-proof zone. The global and/or the individual zones may be configured for self-destruction in response to at least some types of attack. The encapsulated unit can be inaccessible for data transfer from outside the unit including communication via electromagnetic waves (to transfer data). The encapsulated unit can be is insusceptible to radio frequency electromagnetic radiation from outside the unit. The casing can enclose the encapsulated unit. The casing can be adapted to provide an electromagnetic shielding. The shielding can be a barrier made of conductive material. Alternatively or additionally, the shielding can be adapted to prevent a coupling of the encapsulated unit to electromagnetic waves. Details for implementing the demilitarized zone and/or the tamper-proof zone are provided in published International application WO 2013 004442 A1.

The disjoint sets of temporally connected functional units improve both security and safety. The supervising functional unit may be adapted to survey the data transfer by at least one of triggering the connecting component to connect the set of functional units involved in the transfer, monitoring output data, monitoring input data, and triggering the connecting component to disconnect all functional units involved in the transfer.

- 13 -

Prior to connecting a functional unit, the supervising functional unit may assess identity, authenticity and authorization of the requesting application and/or of the functional unit to be connected. The security anchor mentioned above may be used by the supervising functional unit for assessing identity, authenticity and
5 authorization of the requesting application. Alternatively or in addition, a substantially irreproducible feature such as a parameter of a chaotic map or a physical unclonable feature of the requesting functional unit may be used by the supervising functional unit for assessing identity, authenticity and authorization of the functional unit. The parameters of the chaotic map or the physical unclonable feature may be stored in a
10 token within the functional unit. The token may be a secured memory. Intrusion to the token may cause self-destruction of the content of the secured memory. The Mandelbrot map generating the Mandelbrot set is an example of a chaotic map for certain values of its parameter. The parameter may be used as the private key of the functional unit, which is never distributed outside of the functional unit. Bitmaps of
15 different sizes and different parts of the Mandelbrot set can be used as the signature.

The processing component and the memory component may also be referred to as examples of functional sub-units. Each of the at least three functional units may
20 including at least three functional sub-units, and at least two connecting sub-units, each of which is adapted to provide one link for data transfer. Each connecting sub-unit may be adapted to switchably couple at most two of the functional sub-units according to a switching state of the connecting sub-unit. The connecting sub-unit may be adapted to switchably couple the processing sub-unit and the memory sub-
25 unit. The connecting component may include at least two connecting units, each of which is adapted to provide one link for data transfer. The coupling by the connecting unit may provide unidirectional data transfer. Each connecting unit may be adapted to switchably couple at most two of the functional units according to a switching state of the connecting unit.

Each of the units and/or each of the sub-units may be encapsulated. A number of interfaces of the encapsulated unit may be limited to interfaces towards other units assigned to the encapsulated unit by the supervising functional unit. The other units may include connecting units or functional units other than the encapsulated unit.

The supervising functional unit may be adapted to set the switching states according to a limited number of connection states of the device, each connection state being

- 14 -

defined by the entirety of the switching states. The entirety of the switching states may include the switching states of all connecting units and all connection sub-units. The switchably coupling can further include a self-coupling of one or more of the functional units.

5

The limited number of connection states may be physically limited to exclude a coupling of two functional units and a simultaneous coupling of two functional sub-units of one of the two functional units. The limited number of connection states may be physically limited to exclude a simultaneous coupling of all functional units and/or all functional sub-units. The connection state may be the Cartesian product of all switching states.

10

Each of the switching states may comprise an on state, an off state, and an active state. The switchable coupling provides the galvanic decoupling. The supervising functional unit or at least one of the connecting units may further be adapted to verify data exchanged via the coupling.

15

The verification may include a comparison of at least one of the exchanged data and an entropy measure of the exchanged data with respect to an (upper and/or lower) limiting value.

20

The memory sub-unit of the supervising functional unit may be encoded by a representation of a graph. The graph may comprise nodes and edges. Each of the nodes may unambiguously represent one of the functional units and the functional sub-units, and wherein each of the edges unambiguously represents one of the connecting units and the connecting sub-units.

25

The graph may be a connected graph. Furthermore, the graph may be a directed graph. Still further, the graph can be a hypergraph. The graph may be an isomorphic representation of all units, e.g., including all sub-units.

30

Boundary conditions, e.g. including at least one of limiting values and boundary monitoring values, may be stored in a matrix representation of the graph. A reaction to an exception of each unit can be defined by a column of the matrix.

35

The functionalities provided by functional units may include one or more of: reading data, particularly reading image data; storing data, particularly storing image data;

- 15 -

communicating data, particularly communicating image data; controlling a scanner; particularly controlling a fingerprint scanner; processing data, particularly processing image data; and analyzing data, particularly analyzing fingerprint data. One of the functional units may be adapted to selectively couple to an external network such as the Internet.

The device may comprises a plurality of sensors, each of which is adapted to measure a physical quantity. The physical quantity may include one or more of: (local) temperature, (local) pressure, sound waves, vibrations, (gravitational) acceleration (e.g., including magnitude and direction), a change in capacity or inductance.

Some or all of the sensors may be implemented by a dedicated functional unit. Such a sensor can include a temperature sensor inside a casing of the device, a microphone in a casing of the device, or a proximity sensor. Alternatively or additionally, some or all of the of sensors can be sub-units. Such sensors can include a sensor connected to processor sub-unit of the same unit. Still further, some or all of the sensors can be implemented as sub-sub-units. Such sensors can include a temperature sensor, a scan cell, or test cell inside a processing sub-unit.

The proximity sensor can be a capacitive sensor. The capacitive sensor can generate an electrical field that fills the interior of a casing of the device. The electrical field can further extent (on one side of the casing) to the outside of the casing (with short range, preferably in the order of few millimeters) as a touch switch. A change in capacitance measured by the electrical field can indicate pressing the touch switch (as long as the casing is closed). Due to the electrical field filling the interior of the casing, opening the casing can cause a significant change in capacitance. In response to the significant change, the capacitive sensor can signal an exception (to the supervising functional unit).

The supervising functional unit may be adapted to perform the functionality of monitoring a device state, wherein the device state can indicate an exception. The supervising functional unit may externally monitor the non-supervising functional units. The supervising functional unit may be adapted to reset one or all other sub-units of a unit, preferably including a corresponding memory sub-unit, in response of the exception. The device state may comprise the connection state and a sensor

- 16 -

state. The supervising functional unit may be adapted to perform the functionality of controlling the connection state and of monitoring the sensor state of the device.

The supervising functional unit may be adapted to assign the functionality to the functional units. Each functional unit may be adapted to perform the functionality according the functionality assignment. The supervising functional unit may be adapted to shift the assigned functionality to another functional unit or to swap the functionality assignment among two of the functional units. The shifting or swapping may include the functionality of monitoring the device state. The supervising functional unit may be adapted to shift or to swap one or more functions in response to an exception. The connecting unit may be a functional unit, to which the supervising functional unit assigns the functionality of the switchable coupling.

The sub-units may further comprise sub-sub-units rendering the sub-units self-similar to the units. Alternatively or in combination, the units may be comprised in one or more super-units rendering the super-units self-similar to the units.

At least one of the sub-units, preferably comprising a microchip (such as the processing sub-unit or the memory sub-unit), may further comprises (as a sub-unit or sub-sub-unit) a scan cell providing a JTAG functionality or a test cell providing a built-in self-test. The JTAG functionality can be irreversibly disconnected upon production of the device. The complexity of each (functional or connection) unit can be low enough to allow for a complete functional verification of the corresponding unit.

The functionality of (one or all of) the functional units may be fully implemented in hardware of the functional unit, or may be fully implemented in software of the functional unit, or may be a combined implementation of hardware and software. The software may be inaccessible and/or not alterable from the outside of the unit. Particularly, one or all connecting units can be implemented as a software switch.

Brief Description of the Drawings

In the following, the disclosure will be described with reference to exemplary embodiments illustrated in the drawings, wherein:

Fig. 1 schematically illustrates a graph representation of an architecture of a device for data processing;

- 17 -

Fig. 2 schematically illustrates protocol stacks implemented at each functional unit in the device of Fig. 1;

Fig. 3 shows an example of a connection state of functional units in the device of Fig. 1;

Fig. 4a to 4f schematically illustrate connecting components for connecting the functional units of the device shown in Fig. 1;

Fig. 5 schematically illustrates a switching element as building block of the connecting components shown in Fig. 4a to 4f;

Fig. 6 schematically illustrates a bilateral communication between a pair of functional units;

Fig. 7 shows a flowchart of a method of verifying an identity of a peer entity performed by a challenging entity;

Fig. 8 shows a flowchart of a method of verifying an identity of a peer entity performed by a challenged entity;

Fig. 9 shows a schematic block diagram of an apparatus for performing the method of Fig. 8;

Fig. 10 schematically illustrates an exemplary dependency of a generate fractal pattern as a function of a parameter exchanged in the communication of Fig. 6;

Fig. 11 schematically illustrates an identification method based on a holomorphic or meromorphic function as a logical unclonable feature;

Fig. 12 schematically illustrates an identification method based on a complex function including an essential singularity as a logical unclonable feature;

Fig. 13 schematically illustrates an identification method based on a transfer function as a physical unclonable feature; and

- 18 -

Figs. 14 to 16 schematically illustrate an identification method based on phosphorescence as a physical unclonable feature.

Description of Preferred Embodiments

5 In the following, for purposes of explanation and not limitation, specific details are set forth, such as particular topologies, connectivities, interfaces and configurations, in order to provide a thorough understanding of the disclosure. It will be apparent to one skilled in the art that the disclosure may be practiced in other embodiments that
10 depart from these specific details. For example, while the embodiments will be described with reference to a portable USB device, it will be apparent to the skilled person that the invention can also be practiced in a stationary device and/or in context of other interface technologies, including wireless external connections. Furthermore, the invention can be implemented in a distributed manner by means of
15 a communication network and/or as part of a communication network. The concept of separated functional units, as detailed herein, may be implemented in the communication network such that some or all of the functional units form nodes of the communication network.

20 Moreover, those skilled in the art will appreciate that the functions and processes, in particular some or all of the functional units, explained herein may be implemented using software functioning in conjunction with a programmed microprocessor and/or using an Application Specific Integrated Circuit (ASIC) or a general purpose computer. It will also be appreciated that while the following embodiments will
25 primarily be described in context with methods and devices, the invention may also be embodied in a computer program product as well as in a system comprising a computer processor and a memory coupled to the processor, wherein the memory is encoded with one or more programs that may perform the services, functions and steps disclosed herein.

I. General Structure of a Device

With reference to Fig. 1, a general description of the architecture of a device 100 for data processing is provided.

I.1 Functional Subdivision and Complexity Limit

The device 100 has a functional capability, i.e., an overall function is defined for the device 100 in a supervisor program. As a first building principle, i.e., a guiding principle of the architecture of the device 100, the complexity of the device 100 is reduced. More specifically, the functional capability is subdivided into (or distributed to) a plurality of functional units 110, 120, 130 and 140. Each of the functional units 110, 120, 130 and 140 is adapted to perform a partial functionality of the functional capability of the device 100.

A number of functional units is chosen such that a complexity of each functional unit does not exceed a predefined complexity limit. The functional capability of the device 100 can be subdivided into (or distributed to) a single layer of functional units 110, 120, 130 and 140. Alternatively, as shown in Fig. 1, the functional capability can be distributed over two or more functional layers. Specifically, one or more functional units can be further subdivided into two or more functional subunits. In the embodiment of Fig. 1, the functional units 110 and 130 are further subdivided. The functional unit 110 is subdivided into subunit 112 and 114. The functional unit 130 is subdivided into functional subunits 132, 134, 136, 137, 138 and 139.

The functional units 110, 120, 130, 140 define a first functional layer of the device 100, and the subunits 132, 134, 136, 137, 138, 139 define a second functional layer of the device. While the embodiment illustrated in Fig. 1 has two layers, the functional capability of the device 100 can be subdivided three or more times. In other words, the subdivision can be iterated. A number of the iterations can correspond to a number of functional layers. As can be seen from Fig. 1, the number of subdivisions of different functional units can be different. The functional units 110 and 130 extend to both the first functional layer and the second functional layer. The functional units 120 and 140 are restricted to the first functional layer. A number of subunits per unit or the number of iterations of the subdivision is chosen such that the complexity of each subunit fulfills the complexity limit.

Units, subunits, and sub-subunits etc. are collectively referred to as components of the device 100. In Fig. 1, large circles represent components on the first functional layer, and small circles represent components on the second functional layer. Each component of the device 100 is required to fulfill the complexity limit. The complexity limit is determined by one or more complexity requirements. A first complexity

- 20 -

requirement is verifiability (also referred to as checkability). As a hardware perspective, verifiability encompasses the objective possibility (and practical feasibility) to verify, in finite time, that the respective component is free from defects. As a functional perspective, verifiability encompasses the objective possibility (and practical feasibility) to verify, in finite time, that the respective component operates free of error. The verification can be performed at the time of design, at the time of production, or during operation of the device 100. Alternatively or in addition, a second complexity requirement is that the respective component assumes only a finite number of (internal) states. Each of the components is encapsulated. In a preferred embodiment, the components are strictly separated from each other. The separated components can be implemented by spaced apart components. In sum, the components of the device 100 define a functional granularity, such that each component is encapsulated and fulfills the complexity limit.

I.2 Connectivity within the Device

Straight lines in Fig. 1 schematically illustrate a connectivity (also referred to a topology) of the components in the device 100. The lines represent a signal line, a data link or any other connection between the components within the device 100. In a preferred embodiment, all functional units are mutually connected. For a number of N functional units, there are at most $N(N-1)/2$ mutual connections. The embodiment of the device 100 shown in Fig. 1 comprises $N=4$ functional units 110, 120, 130, 140 with $N=6$ mutual connections 145, 150, 155, 160, 165 and 170. On a physical level, each of the connections is implemented by one or more unidirectional links. In the preferred embodiment, the connection comprises a pair of two unidirectional links having opposite directions (configured for bidirectional communication). As discussed in more detail in Subsection I.3 and Section III, connections within the same layer use one communication protocol.

In the presence of subunits (and sub-subunits, etc.), the connectivity of the device 100 defines (or corresponds to) a hierarchical structure of the components. One or more of the subunits are assigned to one (and only one) of the functional units by a sub-connection (between the subunit and the functional unit, to which the subunit is assigned). The sub-connection thus defines the assignment, which extends across different functional layers. In Fig. 1, the subunits 112 and 114 are assigned to the functional unit 110 by sub-connections 190 and 191, respectively. The subunits 132, 134 and 136 are assigned to the functional unit 130 by sub-connections 192, 193

- 21 -

and 194, respectively. Each of the assigned subunits provides its sub-functionality to the functional unit, to which the subunit is assigned.

Similarly to the connectivity in the first functional layer, two or more subunits can be
5 connected provided that the two or more subunits are assigned to the same functional unit. In Fig. 1, the subunits 136, 137, 138, 139 have mutual connections 180, 181, 182, 183, 184 and 185. In general, the hierarchical structure of components can be self-similar. In the embodiment of the device 100 shown in Fig. 1, a tetrahedron connectivity of the subunits 136, 137, 138, 139 on the second
10 functional layer resembles a tetrahedron connectivity of the functional units 110, 120, 130, 140 on the first functional layer.

As a second building principle of the connectivity in the device 100, connections within the device 100 are physically reduced to a minimum. More specifically, the
15 connectivity prevents (i.e., renders physically impossible) any direct access across different layers (without assignment). As an example indicated by dashed lines 195 and 196 in Fig. 1, a connection 195 from a first unit (such as the unit 110) to a subunit (such as the subunit 132) assigned to a second unit (such as the unit 130) that is different from the first unit is excluded by the connectivity. Furthermore, the
20 connectivity excludes any direct connection 196 from a first subunit (such as the subunit 112) to a second subunit (such as the subunit 132), if the first subunit and the second subunit are assigned to different functional units.

I.3 Unique Addresses

Each component is addressable by a unique address. The device 100 applies a single global system of addresses.

I.4 System Representation by Hypergraphs

As explained above in Subsections I.1 and I.2, the device 100 comprises components and connections, respectively. The connectivity of the device 100 is represented by a graph. The graph comprises vertices (also referred to as nodes), each of which represents one of the components of the device 100, and edges (also
35 referred to as lines), each of which represents one of the connections of the device 100.

- 22 -

Graphs represent pairwise relations between objects. The embodiment of the device 100 shown in Fig. 1 is representable by a graph defining pairwise relations between objects, namely the connections between the components. In general, the connectivity of the device 100 is not limited to pairwise relations. Rather, the connectivity of the device 100 is represented by a hypergraph. A hypergraph comprises vertices and hyperedges. While an edge represents a relation of a pair of vertices, a hyperedge represents a relation of a (general) set of vertices.

As a result of the hypergraph representation, each of the connections is representable by one vertex of the hypergraph, and each of the components is representable by one hyperedge of the hypergraph. In other words, the role of components and connections can be interchanged by virtue of the hypergraph representation. As an example, the embodiment of the device 100 shown in Fig. 1 has a complete mutual connectivity in the first layer. There are $N_1=4$ functional units in the first layer, and $N_1-1=3$ connections end at each of the functional units has. In the hypergraph representation of the connectivity, the functional unit 120 is represented by a hyperedge, namely the set of $N_1-1=3$ connections 145, 150, 155.

Referring back to the complexity limit introduced above in Section I for the components of the device 100, the complexity limit is defined by one or more complexity requirements. As a more detailed complexity requirement, storage and logic in the components are separated such that a boundary monitoring is not history dependent. Additionally, the boundary monitoring is required to have access to each subcomponent in the component without conventional boundary scan latch cells. Additionally, correctness of a component layout has to be confirmable by at least one of analytical means or a computer-implemented algorithm adapted to parse a (digital) representation of the component's design.

II. Communication Protocols

Specific protocol layers are implemented in addition to at least one conventional protocol layer, e.g., one or more layers of a conventional Open Systems Interconnection (OSI) model or the Internet protocol suite. Fig. 2 schematically illustrates protocol stacks 210, 220 including the layers 1 to 7 of the conventional OSI protocol stack. In an embodiment of the device, each of the functional units includes a protocol stack 210 or 220 having additional layers that are absent in the conventional OSI protocol stack.

- 23 -

An example of three additional layers 1e, 3e and 5e is schematically illustrated in Fig. 2 for a pair of connected functional units 110 and 120. The protocol layers 1e and 5e mutually secure identity, authentication and authority of the connected functional units as communication terminals. The protocol layer 3e provides
5 configuration parameters to a connecting component 240 in addition to the master control of the supervising functional unit over connecting component 240.

A data packet 230 is received on Physical Layer 1 of the stack 210. The received data packet includes a signature of a substantially irreproducible feature of the peer.
10 For an internal communication illustrated in Fig. 2, the peer is the sending functional unit. In one implementation, the signature is a response provided by the token of the peer based on a challenge previously sent by the receiving functional unit to the sending functional unit. The substantially irreproducible feature of the peer is a function that outputs the signature given the challenge. In another implementation,
15 the substantially irreproducible feature is a physical unclonable feature or a mathematical algorithm generating high-entropy results for a wide range of parameters, such as a Mandelbrot set or any other fractal (as discussed below). The mathematical function is substantially non-invertible with respect to the parameter set for a given (finite) portion of the result generated for a parameter instance. A first
20 portion denoted by domain_puf_B' of the result is used as the signature of domain B, i.e., the functional unit 120. A second portion denoted by domain_puf_B that at least overlaps with the first portion has been previously provided from the sending functional unit 120 to the receiving functional unit 110 via a secure channel. Preferably, the second portion fully includes the first portion.

25 The second portion or any other reference signature generated by the substantially non-invertible function using the unexchanged parameter is secretly provided to the receiving Resources Control Layer. The exchanged data includes at least the second portion of the result. The identifying, verifying of the authentication and the
30 verification of the authorization are positively assessed based on a match between the received first portion and the stored second portion.

The first portion thus serves as a signature allowing uniquely identifying the hardware of the peer functional unit 120. If and only if the signature matches or
35 corresponds to the expected response to the "challenge", the extended layer Resources Control Layer 1e passes the data packet to the Data Link Layer 2 and further to the Network Layer 3.

- 24 -

On the extended Physical Switching Layer 3e, the correctness of the connecting component as to the correct hardware and functionality (e.g., absence of a modifying external access to the transferred data) is accessed. For the case of an external communication (i.e., the peer is not another functional unit within the device), the extended Physical Switching Layer 3e also verifies the correctness of the software implementing the routing of data packets. The extended Physical Switching Layer 3e authenticates the communication route, also in an open network such as the Internet.

Only in the case of positive assessment as to the connecting component 240 (in the case of an internal communication) or any other communication route (in the case of an external communication), the data packet 230 is passed on to the higher Transport Layer 4 and the Session Layer 5. An extended Object Control Layer 5e receives the encrypted data field of the data packet and provides a key for decrypting the data packet on the presentation layer 6. The key for the decryption is stored in a security anchor within the extended Object Control Layer 5e.

The security anchor is provided by the local user. For example the security anchor is implemented by means of a storage device (such as a card reader or memory stick) entered by the user. The security anchor can also be implemented by a passphrase (such as a gesture input at a touchscreen). As a result of the decryption performed on the Presentation Layer 6 in conjunction with the extended Object Control Layer 5e, plain text is provided to the Application Layer 7.

In one embodiment, the extended layers 1e, 3e and 5e are implemented at the (current) supervising functional unit in a centralized manner for all non-supervising functional units. In another embodiment, all protocol layers, which do not necessarily include all the layers shown in Fig. 2, are locally implemented in each of the functional units.

The functional unit further verifies an entanglement 250 between the peer hardware identified by the extended Resources Control Layer 1e and the application identified by the extended Object Control Layer 5e. The entanglement 250 is defined in one embodiment for the duration of a transfer. In another embodiment, the entanglement 250 between peer hardware and application is defined separately for each data packet 230.

- 25 -

The data processing defined by the supervisor program (for the entire device 100) and/or the instruction module (associated to the individual non-supervising functional units) can be implemented on at least one of the layers 6 and 7.

- 5 The same protocol stack can be implemented for both an external communication and an internal communication (i.e., within the device 100).

10 Layer 3e provides an intermediate layer or neutral layer for abstracting functionality added after launching the kernel. For example, a generic application can be launched into the application functional unit and operated in encapsulation. The operation of the generic application is externally monitored by the supervisor. The monitoring uses the push-pull procedure described above.

15 A layer below the layer 3e, e.g., layer 1e or 3, is adapted to trigger an exception responsive to a physical interruption of the connection between the functional units. The layer 3e allows physically separating the link without causing an exception. For example, the layer 1e or 3 includes a routine implementing the exception handling (exception handler). A pointer for the exception handler and the exception handler are resident in the memory component. For decoupling the link, the exception pointer is masked and the exception handler remains resident. The Physical Switching Layer
20 sets a switching address in the switching matrix. No delay occurs, e.g., caused by loading or executing the exception handler, to the benefit of a high switching rate.

III. Switches of the Connectivity

25 Fig. 3 shows a hypermatrix representation 300 of a temporary connection state between the functional units 110, 120, 130 and 140 within the device 100. The example shown in Fig. 3 has $j=3$ dimension in order to represent three-party connections. Connections are indicated by dark nodes 310, 320 and 330. The
30 connected subsets of size $j=3$ are disjoint if and only if there only one connection on each of the j axes. Therefore, the connection 310 is marked as cancelled in favor of the connection 320. Furthermore, the configuration space of connections has to be restricted to a corner having $1/j!$ of the size of the full hypermatrix in order to eliminate redundancy due to index permutations. The dimension j is thus the number
35 of mutually connected functional units out of the n (with $n \geq j$) functional units.

- 26 -

The connectivity is realized by means of the connecting component 240. The connection state of the connecting component 240 is represented by the hypermatrix 300.

5 The switchable data lines can be centralized in one or few switching networks 400. Figs. 4a to 4d schematically illustrate embodiments of the switching network 400. The switching network 400 defines a (global) connectivity between the plurality of functional units 110 to 140. The functional units 110 to 140 are consecutively numbered. E.g., "FU1" may refer to the supervising functional unit 110, "FU2" may
10 refer to the application functional unit 120, etc.

Figs. 4a, 4b and 4c show embodiments of the switching network 400, each of which has a matrix structure that is adapted to couple or decouple each pair of input (IN_{FU1} , IN_{FU2} , ...) and output (OUT_{FU1} , OUT_{FU2} , ...) of the functional units. The coupling is
15 illustrated by circles 402 to 410. The coupling provides a direct electrical coupling, which state is also referred to as an enabled state of a respective data line 412. The decoupling provides a galvanic isolation, which state is also referred to as a disabled state of a respective data line 414.

20 In the embodiment illustrated in Fig. 4b, the switching network grounds all data lines that are in the disabled state. The grounding is preferably included in the switching network. I.e., ground lines 416, 418 are part of the switching network 400. In a variant thereof, all data lines in the disabled state are set to another reference potential. In the embodiment illustrated in Fig. 4c, all data lines in the disabled state
25 are connected to a short-cut line 420.

Alternatively or in addition, the switching network comprises switches 450, 460. The switches 450 and 460 are in-line. I.e., there is at least one switch 450 or 460 in the respective data line 412 or 414. The switch 450 is integrated between a fixed pair of
30 the functional units. The switch 450 only sets the data line 412 to the reference potential 424 in the disabled state, as is shown in Fig. 4d. As opposed to that, those functional units that participate in a flexible connectivity are connected to switches 460 shown in Fig. 4e. The switch 460 provides an additional galvanic isolation in the disabled state of the reference potential line 424. In other words, a functional unit
35 coupleable to a (fixed) predefined functional unit has at least one of its input and its output connected to the switch 450, which only sets the data line to the reference potential in the disabled state. A functional unit participating in the (fully flexible)

- 27 -

connectivity of the device 100 is connected to the switch 460 or the switching network 400, which sets the data line to the reference potential in the disabled state and provides galvanic isolation (between input and output). The reference potential may be ground (0 V), a maximum signal voltage (e.g., 5 V) or any constant voltage level in-between.

The connecting component 240 includes at least one of the switching network 400 and the individual switches 450 and/or 460.

The connectivity is not static but controllable. A single one of the functional units, which is referred to as a supervising functional unit, is adapted to control the connectivity. I.e., a status 422 of the individual switches 450, 460 and the entire switching network 400 is controlled by the supervising functional unit only. There is only one supervising functional unit (at the same time) in the device 100. In an advanced embodiment of the device 100, the role of the functional unit is interchangeable between the functional units 110 to 140. In a preferred embodiment of the device 100, one functional unit is predetermined to operate as the supervising functional unit 110. All other functional units 120 to 140 are referred to as executing functional units.

In a more cost-effective embodiment shown in Fig. 4f, the switching network 400 is adapted to forward only one signal at the same time. The latter embodiment is preferably implemented by means of a pair of a multiplexer ("MUX") and a demultiplexer ("DEMUX") as shown in Fig. 4f. As opposed to a conventional pair of multiplexer and demultiplexer (e.g., in communication technology), in the switching network 400 shown in Fig. 4d, the "multiplexer" and "demultiplexer" are not controlled so as to synchronously select the same "channel", but to select the two (generally different) functional units to be coupled. In the instance shown in Fig. 4f, the functional units FU2 and FU4 are connected.

The input line 414 to the functional unit FU4 is an example for a galvanically isolated data line. The status specifying one output data line and one input data line is indicated by reference sign 422.

Any variant of connecting component 240 can be realized as part of the supervising functional unit 110 or as an independent one of the functional units.

- 28 -

Fig. 5 shows a more advanced building block 500 of a connecting component 240 for connecting functional units 110 and 120. The interfaces 510 and 520 of the functional units 110 and 120, respectively, include to galvanic lines, which are connected and disconnected according to the configuration data 530 read by the connecting component 240 from the data packet 230 in addition to the master control of the supervising functional unit 130.

IV. Verification of Peer Identity

Fig. 6 schematically illustrates a communication 600 for peer identification. The functional units 110 and 120 communicate for bilateral identification.

The functional unit 110 verifies the identity of the functional unit 120 by means of a fractal. Each of the functional units 110 and 120 includes an equivalent fractal generator 610 and 620, respectively. Each of the fractal generators 610 and 620 is adapted to generate a fractal pattern based on one or more parameters. The functional unit 120 further comprises a random filter 622 adapted to modify the fractal pattern generated by the fractal generator 620.

The functional unit 110 further comprises an invariant filter 612 adapted to compute an invariant based on the modified fractal pattern. The invariant is independent of the modification brought about by the random filter 622. The invariant depends on the fractal pattern, which in turn depends on the one or more parameters.

Fig. 7 shows a flowchart of a method 700 for verifying an identity. The method 700 can be performed by the functional unit 110 for identifying a peer functional unit 120, e.g., in the context of the communication 600.

In a step 702 of the method 700, a random parameter is generated and sent to the peer functional unit 120. A first fractal pattern is generated based on the parameter in a step 704.

A further pattern is received from the peer functional unit 120 in a step 706. A first invariant is computed in a step 708 based on the first fractal pattern. In a step 710, a second invariant is computed based on the received pattern.

- 29 -

The first invariant and the second invariant are compared in a step 712. A match of the first and second invariants indicates the correctness of the identity for the peer functional unit 120.

5 The step 704 can be performed by the fractal generator 610 of the functional unit 110. The one or more parameters include an initial value for the computation of a fractal pattern, e. g., by the fractal generators 620. The one or more parameters may further specify a section and/or a resolution for a bitmap. The bitmap is an approximate representation of the fractal pattern. The bitmap may be an
10 approximation in that it is an incomplete representation of the fractal pattern as to size and resolution. The bitmap specifies format and extent of the fractal pattern received from the peer functional unit 120 in the step 706.

Fig. 8 shows a flowchart of a method 800 for verifying an identity. The method 800
15 can be performed by the functional unit 120, e.g., in the context of the communication 600.

In a step 802 of the method 800, one or more parameters for a fractal generator are received from the peer functional unit 110. A fractal pattern is generated based on
20 the received parameter in a step 804. The generated pattern is subjected to a random filter in a step 806. The pattern modified by the random filter is sent to the peer functional unit 110 in a step 808.

The fractal generator 620 performs the step 804. The random filter 622 performs the
25 step 806.

The invariant filter 602 of the functional unit 110 computes a scalar quantity that is independent of the modification caused by the random filter 622 and highly dependent on the fractal pattern generated by the fractal generator 620 based on the
30 parameter. Accordingly, the output of the invariant filter is sensitive to the fractal generator 620 being equivalent to the fractal generator 610. If the first invariant computed in the steps 708 corresponds to the second invariant computed in the step 710, the functional unit 110 has verified the identity of its peer functional unit in the current communication 600 as being the functional unit 120.

35 The fractal generator 620 used by the functional unit 120 for responding in the communication 600 thus identifies the functional unit 120. Each of the functional

- 30 -

units 110 to 140 has a unique fractal generator co-located with the respective functional unit. The fractal generators 610 and 620 are preferably implemented by means of a Field-Programmable Gate Array (FPGA).

5 In a first embodiment of the architecture 100, each of the functional units 110 to 140 has a fractal generator representing its own identity (i.e., an identifying fractal generator, e.g., used when acting as the challenged entity). Each of the functional units 110 to 140 further includes one fractal generator for each of the other functional units for verifying their peer identity (e.g., when acting as the challenging entity).

10 In a second embodiment of the architecture 100, each of the functional units 110 to 140 has a fractal generator for its own identity. Each of the functional units 110 to 140 further includes memory storing a pair of parameter and invariant corresponding to the parameter for each of the other functional units. Latter embodiment has the advantage that a successful temper attack against one of the functional units does
15 not provide means for impersonating the temper functional unit, because it is impossible to derive the fractal generator, e.g., the fractal generating algorithm, from the limited number of one or more pairs including a parameter value and an invariant value.

20 The random filter 622 introduces a distraction dynamics into the communication 600, so that eavesdropping the communication 600 does not even allow guessing the fractal generator based on the input parameter and the output bitmap, because the random filter introduces an additional dependency into the bitmap, which the
25 eavesdropper cannot distinguish from the non-linear dependency of the generated fractal pattern as a function of the input parameter.

Fig. 9 schematically illustrated a block diagram 900 of a challenged functional unit 120. The parameter represents the challenge. The parameter includes an initial
30 value 902 and specifies a section 904 of a fractal pattern 906 generated by the fractal generator 620.

The random filter 622 distorts the bitmap representing the fractal pattern 906 within the section 904. The bitmap 908 including the modified section 904 of the fractal
35 pattern 906 is a response in the communication 600.

- 31 -

The distortion introduced by the random filter 622 includes at least one of rotation, squeezing, expanding and non-linearly warping the section 904 of the fractal pattern 906. The action of the random filter 622 is independent of the parameter. The invariant computed by the invariant filter 612 is independent of the distortion
5 introduced by the random filter 622.

Fig. 10 schematically illustrates a dependency of the fractal pattern 906 on the parameter. In the embodiment of the apparatus 900 including only the identifying fractal generator 620 or in an embodiment of the communication 600 using
10 equivalent fractal generators 610 and 620, a fractal dimension, D_F , depends on the parameter. The invariant filter 612 in the challenging functional unit 110 computes the fractal dimension.

Alternatively or in addition to aforementioned identification based on fractals, the
15 identification of a functional unit is based on complex function. The complex function is holomorphic in a domain 1102 of the complex plane 1100 schematically illustrated in Fig. 11. Any holomorphic function, f , and any domain 1102 can be stored in the peer units for identification. The domain 1102 is specified as to any one of its shape, size and punctured isolated singularities. The function, f , identifies the challenged
20 functional unit 120. The function is represented by a sequence of coefficients of a Taylor or Laurent series expansion converging within the domain 1102. The number of coefficients may be limited so that a truncated series approximates the function with machine precision.

In a first implementation of a method of identification, a challenge message from the
25 challenging functional unit 110 requests samples of the function at sample points 1104 within the domain 1102. The challenged functional unit 120 numerically computes the function at the requested samples and responds to the challenging functional unit 110 by sending the computed complex values. Since it is impossible
30 to deduce the holomorphic function based on the discrete set of isolated samples points 1104, the function remains secret even if the communication is eavesdropped on. In this sense, the holomorphic function is not clonable. The method can be generalized using a meromorphic function by excluding isolated poles of the function from the domain 1102, or by including the non-numerical value "Not a Number"
35 (NaN) for the pole values or values close to the pole in the response.

- 32 -

A second implementation corresponds to the methods 700 and 800, wherein the "fractal generator" is replaced by the computation of the function, and the "fractal pattern" is replaced by an array of samples of the function. The "parameter" specifies the set of sample points 1104.

A third implementation of the method is related to the Casorati–Weierstrass theorem, which implies: A function f be holomorphic in a punctured neighborhood of z_0 and has an essential singularity at z_0 . Then, $f(z)$ approaches any complex value within the punctured neighborhood.

The function is numerically computed by the challenged functional unit 120 at the sample points. The set of sample values at the sample points is referred to as a pattern of the function. Again, the pattern is an incomplete representation of the function, i.e., it is not possible to determine the function based on the pattern sent to the challenging unit 110.

The sample points for the pattern are selected by the challenging functional unit 110 within a ring 1202,

$$\{z \in \mathbb{C} \mid r_1 < |z - z_0| < r_2\}.$$

The ring 1202 is fully comprised within the neighborhood 1200 of z_0 , as is schematically illustrated in Fig. 12. An inner boundary 1204 of the ring 1202 corresponds to a first radius r_1 . An outer boundary 1206 of the ring 1202 corresponds to a second radius r_2 that is greater than the first radius.

The series expansion used for the numerical computation includes a number of terms (and a corresponding number of coefficients) so that the remainder of the series is below a value for the machine precision. For a given machine precision, the number of stored coefficients of a Laurent series expansion about the essential singularity z_0 depends on the first radius r_1 . More specifically, a smaller first radius requires a higher number of coefficients to be stored and used in the computation.

The function f varies more and more rapidly the closer the sample points are to the essential singularity z_0 , which is a consequence of a stronger theorem by Émile Picard, known as the Big Picard Theorem. Hence, more coefficients are used for a smaller first radius r_1 .

- 33 -

In an advanced implementation, the first radius r_1 is chosen such that the machine precision, i.e., a floating-point representation of the sample points z_i , cannot resolve the "oscillations" off near r_1 . As a consequence, the pattern sent by the challenged functional unit 120 in the response to the challenging functional unit 110 is not only indicative of the function f . The pattern is also indicative of the hardware of the challenged functional unit 120, e.g., a floating-point unit included in the challenged functional unit 120 and used for the numerical computation. Alternatively or in addition, the pattern is also indicative of the software of the challenged functional unit 120, e.g., the algorithm used for the numerical computation.

Abovementioned fractals and complex functions are examples for a logical unclonable feature (LUF). Alternatively or in combination, the unclonable feature used for the identification is a physical unclonable feature (PUF).

Fig. 13 schematically illustrates a cross-sectional view of an electromagnetic cavity 1300 implementing a physical unclonable feature. A closed metallic surface 1302 defines the geometry and, thus, the electromagnetic properties of the cavity 1300.

Eigen functions of a stationary electromagnetic radiation field, which is induced by an emitting antenna 1304, are susceptible to the shape of the inner surface 1302 and thus represent the physically unclonable feature. A receiving antenna 1306 detects the local electromagnetic field as a function of frequency. A transfer function for electromagnetic radiation from the emitting antenna 1304 to the receiving antenna 1306 is determined as a function of frequency. The transfer function is a precise indicator for identity, because it is sensitive to details of, and minute changes in, the shape of the cavity 1300.

In addition, an inner volume 1308 enclosed by the cavity surface 1302 is substantially evacuated or filled by a dielectric. The dielectric filling 1308 includes, e.g., a gas, a homogeneous liquid or a solid. The solid filling can comprise different sections, each section having a different dielectric constant ϵ_r . The dielectric filling 1308 of the cavity 1300 individualizes the transfer function beyond the geometrical shape of the cavity 1300. Preferably, the solid filling is manufactured by subsequently filling different glues into the inner volume so as to prepare layers of different glues. A short rotational acceleration of the closed cavity 1300 twists the different layers. The twisted layers of glue are cured for a unique and reproducible transfer function.

- 34 -

The challenging functional unit 110 specifies a set of sample frequencies. The challenged functional unit 120 responds by sending real and imaginary parts of the transfer function as measured for the set of sample frequencies. Moreover, the transfer function (e.g., as a function of complex-valued frequency) is also a meromorphic function. Its poles are related to causality. Consequently, the method described above for the logical unclonable feature is also applicable to the transfer function as a physical unclonable feature.

The physical unclonable feature, optionally in combination with one or more of the other unclonable features described herein, is based on a phosphorescent decay. Fig. 14 schematically illustrates a cross-sectional view of an enclosure 1400 including a phosphorescent layer 1402, an excitation light source 1404 and a plurality of frequency-selective light detectors 1406 to 1410.

The excitation light is UV light and the phosphorescent material 1402 emits light in the visual spectrum. For example, the excitation light source 1404 is a UV light diode emitting the UV light in a range from 100 nm to 380 nm, preferably from 300 nm to 350 nm.

In the embodiment shown in Fig. 14, each of the frequency-selective light detectors 1406 to 1410 includes a photodiode and a foil filter 1412 to 1416 selective for a spectral line in the visual spectrum.

Phosphorescence is measured by the challenged functional unit 120 as a function of time at different wavelengths and/or for different excitation modes. The excitation mode includes an excitation intensity and an excitation period during which the excitation light irradiates upon the phosphorescent material 1402.

The phosphorescence material 1402 individualizes an associated functional unit, e.g., by a unique combination of constituents of the phosphorescence material 1402. The constituents may include cerium (Ce) and/or terbium (Tb).

Fig. 15 schematically illustrates the phosphorescence intensity (on the vertical axis) as a function of time (in arbitrary units) for three different wavelengths. The intensities at different wavelengths are normalized with respect to an initial phosphorescence intensity, i.e., the phosphorescence intensity at the time when the excitation period ends.

- 35 -

Fig. 16 schematically illustrates the phosphorescence intensity (on the vertical axis in arbitrary units without renormalizing the phosphorescence intensities) as a function of time for three different excitation intensities. Preferably, the excitation mode is chosen by the challenged functional unit such that a characteristic decay time of the response function of the phosphorescent material is substantially identical for the different wavelengths measured by the detectors 1406 to 1410.

The challenge specifies a set of points in time and the response provides the intensity measured at the specified points in time. The phosphorescence intensities of the different wavelengths represent a vector-valued function of time. Any one or all of the steps described in the context of logical unclonable features are applicable to the vector-valued function of time.

Furthermore, the values of one or more identifying function (including the cases of a logical function and/or a physical function, e.g., described with reference to any one of Figs. 10 to 16) can be sent in the response by means of a data split. The data split can be generated by including the response according to a first unclonable feature in even bits of each byte and including the response according to a second unclonable feature in the odd bits of the bytes. Since each of the first and second response has an entropy close to unity, the data split representation of both the first and second responses has an even higher entropy (i.e., even closer to unity), since the responses are uncorrelated. This large entropy (i.e., the entropy close to unity) prevents a man-in-the-middle attack.

Any one or all of the identification means for a functional unit described above can be included in a subunit associated to the functional unit. The subunit is also referred to as a token in this context. For example, the subunit 114 includes a holomorphic function for identifying the functional unit 110 (e.g., when in the role of the challenged functional unit) and/or verification means (e.g., when in the role of the challenging functional unit).

As has become apparent from above description of embodiments, the technique disclosed herein provides an architecture for both secure and reliable data processing.

Claims

- 5 1. A device (100) for data processing, comprising at least three functional units (110, 120, 130, 140), each of which includes a processing component and a memory component coupled to the processing component, wherein the processing component is adapted to perform instructions encoded into the memory component.
- 10 2. The device (100) of claim 1, further comprising a connecting component (240) adapted to selectively connect and/or selectively disconnect the functional units (110, 120, 130, 140).
3. The device (100) of claim 2, wherein the disconnected functional units (110,
15 120, 130, 140) are mutually galvanically separated.
4. The device (100) of claim 2 or 3, wherein the functional units (110, 120, 130, 140) are selectively connected or disconnected by means of at least one of field-effect transistors and microelectromechanical systems.
- 20 5. The device (100) of any one of claims 2 to 4, wherein all interfaces (510, 520) of each of the disconnected functional units (110, 120, 130, 140) are isolated.
6. The device (100) of any one of claims 2 to 5, wherein the connecting
25 component (240) includes a light transmitter, fiber optics and a light receiver arranged to selectively connect the functional units (110, 120, 130, 140), wherein the connecting component (240) is adapted to selectively disconnect the functional units (110, 120, 130, 140) by powering off at least one of the light transmitter and light receiver, and/or by optically obstructing the fiber optics.
- 30 7. The device (100) of any one of claims 2 to 6, wherein the connected functional units (110, 120, 130, 140) are pairwise connected.
8. The device (100) of any one of claims 2 to 7, wherein the connecting
35 component (240) is adapted to selectively connect functional units (110, 120, 130, 140) in pairs so that each connected functional unit (110; 120; 130; 140) is connected to only one other connected functional unit (110; 120; 130; 140).

- 37 -

9. The device (100) of any one of claims 2 to 8, wherein the connecting component (240) is adapted to connect no more than two of the functional units (110, 120, 130, 140) at a time.

5

10. The device (100) of any one of claims 2 to 9, wherein the connecting component (240) includes a crossbar switch.

10

11. The device (100) of any one of claims 2 to 10, wherein the functional units (110, 120, 130, 140) are adapted for data transfer only via the connecting component (240).

15

12. The device (100) of any one of claims 2 to 11, wherein each of the functional units (110, 120, 130, 140) includes a protocol stack (210; 220) implementing communication protocols for data transfer.

13. The device (100) of claim 12, wherein the protocol stack (210; 220) includes at least one of:

20

(i) a Resources Control Layer (1_e) adapted to identify a peer functional unit (110; 120; 130; 140) based on data received from the peer functional unit (110; 120; 130; 140), to verify the authenticity of the peer functional unit (110; 120; 130; 140) based on the received data, and/or to verify the authorization of the peer functional unit (110; 120; 130; 140) for applying a Data Link Layer communication protocol or for passing the data packet (230) to a Data Link Layer (2) of the protocol stack (210; 220);

25

(ii) a Physical Switching Layer (3_e) adapted to include configuration data (530) defining a mode of operation of the connecting component (240) into a data packet (230) to be transferred; and

30

(iii) an Object Control Layer (5_e) adapted to identify a peer application based on data received from the peer application, to verify the authenticity of the peer application based on the received data, and/or

35

- 38 -

to verify the authorization of the peer application for providing the transferred data to an Application Layer (7) of the protocol stack (210; 220).

- 5 14. The device (100) of claim 13, wherein the connecting component (240) is adapted to read configuration data (530) defining a mode of operation of the connecting component (240) from a data packet (230) received for transfer via the connecting component (240) and to enter the mode of operation according to the configuration data (530).
- 10 15. The device (100) of claim 14, wherein the mode of operation limits or defines the number of mutually connected functional units (110, 120, 130, 140) for the transfer of the data packet (230).
- 15 16. The device (100) of any one of claims 13 to 15, wherein the peer functional unit (110; 120; 130; 140) is identified based on a logical unclonable feature, optionally a meromorphic function.
- 20 17. The device (100) of any one of claims 13 to 16, wherein the Resources Control Layer (1_e) is further adapted to include a signature derived from a substantially irreproducible feature into a data packet (230) to be transmitted.
- 25 18. The device (100) of any one of claims 13 to 17, wherein a security anchor is located within each of the functional units (110, 120, 130, 140), wherein the security anchor stores a private key of the respective functional unit (110; 120; 130; 140).
- 30 19. The device (100) of claim 18, wherein the Object Control Layer (5_e) is adapted to retrieve the private key from the security anchor and to provide the private key to a presentation layer (6) for decrypting at least a part of the received data packet (230) using the private key.
- 35 20. The device (100) of claim 18, wherein the Object Control Layer (5_e) is adapted to retrieve the private key from the security anchor and to decrypt at least a part of the received data packet (230) using the private key.
21. The device (100) of any one of claims 2 to 20, wherein one of the functional units (110, 120, 130, 140), which is referred to as a supervising functional unit (110;

- 39 -

120; 130; 140), is adapted to control and/or currently controls the connecting component (240).

22. The device (100) of claim 21, wherein a supervisor program is encoded into the memory of the supervising functional unit (110; 120; 130; 140).

23. The device (100) of claim 22, wherein the supervising functional unit (110; 120; 130; 140) alternates between a monitoring state and a processing state for executing the supervisor program.

24. The device (100) of any one of claims 21 to 23, wherein another one of the functional units (110, 120, 130, 140) is adapted to take control over the connecting component (240) in response to a failure or an exception occurring at the functional unit (110; 120; 130; 140) currently controlling the connecting component (240).

25. The device (100) of any one of claims 21 to 24, wherein the supervising functional unit (110; 120; 130; 140) is adapted to segment the supervisor program into instruction modules and to associate at least two of the other functional units (110, 120, 130, 140) to different instruction modules for executing the respective one or more instruction modules by the associated functional unit (110; 120; 130; 140).

26. The device (100) of claim 25, wherein the supervising functional unit (110; 120; 130; 140) is adapted to switch from the monitoring state indicating no error for an associated functional unit (110; 120; 130; 140) to the processing state, and to send the respective instruction module to the associated functional unit (110; 120; 130; 140).

27. The device (100) of claim 26, wherein the supervising functional unit (110; 120; 130; 140) is further adapted to switch from the processing state, when the associated functional unit (110; 120; 130; 140) has completed or terminated the transferred instruction module, to the monitoring state for the associated functional unit (110; 120; 130; 140).

28. The device (100) of any one of claims 21 to 27, wherein the supervisor functional unit (110; 120; 130; 140) is adapted to control the connecting component (240) to connect a sending functional unit (110; 120; 130; 140) and a receiving

- 40 -

functional unit (110; 120; 130; 140) upon receiving matching requests from the sending functional unit (110; 120; 130; 140) and the receiving functional unit (110; 120; 130; 140).

5 29. The device (100) of claim 28, wherein the matching requests include
(i) an output request from the sending functional unit (110; 120; 130; 140),
the output request indicating the receiving functional unit (110; 120; 130;
140) as recipient of the data to be transferred, and
(ii) an input request from the receiving functional unit (110; 120; 130; 140),
10 the input request indicating the sending functional unit (110; 120; 130;
140) as originator of the data to be transferred.

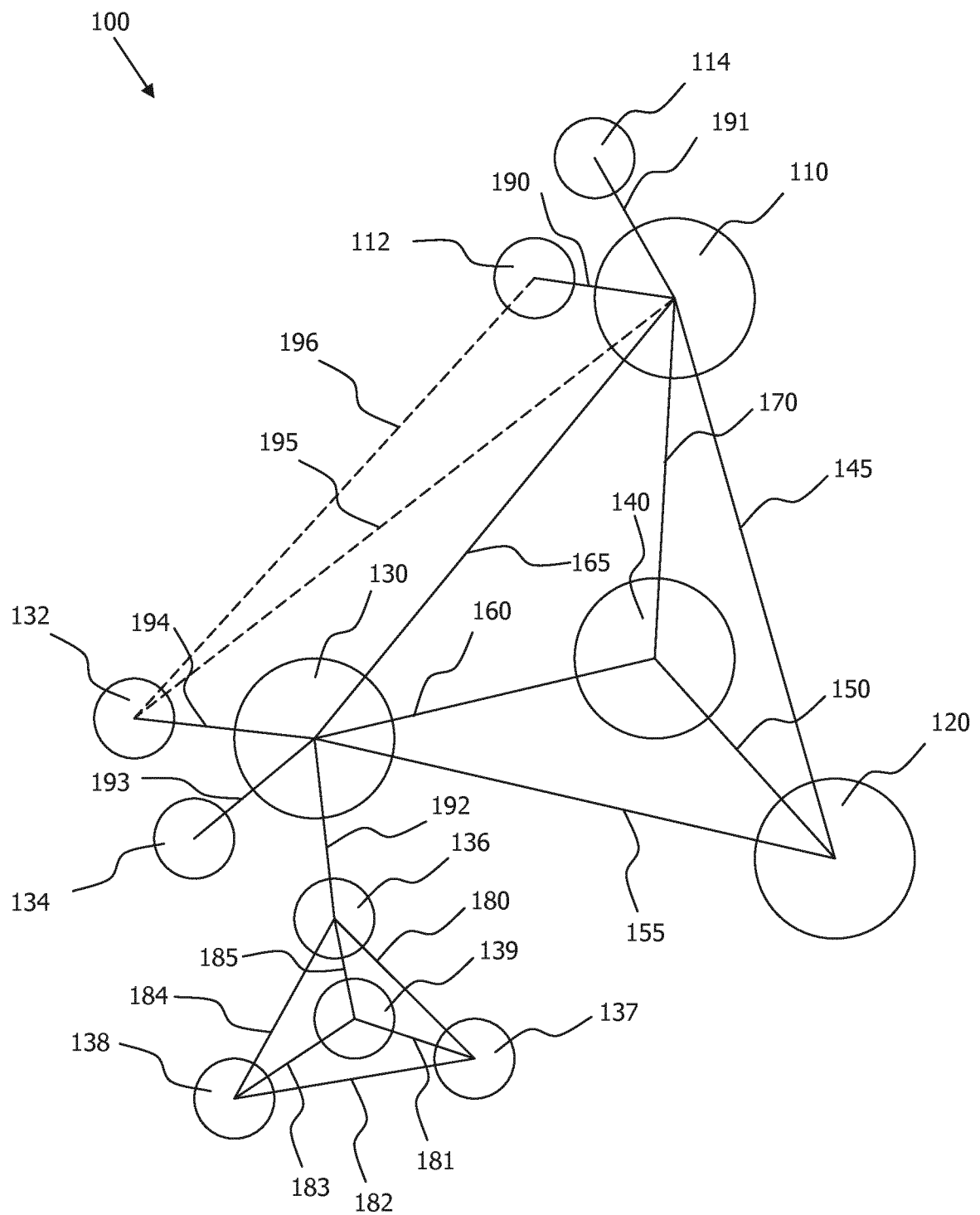
30. The device (100) of claim 26 or 27 in conjunction with claim 28 or 29, wherein
the supervising functional unit (110; 120; 130; 140) is further adapted to resend the
15 instruction module previously sent to a first functional unit (110; 120; 130; 140) in
response to receiving from the first functional unit (110; 120; 130; 140) an output
request indicating a second functional unit (110; 120; 130; 140) as recipient that
does upon completion of its instruction module not provide a matching input request
indicating the first functional module as originator.

20 31. The device (100) of claim 26 or 27 in conjunction any one of claims 28 to 30,
wherein the supervising functional unit (110; 120; 130; 140) is further adapted to
resend an instruction module previously sent to a first functional unit (110; 120; 130;
140) in response to receiving from the first functional unit (110; 120; 130; 140) an
25 input request indicating a second functional unit (110; 120; 130; 140) as originator
that does upon completion of its instruction module not provide a matching output
request indicating the first functional module as recipient.

32. The device (100) of any one of claims 1 to 31, wherein the functional units
30 (110, 120, 130, 140) are spatially divided from each other.

33. The device (100) of any one of claims 1 to 32, wherein the functional units
(110, 120, 130, 140) are separately arranged next to one another.

35 34. The device (100) of any one of claims 1 to 33, wherein the at least three
functional units (110, 120, 130, 140) and the connecting component (240) are
encapsulated into a demilitarized zone and/or a tamper-proof zone.

**Fig. 1**

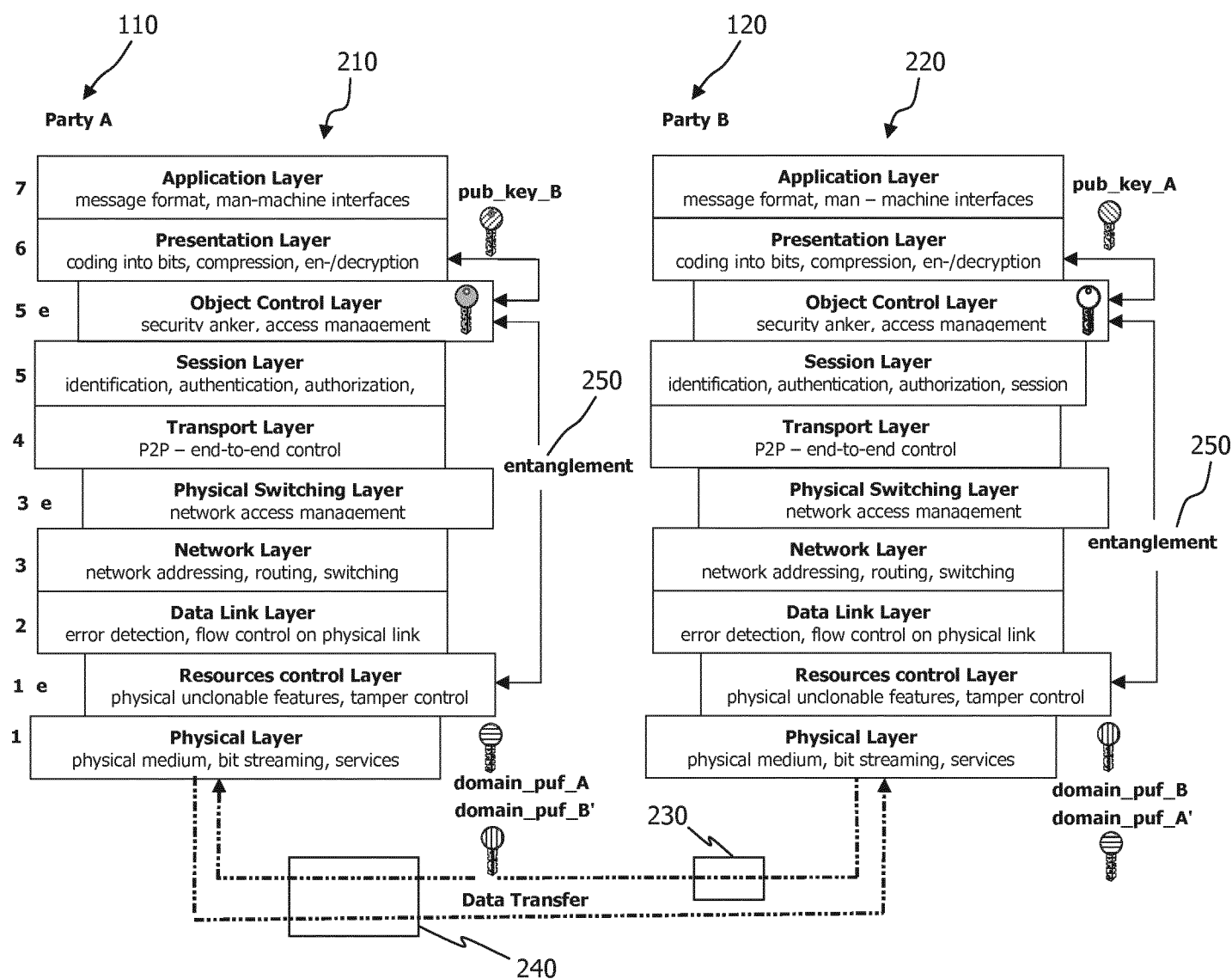


Fig. 2

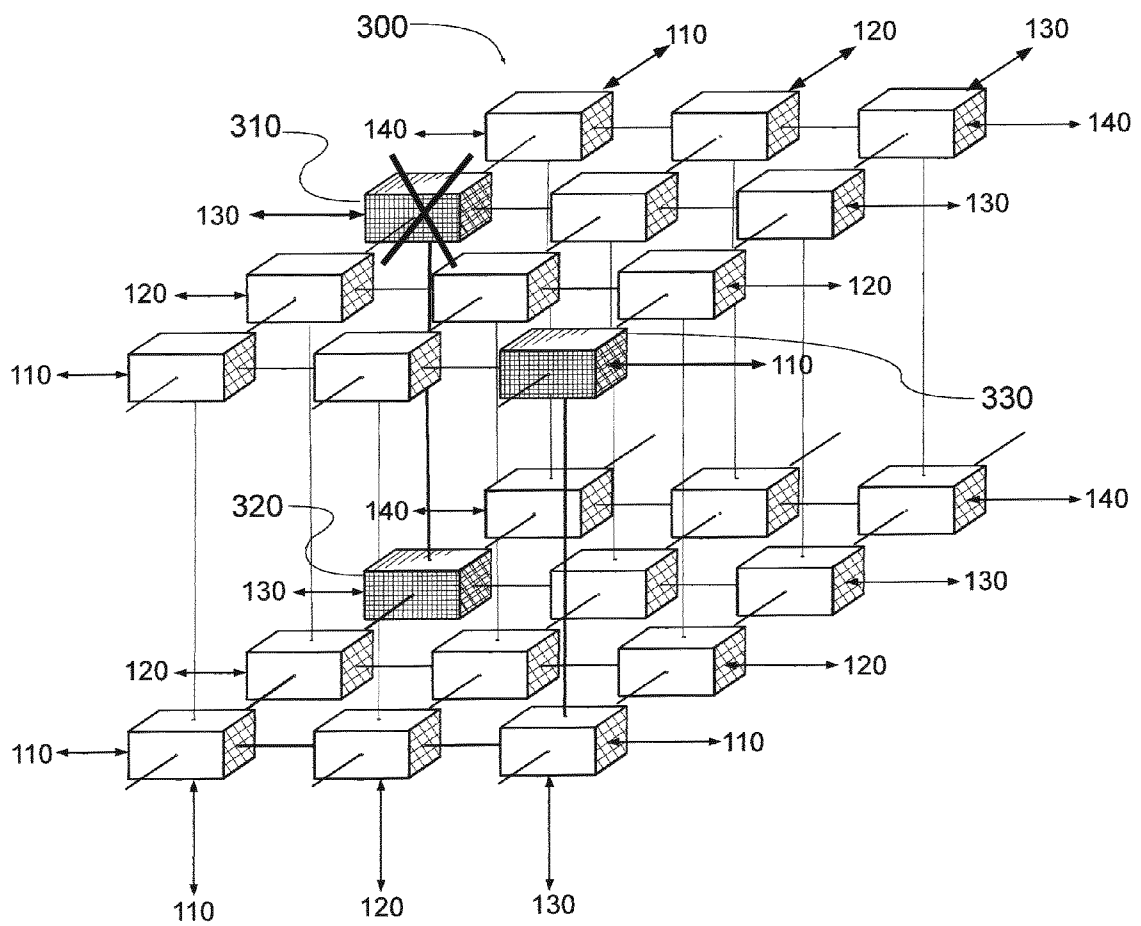


Fig. 3

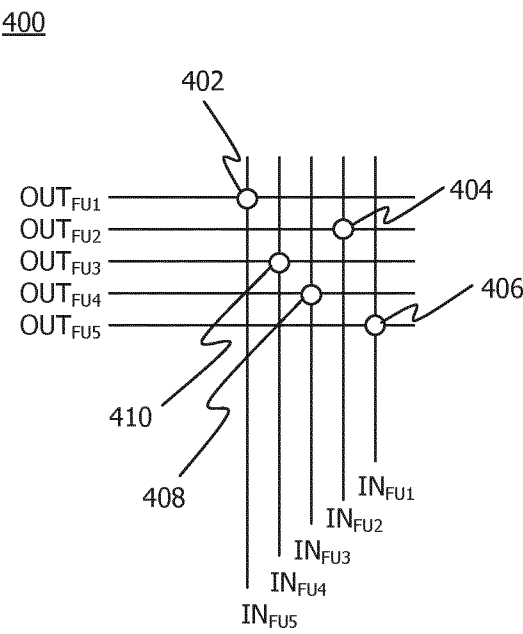


Fig. 4a

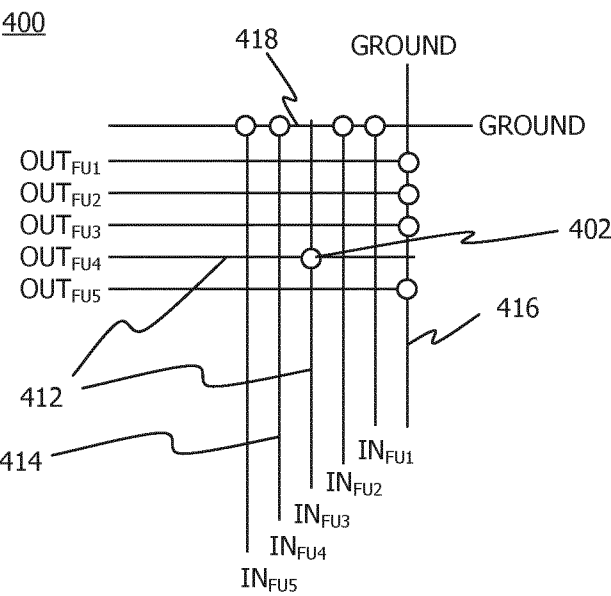


Fig. 4b

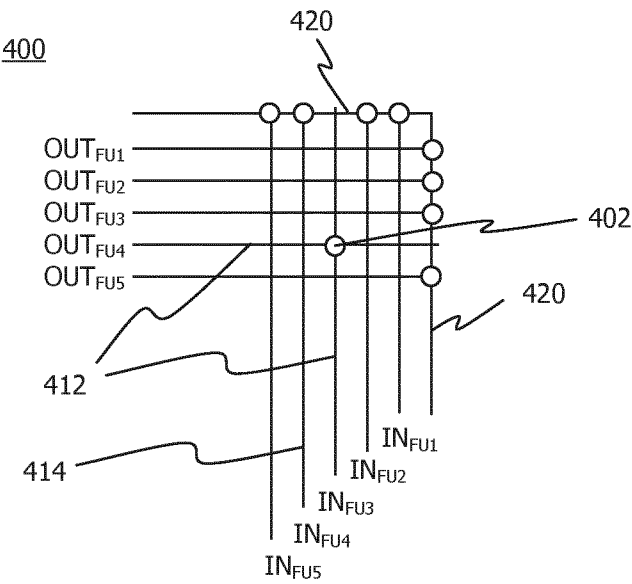


Fig. 4c

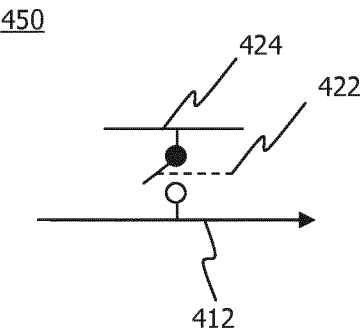


Fig. 4d

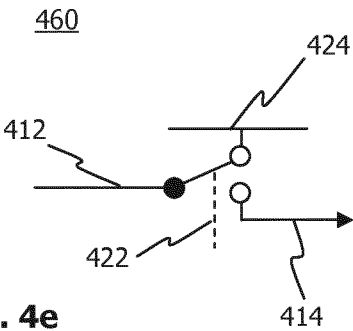


Fig. 4e

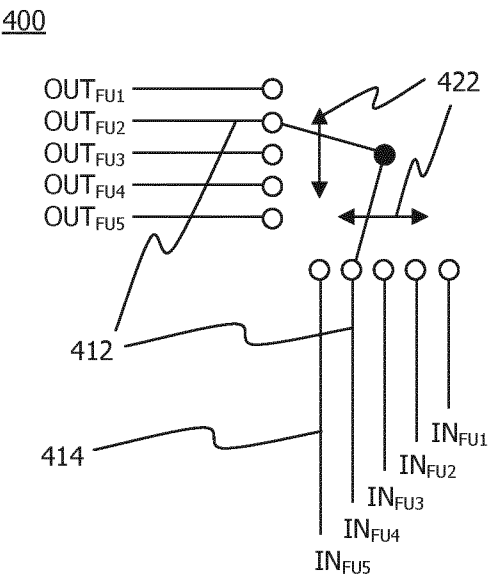
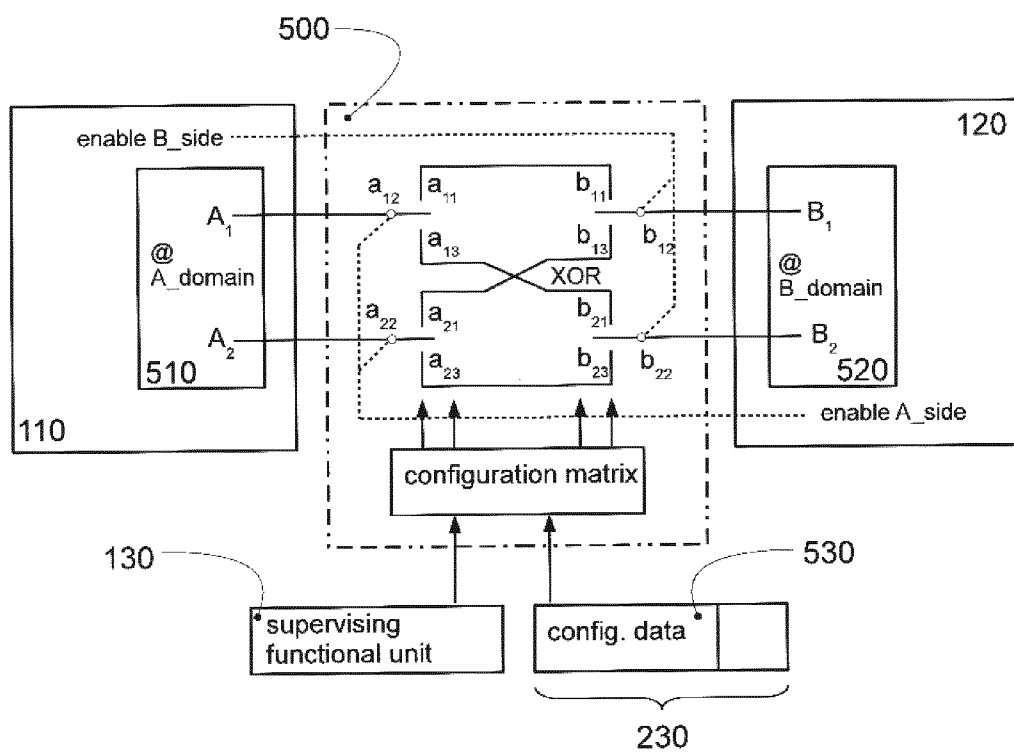
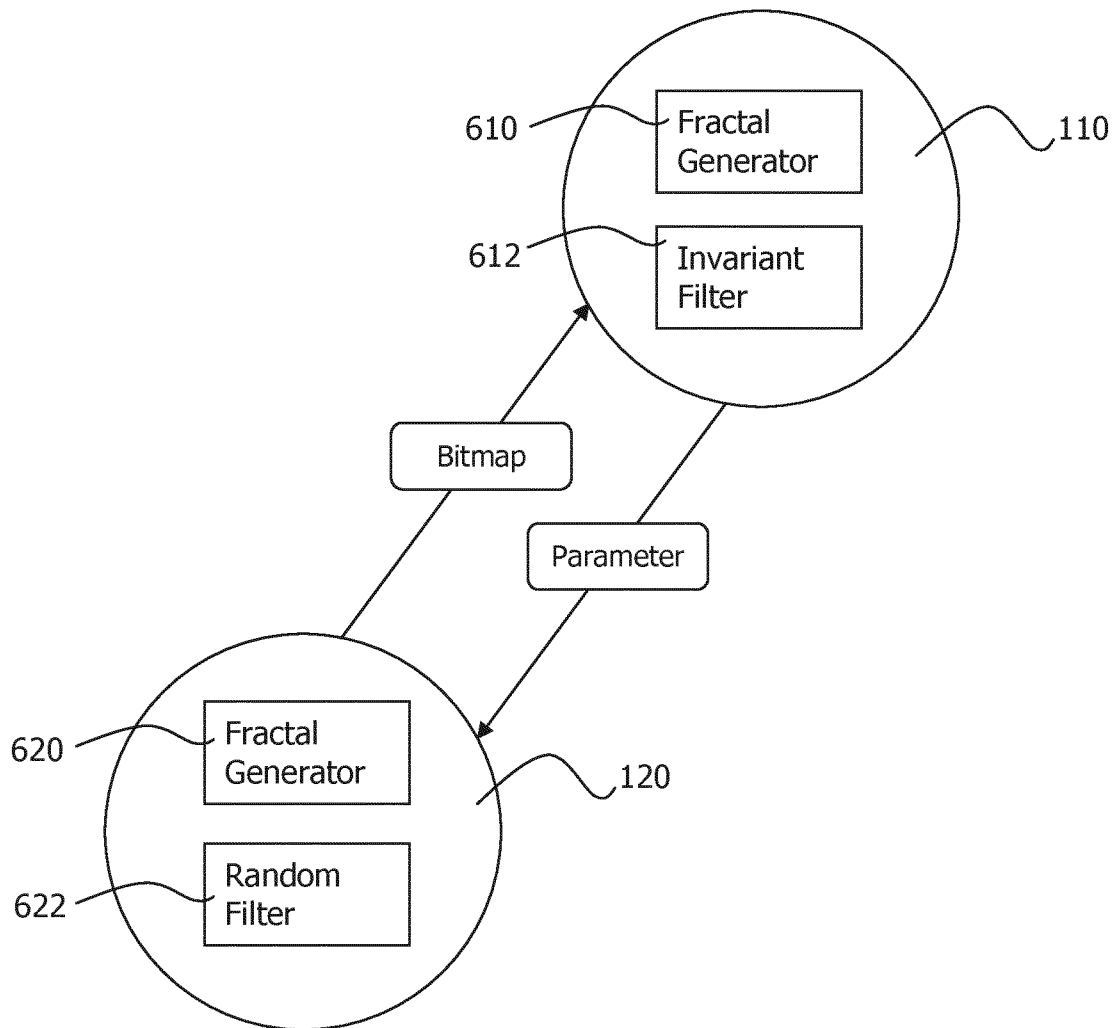
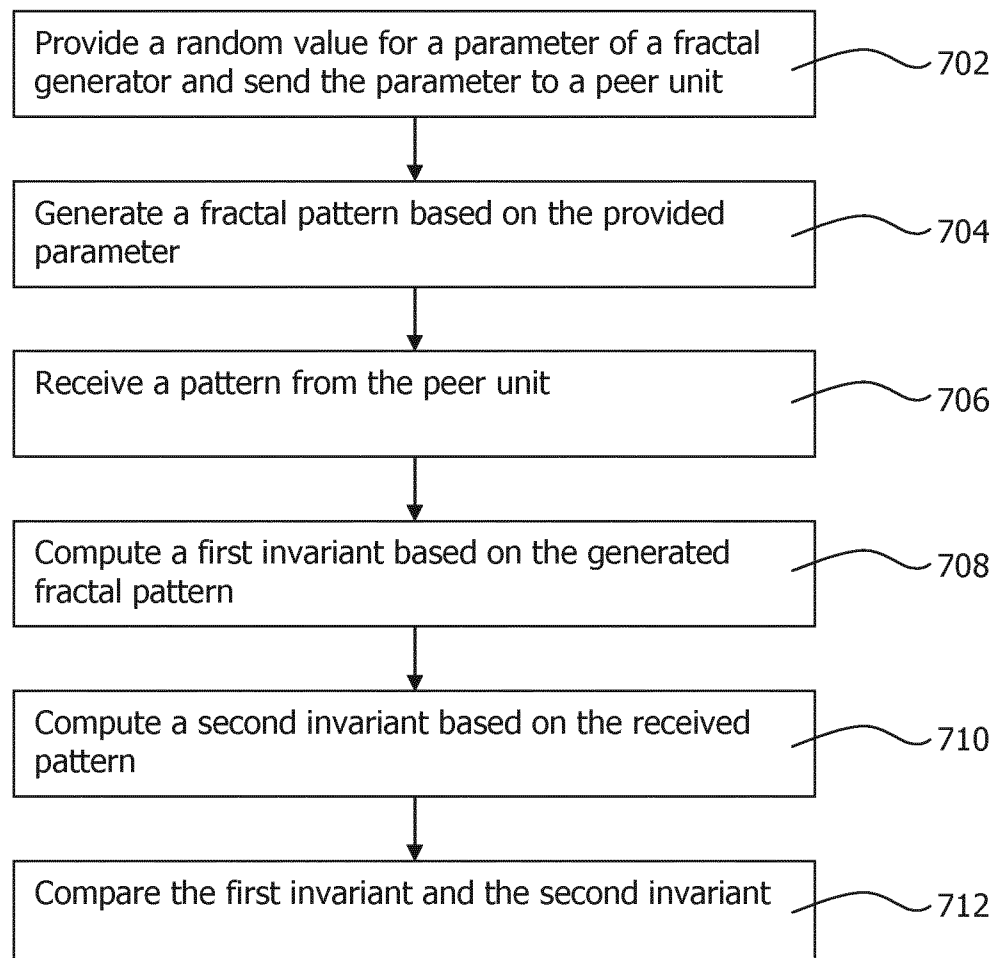
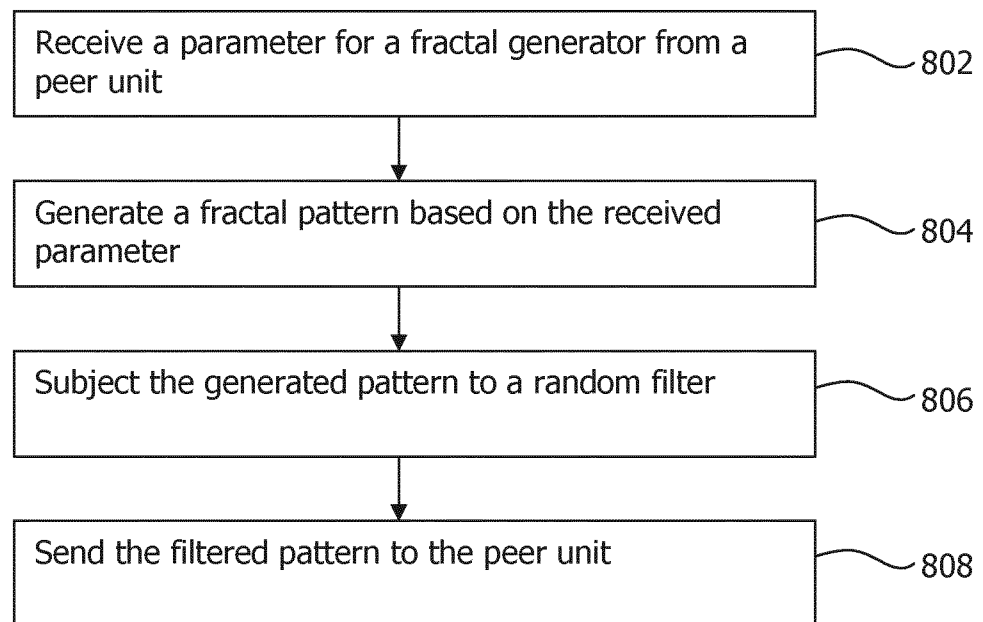


Fig. 4f

**Fig. 5**

600**Fig. 6**

700**Fig. 7**

800**Fig. 8**

900

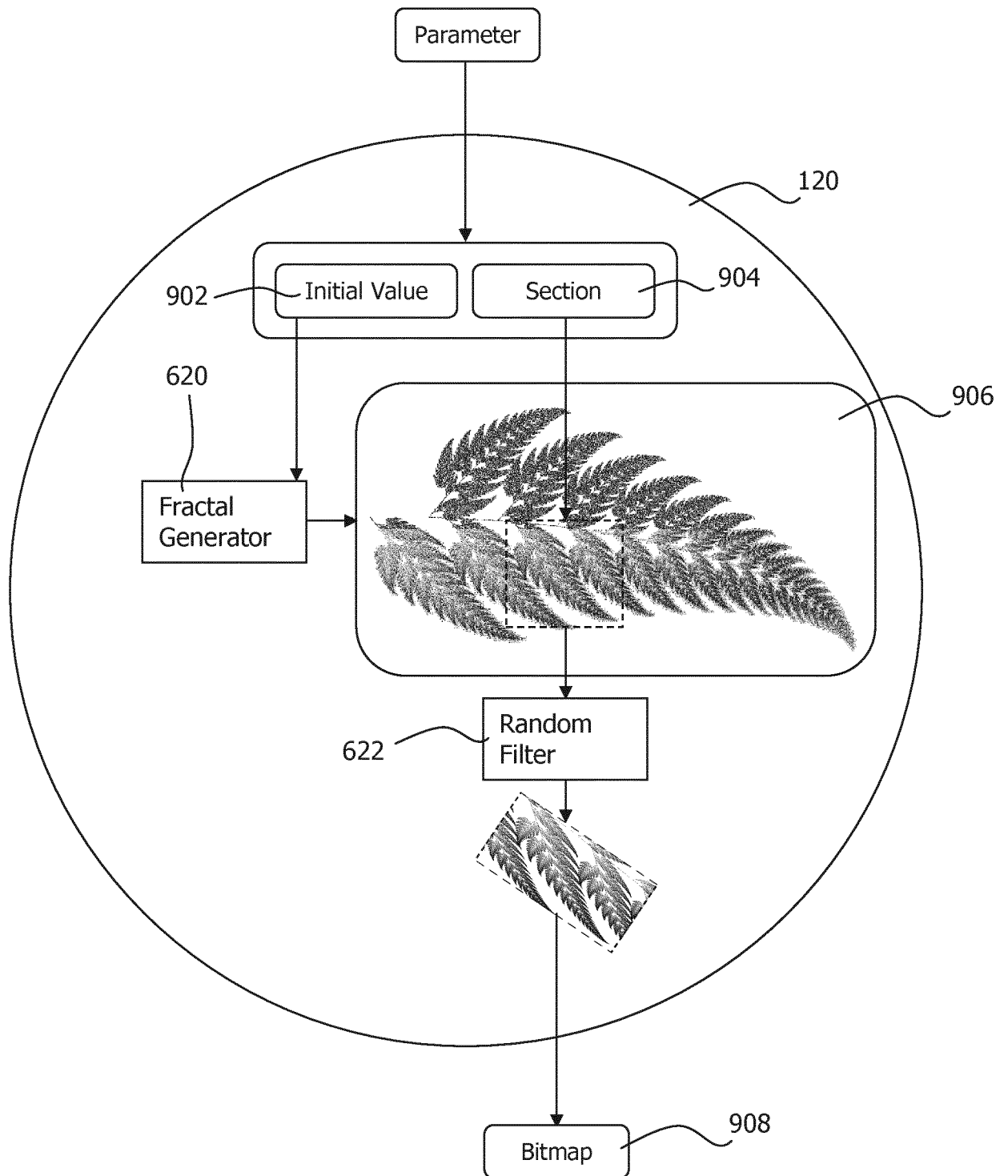
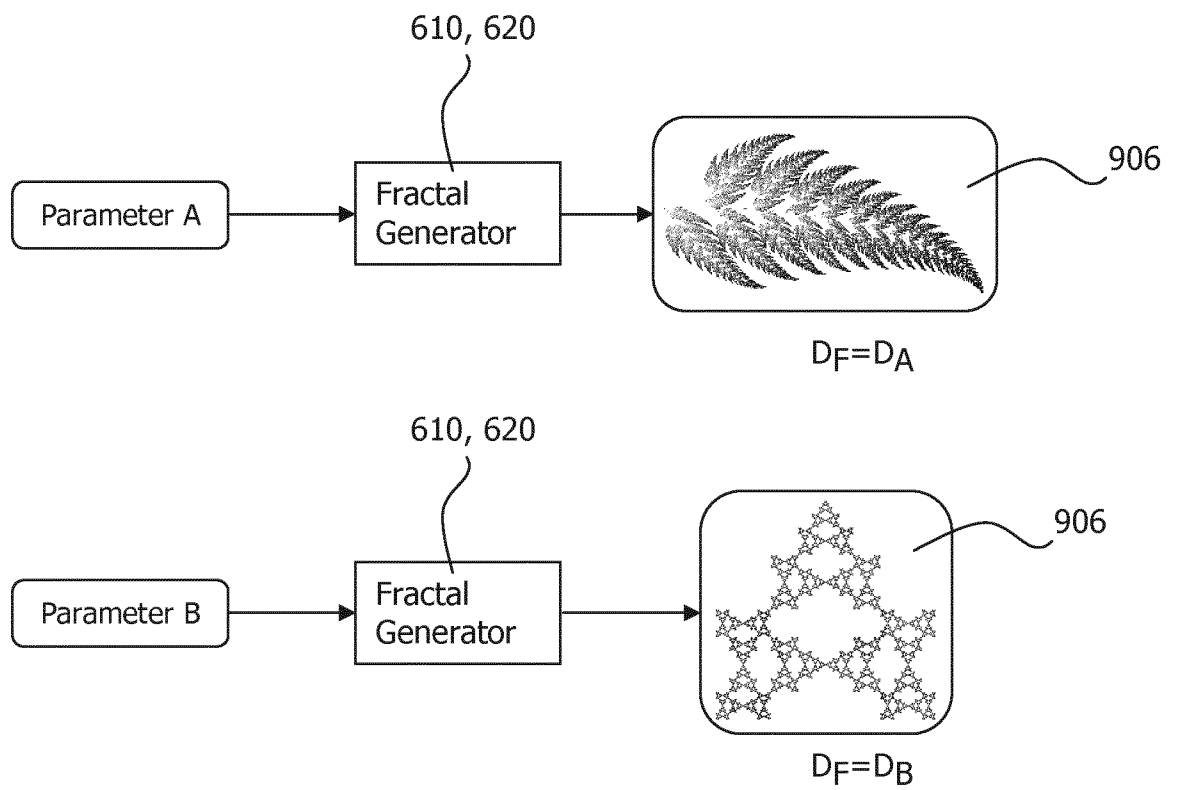


Fig. 9

**Fig. 10**

1100

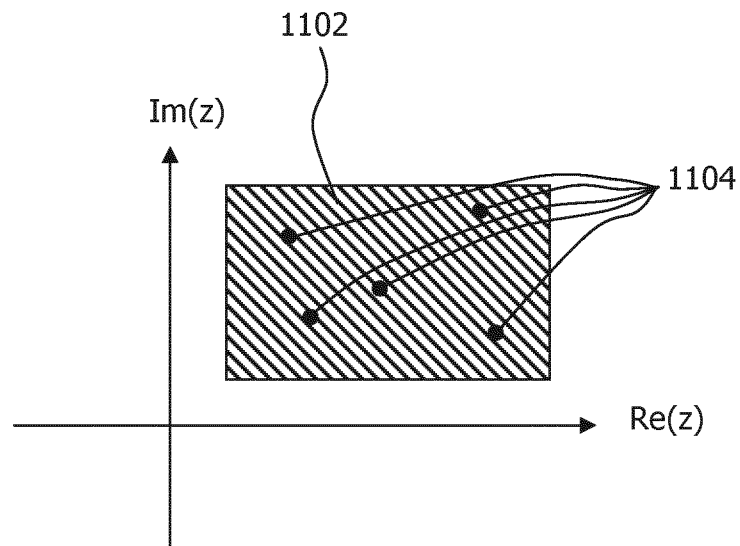


Fig. 11

1200

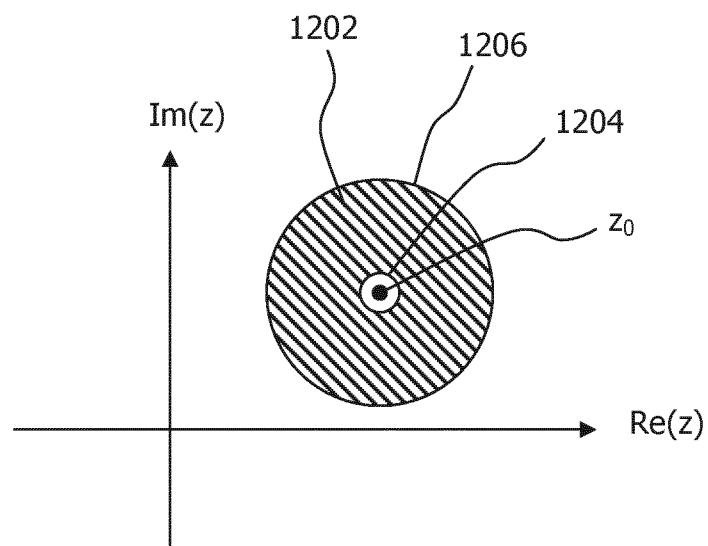


Fig. 12

1300

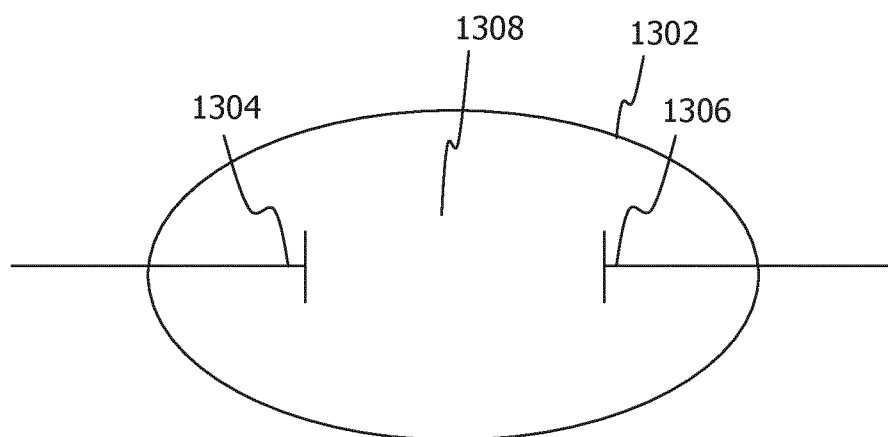
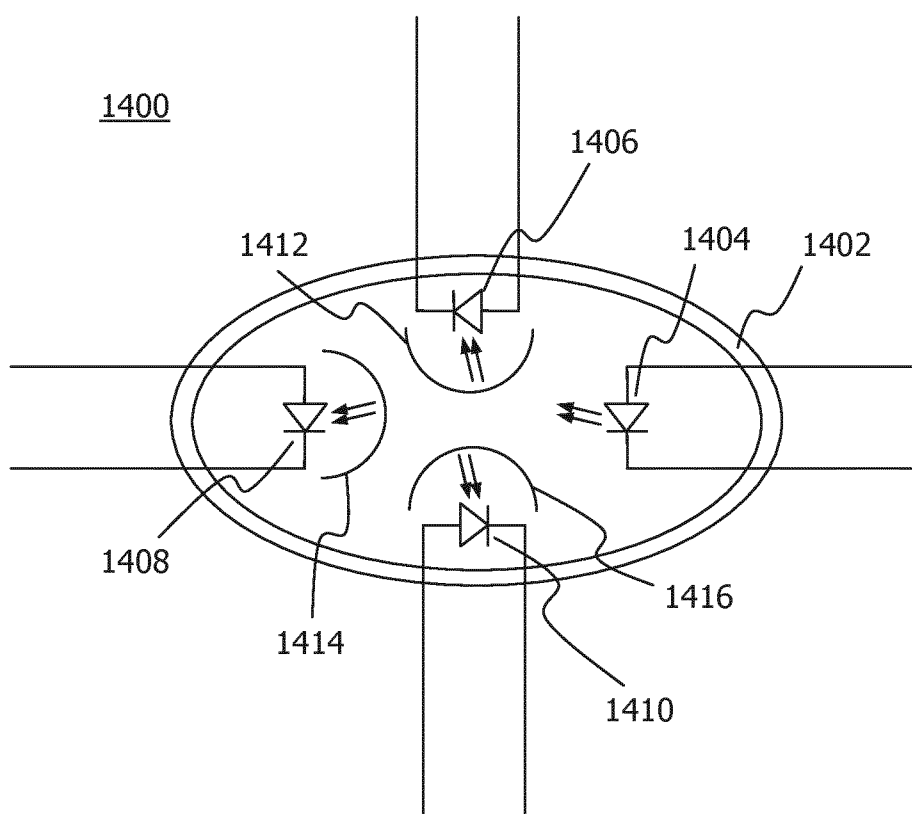


Fig. 13

1400



Intensity of Phosphorescence
at different wavelengths

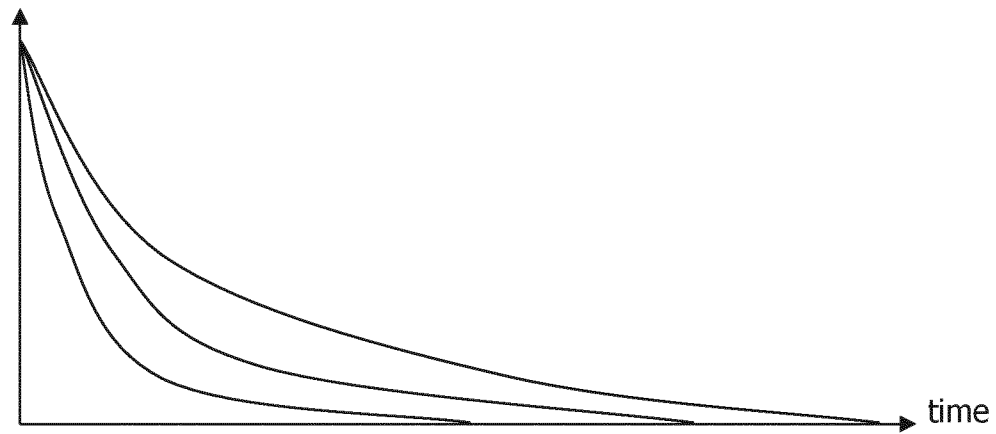


Fig. 15

Phosphorescence Intensity
for different excitation modes

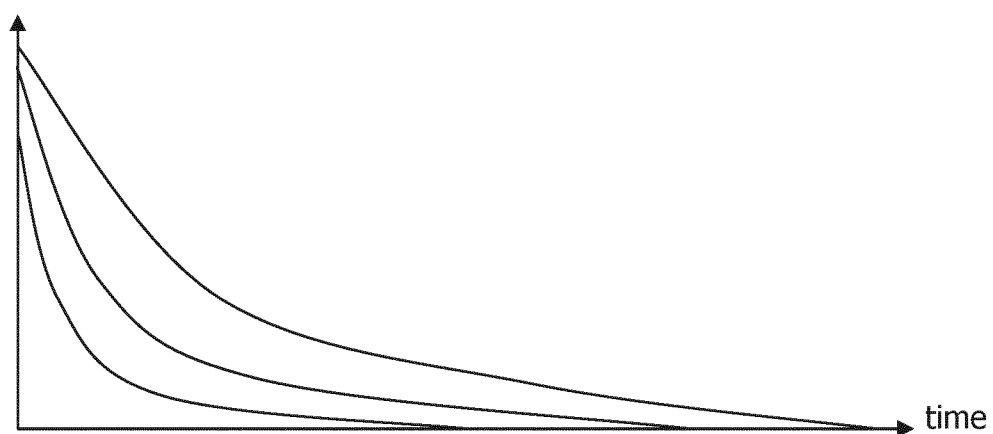


Fig. 16

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2013/071699

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F15/16

ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2008/114163 A1 (NOKIA CORP [FI]; NOKIA INC [US]; FORSELL MARTTI [FI]) 25 September 2008 (2008-09-25) page 3, line 6 - page 4, line 18; claim 38 -----	1-34
X	US 5 649 106 A (TSUJIMICHI SHINGO [JP] ET AL) 15 July 1997 (1997-07-15) claim 1 column 4, lines 5-21 -----	1-34



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

18 December 2013

Date of mailing of the international search report

03/01/2014

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Veen, Gerardus

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2013/071699

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2008114163 A1	25-09-2008	US 2008232445 A1 WO 2008114163 A1	25-09-2008 25-09-2008
US 5649106 A	15-07-1997	JP 2770603 B2 JP H04348451 A US 5649106 A	02-07-1998 03-12-1992 15-07-1997