(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 June 2016 (30.06.2016)

(10) International Publication Number
WO 2016/102739 A1

(51) International Patent Classification:
G01S 1/02 (2010.01) *G01S 5/02* (2010.01)
G01C 21/20 (2006.01) *H04W 12/00* (2009.01)
H04L 9/08 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/FI2014/051038

(22) International Filing Date:
22 December 2014 (22.12.2014)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: NOKIA TECHNOLOGIES OY [FI/FI]; Karaportti 3, FI-02610 Espoo (FI).

(72) Inventor: KERAI, Kanji; 1 Boycroft Avenue, London, NW9 8AH (GB).

(74) Agents: NOKIA TECHNOLOGIES OY et al.; Ari Aarnio, IPR Department, Karakaari 7, FI-02610 Espoo (FI).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SECURE BLUETOOTH LOW ENERGY TAGS

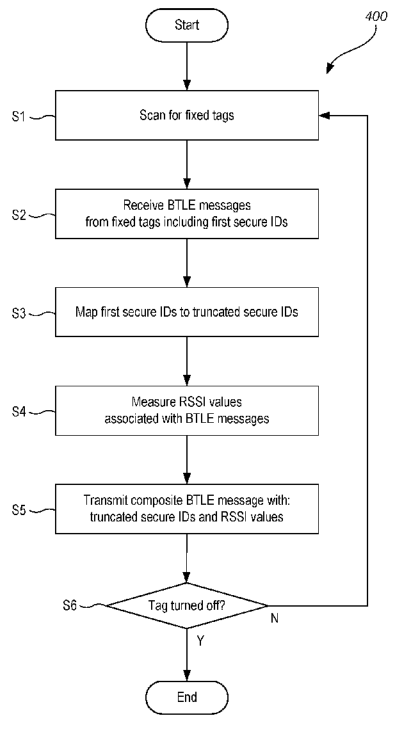

FIG. 4

(57) Abstract: A method comprises: a fixed tag obtaining an instant value from a dynamic number generator within said fixed tag; the fixed tag using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and the fixed tag transmitting a first Bluetooth Low Energy message including said first secure identifier. A further method (400) comprises: a mobile tag receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags (S2), each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag; the mobile tag using a second function to map each first secure identifier to a respective second secure identifier (S3); the mobile tag measuring a radio parameter for each of the first Bluetooth Low Energy messages (S4); and the mobile tag transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages (S5), said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

WO 2016/102739 A1

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

Secure Bluetooth Low Energy Tags

Field

The present disclosure relates to secure position tags, and secure low power position calculation using Bluetooth low energy.

Background

Bluetooth Low Energy (BTLE) is a wireless communication technology published by the Bluetooth Special Interest Group (BT-SIG) standard as a component of Bluetooth Core Specification Version 4.0. BTLE is a lower power, lower complexity, and lower cost wireless communication protocol, designed for applications requiring lower data rates and shorter duty cycles. Inheriting the protocol stack and star topology of classical Bluetooth, BTLE redefines the physical layer specification, and involves many new features such as a very-low power idle mode, a simple device discovery, and short data packets, etc.

BTLE technology is aimed at devices requiring a low level of power consumption, for example devices that may operate with one or more button cell batteries such as sensors, key fobs, and/or the like. BTLE can also be incorporated into devices such as mobile phones, smart phones, tablet computers, laptop computers, desktop computers etc.

Summary of Embodiments

Various aspects of examples of the present disclosure are set out in the claims.

A first aspect of the present disclosure provides a method comprising: a fixed tag obtaining an instant value from a dynamic number generator within said fixed tag; the fixed tag using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and the fixed tag transmitting a first Bluetooth Low Energy message including said first secure identifier.

The tag identifier may comprise a first identifier and a second identifier.

The first function may comprise: the fixed tag generating an encrypted value using said instant value and said first identifier; and the fixed tag concatenating said encrypted value with said second identifier.

The method may further comprise: the fixed tag using the first function to map said first and second identifiers and a new instant value obtained from said number generator to a further first secure identifier; and the fixed tag transmitting a further first Bluetooth Low Energy message including said further first secure identifier.

The encrypted value may be generated using an advanced encryption standard cipher.

The dynamic number may be generated using a network time protocol.

The instant value may be a timestamp.

The radio parameter may be received signal strength.

A second aspect of the present disclosure provides a method comprising: a mobile tag receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags, each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag; the mobile tag using a second function to map each first secure identifier to a respective second secure identifier; the mobile tag measuring a radio parameter for each of the first Bluetooth Low Energy messages; and the mobile tag transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages, said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

The second function may comprise the mobile tag truncating each first secure identifier to generate said second secure identifier.

The second Bluetooth Low Energy message may include an identifier corresponding to the mobile tag.

The radio parameter may be received signal strength.

The mobile tag may receive at least one first Bluetooth Low Energy message from each of at least three fixed tags.

A third aspect of the present disclosure provides a method comprising a Bluetooth receiver device receiving at least one first Bluetooth Low Energy message from each of at least two fixed tags; the Bluetooth receiver device receiving at least one second Bluetooth Low Energy message from a mobile tag; the Bluetooth receiver device decoding from each first Bluetooth Low Energy message a first secure identifier corresponding to a respective fixed tag; the Bluetooth receiver device decoding from each second Bluetooth Low Energy message at least one second secure identifier corresponding to each fixed tag, said second secure identifier being derived from said first secure identifier; the Bluetooth receiver device decoding from each second Bluetooth Low Energy message measured radio parameters associated with each of said first Bluetooth Low Energy messages; sever apparatus using said first and second secure identifiers to identify each fixed tag; the server apparatus retrieving position data corresponding to each fixed tag; and the server apparatus using the position data and the radio parameters to calculate a position of the mobile tag.

The method may further comprise the Bluetooth receiver device decoding from the second Bluetooth Low Energy message an identifier corresponding to the mobile tag.

The method may further comprise the server apparatus detecting whether the mobile tag is moving.

The method may further comprise the server apparatus performing position calculation more frequently in response to detecting that the mobile tag transitions to a moving state.

The method may further comprise the server apparatus using first Bluetooth Low Energy messages from a predetermined number of fixed tags from which strongest signals are received for calculating the position of the mobile tag if signals are received from the predetermined number of fixed tags.

The predetermined number may be at least three.

The position data may contain the latitude, longitude and floor number of the respective device.

The radio parameter may be received signal strength.

The method may further comprise the server apparatus calculating the position of more than one mobile tag.

The method may further comprise the server apparatus using the first Bluetooth Low Energy messages from at least two fixed tags with the highest values of the radio parameter for calculating the position of the mobile tag.

The method may further comprise the server apparatus storing the position data corresponding to each of the fixed tags, together with their corresponding first and second secure identifiers, in a memory device.

The memory device may be a non-volatile memory, preferably a read only memory.

The Bluetooth receiver device may comprise a transceiver device.

The Bluetooth receiver device may have a fixed position.

The Bluetooth receiver device may be capable of communication with the server apparatus via a data link.

The data link may comprise any of: a WiFi connection and a wired connection.

The Bluetooth receiver device may comprise a fixed tag selected from any of said at least two fixed tags.

A fourth aspect of the present disclosure provides apparatus comprising a fixed tag, the fixed tag comprising at least one processor, at least one memory, and computer-readable code stored on the at least one memory, wherein the computer-readable code when executed controls the at least one processor to perform a method comprising: obtaining an instant value from a dynamic number generator within said fixed tag; using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and transmitting a first Bluetooth Low Energy message including said first secure identifier.

The tag identifier may comprise a first identifier and a second identifier.

The first function may comprise: the fixed tag generating an encrypted value using said instant value and said first identifier; and the fixed tag concatenating said encrypted value with said second identifier.

A fifth aspect of the present disclosure provides apparatus comprising a mobile tag, the mobile tag comprising at least one processor, at least one memory, and computer-readable code stored on the at least one memory, wherein the computer-readable code when executed controls the at least one processor to perform a method comprising: receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags, each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag; using a second function to map each first secure identifier to a respective second secure identifier; measuring a radio parameter for each of the first Bluetooth Low Energy messages; and transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages, said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

The second function may comprise the mobile tag truncating each first secure identifier to generate said second secure identifier.

A sixth aspect of the present disclosure provides a computer program comprising computer executable instructions, which, when executed by a fixed tag, causes the fixed tag to perform: obtaining an instant value from a dynamic number generator within said fixed tag; using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and transmitting a first Bluetooth Low Energy message including said first secure identifier.

A seventh aspect of the present disclosure provides a computer program comprising computer executable instructions, which, when executed by a mobile tag, causes the mobile tag to perform: receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags, each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag; using a second function to map each first secure identifier to a respective second secure identifier; measuring a radio parameter for each of the first Bluetooth Low Energy messages; and transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages, said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

An eighth aspect of the present disclosure provides a non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by computing apparatus causes the computing apparatus to perform a method comprising: a fixed tag obtaining an instant value from a dynamic number generator within said fixed tag; the fixed tag using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and the fixed tag transmitting a first Bluetooth Low Energy message including said first secure identifier.

An ninth aspect of the present disclosure provides a non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by computing apparatus causes the computing apparatus to perform a method comprising: a mobile tag receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags, each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag; the mobile tag using a second function to map each first secure identifier to a respective second secure identifier; the mobile tag measuring a radio parameter for each of the first Bluetooth Low Energy messages; and the mobile tag transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages, said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

Brief Description of the Drawings

For a more complete understanding of example embodiments of the present disclosure, reference is now made to the following descriptions taken in connection with the accompanying drawings in which:

Figure 1(a) is a schematic diagram of a BTLE message used in embodiments of the present disclosure;

Figure 1(b) is a schematic diagram of a composite BTLE message used in embodiments of the present disclosure;

Figure 2 is a schematic diagram of a system according to aspects of the present disclosure including components according to aspects of the present disclosure and operating according to aspects of the present disclosure;

Figure 3 is a flow chart illustrating operation of a BTLE fixed tag included in the system of Figure 2 according to an embodiment of the present disclosure;

Figure 4 is a flow chart illustrating operation of a BTLE mobile tag included in the system of Figure 2 according to an embodiment of the present disclosure;

Figure 5 is a schematic diagram of fixed tags and a mobile tag interacting with a server according to aspects of the present disclosure including components according to aspects of the present disclosure and operating according to aspects of the present disclosure;

Figure 6 is a flow chart illustrating operation of a receiving device included in the system of Figure 2 according to embodiments of the present disclosure;

Figure 7 is a schematic diagram of a terminal device interacting with a server according to aspects of the present disclosure including components according to aspects of the present disclosure and operating according to aspects of the present disclosure;

Figure 8(a) is an exemplary screen of a display device indicating the general direction of a mobile tag relative to a terminal device according to embodiments of the present disclosure;

Figures 8(b) and 8(c) show the exemplary screen rotated;

Figure 9(a) is an exemplary screen of a display device indicating the general directions of two mobile tags relative to a terminal device according to embodiments of the present disclosure;

Figures 9(b) and 9(c) show the sequential display of the general directions of the two mobile tags;

Figure 10(a) is an exemplary screen of a display device indicating the coordinates of two mobile tags relative to a terminal device according to embodiments of the present disclosure; points are shown in the Cartesian plane;

Figure 10(b) is an exemplary screen of a display device indicating the locations of two mobile tags relative to a reference point shown in Polar coordinates according to embodiments of the present disclosure;

Figure 11(a) is an exemplary screen of a display device indicating the coordinates of multiple mobile tags relative to a terminal device according to embodiments of the present disclosure;

Figures 11(b) and 11(c) show exemplary screens of a display device selectively displaying the coordinates of multiple mobile tags relative to a terminal device according to embodiments of the present disclosure;

Figure 12 is an exemplary screen of a display device indicating the coordinates of mobile tags in multiple areas relative to a terminal device according to embodiments of the present disclosure; and

Figure 13 is an exemplary non-transitory computer readable storage medium according to embodiments of the present disclosure.

Description

Bluetooth Low Energy (BTLE) technology has been proposed to be used in indoor positioning systems for tracking devices. Such systems involve the use of High Accuracy Indoor Positioning (HAIP) which places requirements on hardware infrastructure such as the need for multiple array antennas. Therefore this makes for a simpler and more efficient system that is easier to implement using BTLE hardware technology that is already available in the market. The principles of BTLE are described in the art.

BTLE devices broadcast BTLE messages that are associated with profiles, services and protocols as defined by the BTLE standard. Information is transmitted in a BTLE message in a series of AD structures. Each of these AD structures contain a header portion and a payload portion. The header portion describes the type of data present in the payload portion and the size of said data.

An exemplary BTLE message 100 is shown in Figure 1(a). This BTLE message contains four AD structures: device name 110, device position 120, service 130 and a manufacturer ID 140. Each AD structure has a header portion and a payload portion; the header portion of an AD structure describes the data contained in the payload portion. Such a description may be in the form of the length, size, and/or format of the data in the payload portion. In the example BTLE message shown in Figure 1(a), AD structure 110 has a header portion 105 and a payload portion 106. While the exemplary BTLE message in Figure 1(a) shows four AD structures, it will be understood that a BTLE message may contain any number of AD structures. A BTLE message may contain just one AD structure relating to the device ID.

As the BTLE message can be large, the BTLE message need not be received by a receiving device in entirety; the receiving device has the option to stop receiving the rest of a BTLE message once the AD structures containing the required data of the BTLE message have been received by the receiving device.

A BTLE message transmitted by a first BTLE device (a fixed tag 220) is received by a further BTLE device (a mobile tag 210), which measures a radio parameter associated with the BTLE message. Examples of a radio parameter are Received Signal Strength Indicator (RSSI) or the Bit Error Rate (BER). A BTLE message transmitted from a fixed tag 220 contains at least a secured form of an identifier of the fixed tag.

As the BTLE messages contain data specific to a BTLE device, it can be envisaged that a third party looking to hack or spoof a system of BTLE devices may be able to monitor the BTLE messages transmitted by such devices and replicate and add to the data contained in the BTLE messages thereby breaching such a system of BTLE devices.

A further exemplary composite BTLE message 150 is shown in Figure 1(b). This composite BTLE message is transmitted by a mobile tag 210 which has received BTLE messages from fixed tags 220. Such a composite BTLE message 150 comprises several AD structures 160 to 164 that describes the identifier of the mobile tag as well as secured forms of identifiers of the fixed tags from which the BTLE messages have been received. The RSSI values associated with each of these BTLE messages are also described by AD structures in the composite BTLE message. Thus in the exemplary composite BTLE message 150 depicted in Figure 1(b), the AD structure 160 corresponds to the identifier IDa1 of the mobile tag, AD structures 161 and 163 correspond to the secured form of the identifiers of the fixed tags from which BTLE messages have been received, S_ID1 and S_ID2, and AD structures 162 and 164 correspond to the RSSI values RSSI1 and RSSI2 of the BTLE messages received from the fixed tags, these RSSI values being measured by the mobile tag. Figure 1(b) depicts a situation in which a mobile tag has received BTLE messages from two fixed tags. However a mobile tag may receive BTLE messages from more than two fixed tags. Accordingly, the length of the composite BTLE message will change depending on the number of fixed tags transmitting BTLE messages to the mobile tag. Figure 1(b) also illustrates a composite BTLE message in which the identifier IDa1 of the mobile tag is transmitted. However it will be appreciated that if the secured form of the identifiers of the fixed tags from which BTLE messages have been received (S_ID1 and S_ID2) are unique to the mobile tag such that these secured identifiers are not repeated for another mobile tag in a system of mobile tags, IDa1 need not be transmitted in the composite BTLE message 150.

Figure 2 shows a system 200 according to embodiments of the present disclosure. The system 200 includes a mobile tag 210, two fixed tags 220 and 230, and a receiver device 290. The mobile tag 210 is portable and its location can be tracked.

The composite BTLE message is transmitted by the mobile tag 210 and is received by the receiver device 290. This receiver device 290 maintains a look-up table 272 that relates the identifiers of the fixed tags 220 to the position data associated with each of the fixed tags 220. This position data is pre-stored in the receiver device 290, and will be described in greater detail below. The position data contains data such as the latitude, longitude and floor number associated with the respective fixed tag 220.

With the position data retrieved from the receiver device 290, and the RSSI values associated with each fixed tag 220 contained in the composite BTLE message, the receiver device 290 is able to calculate a locus of the position of the mobile tag 210 relative to the fixed tags 220.

The position of the mobile tag 210 can be better resolved when BTLE messages from more fixed tags 220 are received by the mobile tag 210. This enables the receiver device 290 to calculate a further locus of the position of the mobile tag 210 relative to each additional fixed tag 220. The receiver device 290 then calculates the intersection of these loci to obtain an accurate position of the mobile tag 210 relative to the fixed tags 220.

This calculated position of the mobile tag 210 is then stored in a server 270 of the receiver device 290, which may, in turn, be requested by a terminal device 250 for display of the position of the mobile tag 210.

It will be understood that in order to obtain an accurate calculation of the position of a mobile tag 210 relative to fixed tags 220, at least two of such fixed tags 220 are required. The calculation of the position of a mobile tag 210 is not so accurate if data from only one fixed tag 220 is used.

The receiver device 290 may calculate the locus of the position of the mobile tag 210 relative to the fixed tags 220 using a circle method or a midpoint method.

In the circle method, the RSSI values associated with each of the BTLE messages transmitted from two fixed tags are used to determine radii of two circles, the centres of each being pinned to the location of the respective fixed tag. The locus of the position of the mobile tag relative to the fixed tags is then determined from the area of overlap of these circles.

The midpoint method is a faster method for calculating the position of the mobile tag relative to the fixed tags. This involves using predetermined ratios of RSSI values associated with each BTLE message to calculate intermediate points on lines that join the respective fixed tags. The position of the mobile tag relative to the fixed tags is then determined by calculating the midpoint of the line that connects the intermediate points.

The receiver device 290 may also contain data pertaining to the region which is being mapped, i.e. the region containing the fixed tags and the mobile tags. Such data may identify structures, such as walls, that my further reduce the locus of the position of the mobile tag relative to the fixed tags. Floor plans of the area identifying walls, for example, may be stored in the receiver device; this data is used to further isolate the location of the mobile tag.

It should be noted that while the circle or midpoint methods have been described, other methods of calculating the position of a mobile tag relative to the fixed tags may be used.

The BTLE fixed tags 220 and 230 are based at different locations within a building or complex of buildings and periodically transmit BTLE messages. The BTLE messages transmitted by the fixed tags are advertising messages that include a secure identifier that is unique to the transmitting device (fixed tag) within the building. The secure identifier is a variant of the device identifier ID1 of the fixed tag and may be varying with time (for example, every 1 minute). This therefore generates a group of secure identifiers over a period of time that describe a specific fixed tag. In a system comprising several fixed tags however, any one secure identifier in the group of secure identifiers describing a fixed tag is not duplicated to describe another fixed tag. Thus all secure identifiers in a group of secure identifiers describing a fixed tag is unique to that fixed tag. For example, BTLE messages transmitted from fixed tag 220 at one instant in time t1 contains a secure identifier IDp1_t1 relating to that fixed tag 220. This secure identifier changes over time such that further BTLE messages transmitted from the fixed tag 220 at different time instances may each contain a different secure identifier IDp1_t2, IDp1_t3, IDp1_t4 ... IDp1_t*n* to that of IDp1_t1. However secure identifiers IDp1_t1, IDp1_t2, IDp1_t3, IDp1_t4 ... IDp1_t*n* are unique to fixed tag 220 and will not be transmitted by another fixed tag 230 in a system of fixed tags.

The BTLE messages transmitted by the fixed tags 220 may additionally include position data of the respective fixed tag 220. This position data contains information such as the latitude, longitude and floor number associated with each fixed tag. When the BTLE messages include position data, they are position advertising messages. The BTLE messages transmitted by the fixed tags 220 may alternatively be absent of data indicating the position of the fixed tag 220.

At an instant of time t1, the mobile tag 210 receives BTLE messages transmitted from each of the fixed tags 220 and 230. The mobile tag 210 then measures a radio parameter associated with each received message; this is in the form of an RSSI value, for example. The mobile tag 210 then generates a truncated version of the secure identifier for each of the fixed tags 220, 230 based on the secure identifier of the fixed tags received at that instant in time. This truncated version is shorter than the secure identifier IDp1_t1 and may include the two most significant bits of the secure identifier and two least significant bits of the secure identifier, as will be described in detail below. Thus for fixed tag 220 with secure identifier IDp1_t1 at time t1, the mobile tag 210 generates a truncated secure identifier S_IDp1_t1. Accordingly at any instant of time t*n*, the mobile tag 210 generates a truncated secure identifier S_IDp1_t*n*.

With the truncated secure identifiers of the fixed tags 220 and 230, and the measured RSSI values associated with BTLE messages received from fixed tags 220 and 230, the mobile tag 210 transmits, at time instant t1, a composite BTLE message. This composite

BTLE message takes the form as depicted in Figure 1(b) and contains truncated secure identifiers S_IDp1_t1 and S_IDp2_t1 generated for each of fixed tags 220 and 230, respectively, at time t1, and the RSSI values of the BTLE messages received from the fixed tags 220 and 230. At any instant of time t*n*, the composite BTLE message transmitted by the mobile tag 210 will contain truncated secure identifiers S_IDp1_t*n*, S_IDp2_t*n*, ... S_IDp*i*_t*n*, for a mobile tag that has received BTLE messages from *i* fixed tags. The composite BTLE message may also contain the identifier of the mobile tag.

In this manner, any third parties monitoring the BTLE messages transmitted by the fixed tags 220, 230 or the mobile tag 210 will not be able to duplicate such messages to hack or spoof a position tracking system using such tags. This is because the identifiers transmitted in such BTLE messages are secured, as will be detailed below, thereby making it difficult for the third party to decipher the data contained in the BTLE messages transmitted by any of the fixed or mobile tags contained in such a system.

Composite BTLE messages transmitted from the mobile tag 210 at time t*n* are received by a receiver device 290. The receiver device 290 comprises a server 270 and an access point 280. The access point 280 is an example of a Bluetooth Low Energy receiver or transceiver. The composite BTLE message is received by the access point 280 of the receiver device 290. The access point 280 decodes the data contained in the composite BTLE message and then forwards the decoded data to the server 270.

The server 270 maintains a look-up table 272 that contains the identifiers IDp*i*, secure identifiers IDp*i*_t*n* and truncated secure identifiers S_IDp*i*_t*n* of all the fixed tags 220, 230 in the area being mapped (the area containing *i* fixed tags), at all instances of time t*n*, and their respective position data. This position data contains information such as the latitude, longitude and floor number associated with each fixed tag.

The access point 270 also receives the BTLE messages transmitted by the fixed tags 220 and 230 at all instances of time t*n*. The access point 280 decodes the data contained in these BTLE messages and then forwards the decoded data to the server 270. The BTLE messages transmitted from the fixed tags 220 and 230 contain secure identifiers IDp1_t*n* and IDp2_t*n*, respectively, at any instant of time t*n*. It follows that for a system containing *i* fixed tags, the secure identifiers transmitted by the fixed tags at any instant of time t*n* can be represented as IDp*i*_t*n*. Thus one of the AD structures contained in the exemplary BTLE message shown in Figure 1(a) may be dedicated to IDp*i*_t*n*.

Thus upon receipt of the secure identifiers IDpi_tn contained in the BTLE messages transmitted by the fixed tags, and the truncated secure identifiers S_IDpi_tn contained in the composite BTLE messages transmitted by the mobile tag at any instant of time tn, the server 270 is able to match IDpi_tn with S_IDpi_tn to identify the respective fixed tags. Once the server 270 has identified the fixed tags 220, 230, it is able to obtain the position data corresponding to the fixed tags 220, 230 that have transmitted BTLE messages to the mobile tag 210 at that time instant tn. The server 270 does this by querying the look-up table. The RSSI values associated with these BTLE messages have also been transmitted in the composite BTLE message from the mobile tag 210.

Although the receiver device 290 has been described as comprising a server 270, it will be understood that the server may in fact comprise a cluster of servers.

With the position data of the fixed tags 220, 230 and the RSSI values associated with the BTLE messages transmitted by these fixed tags, the server 270 is able to calculate the position of the mobile tag 210 relative to the fixed tags 220, 230 using any of the calculation methods previously described. Generally speaking, the server 270 uses the RSSI values of the BTLE messages received from the fixed tags 220, 230 and the position data obtained from the look-up table to calculate a locus of the position of the mobile tag 210 relative to each of the fixed tags 220, 230. The intersection of the loci obtained from the fixed tags 220, 230 is then calculated by the server 270. This gives an accurate position of the mobile tag 210 relative to each of the fixed tags 220, 230.

Building configuration data may be stored within the server 270. This data may be in the form of floor plans of or map data relating to the area containing the mobile tag 210 and the fixed tags 220, 230. This building configuration data may also be used in the calculations performed by the server 270 to increase the accuracy or confidence of the calculated position of the mobile tag 210.

As mentioned above, this calculated position of the mobile tag 210 is then stored in a server 270 of the receiver device 290 and may be requested by a terminal device 250 for display of the position of the mobile tag 210.

The terminal device 250 is capable of causing display of the position of the mobile tag 210. The terminal device 250 requests position data from the server 270 at a particular frequency. The server 270 then provides the calculated position data relating to the mobile tag 210 to the terminal device 250. The position of the terminal device 250 is known to the terminal device 250 either by calculating it or by receiving it from the server 270 or another server. The terminal device 250 uses the position data relating to the mobile tag 210 and the terminal device 250 to create image data representative of the position of the mobile tag 210 relative to the terminal device 250. The terminal device 250 then uses this image data to cause display of the position of the mobile tag 210 relative to the position of the terminal device 250.

It will be understood that there may be more than one mobile tag 210 present. In this case the terminal device 250 obtains position data relating to the mobile tags 210 that a user is interested in and creates image data representative of the position of each of said mobile tags 210 relative to the terminal device 250. The terminal device 250 then uses this image data to cause display of the position of the mobile tags 210 relative to the position of the terminal device 250.

Referring again to Figure 2, the mobile tag 210 includes a BTLE module 212, which operates according to the Bluetooth Low Energy standard. Each of the fixed tags 220 and 230 also includes a BTLE module 212 that operates according to the Bluetooth Low Energy standard.

The mobile tag 210 also includes a processor 211. The processor 211 is connected to volatile memory such as RAM 216 by a bus 217. The bus 217 also connects the processor 211 and the RAM 216 to non-volatile memory, such as ROM 214. A BTLE module 212 is coupled to the bus 217, and thus also to the processor 211 and the memories 214, 216. An antenna 218 is coupled to the BTLE module 212, although each may instead have its own antenna. Within the ROM 214 is stored a software application 215. The software application 215 in these embodiments is an application for forming the composite BTLE message and appending the mobile tag identifier to said message.

The mobile tag 210 also includes a source of power 219. The source of power 219 may be for instance a battery such as a coin cell. The source of power 219 powers the BTLE module 212 and any other components of the mobile tag 210. The mobile tag 210 may optionally include a sensor 213 for detecting movement of the mobile tag 210. The sensor 213 may take the form of a tilt switch, for instance.

The mobile tag 210 may take any suitable form. Generally speaking, the mobile tag 210 may comprise processing circuitry 211, including one or more processors, and a storage device 214, 216, comprising a single memory unit or a plurality of memory units. The storage device 214, 216 may store computer program instructions 215 that, when loaded into the processing circuitry 211, control the operation of the mobile tag 210.

The BTLE module 212 may take any suitable form. Generally speaking, the BTLE module 212 of the mobile tag 210 may comprise processing circuitry, including one or more processors, and a storage device comprising a single memory unit or a plurality of memory units. The storage device may store computer program instructions that, when loaded into the processing circuitry, control the operation of the BTLE module 212.

The BTLE module 212 includes a communication stack that is implemented at least partly in software using processor and memory resources (not shown), all of which are included within the BTLE module 212. The BTLE module 212 is configured, when enabled by the processor 211 running application 215, to measure the RSSI values of the BTLE messages received by the mobile tag and to create a composite BTLE message containing at least the secure identifiers of the fixed tags from which BTLE messages have been received, and the RSSI values of the BTLE messages received from the fixed tags. The composite BTLE message is then transmitted from the mobile tag 210 via the antenna 218.

The BTLE module 212 of the mobile tag 210 is both a transmitter and a receiver.

Each of the fixed tags 220 and 230 includes a BTLE module 222, an antenna 228, a source of power 229, a processor 221, RAM 226, ROM 224 containing computer readable instructions and a bus 227, which are constituted and connected in any suitable way. The ROM 224 of each of the fixed tags 220 and 230 also stores information 225. The information 225 includes the location of the fixed tag 220 and 230, a unique tag identifier (IDp1) and a unique counter value (CNT1). Information 225 may be written to each of the fixed tags 220 and 230 using an external tool 260 connected to each fixed tag 220 and 230 through the BTLE module 222 thereof.

Each of the fixed tags 220 and 230 may take any suitable form. Generally speaking, these devices may comprise processing circuitry, including one or more processors, and a storage device, comprising a single memory unit or a plurality of memory units. The storage device may store computer program instructions that, when loaded into the processing circuitry, control the operation of the fixed tag 220 and 230 (such as the frequency of transmission of BTLE messages).

Each of the fixed tags 220 and 230 may be configured to operate in the same way. However these devices may differ in that the information 225 stored in the ROM 224 includes a different identifier for each of the fixed tags 220 and 230. The identifier of each of the fixed tags 220 and 230 is included in the BTLE messages transmitted from each of the fixed tags 220 and 230.

The source of power 229 may be for instance a battery such as a coin cell. The source of power 229 powers the BTLE module 222 and any other components of each of the fixed tags 220 and 230.

The BTLE module 222 of each of the fixed tags 220 and 230 is a transmitter.

The receiver device 290 comprises the server 270 and the access point 280. The server includes a processor 271. The processor 271 is connected to volatile memory such as RAM 276 by a bus 277. The bus 277 also connects the processor 271 and the RAM 276 to a non-volatile memory, such as ROM 274. A software application 275 is stored within the ROM 274. The software application 275 is an application for looking up the position data corresponding to the respective fixed tags 220 and 230 from a look-up table 272, and calculating the position of the mobile tag 210. The server also has a communication interface 273 connected to the processor 271 via a bus 277.

The communication interface 273 may be configured to allow two-way communication with external devices and/or networks. The communication interface 273 may be configured to communicate wirelessly via one or more of several protocols such as Global System for Mobile Communications (GSM), Code Division Multiple Access (CDMA), Universal Mobile Telecommunications System (UMTS) and IEEE 712.11 (Wi-Fi). Alternatively or additionally, the communication interface 273 may be configured for wired communication with a device or network.

The communication interface 273 sends the calculated position of the mobile tag 210 stored in the server 270 to a terminal device 250 for display of the position of the mobile tag 210 when requested.

The terminal device 250 (shown in Figure 7) includes a processor 251. The processor 251 is connected to volatile memory such as RAM 256 by a bus 257. The bus 257 also connects the processor 251 and the RAM 256 to a non-volatile memory, such as ROM 254. A communications interface 252 is coupled to the bus 257, and thus also to the processor 251 and the memories 254, 256. A BTLE module 540 is coupled to the bus 257, and thus also to the processor 251 and the memories 254, 256. An antenna 542 is coupled to the BTLE module 540. A software application 255 is stored within the ROM 254. The software application 255 is an application for causing a display of the mobile tag 210 relative to a reference point on display 258, although the display may not be integral to the terminal device 250 and may take some other form.

The terminal device 250 may optionally include a magnetometer 253 for determining the orientation of the terminal device 250 with respect to the magnetic North.

The communication interface 252 may be configured to allow two-way communication with external devices and/or networks. The communication interface 252 may be configured to communicate wirelessly via one or more of several protocols such as Global System for Mobile Communications (GSM), Code Division Multiple Access (CDMA), Universal Mobile Telecommunications System (UMTS) and IEEE 712.11 (Wi-Fi). Alternatively or additionally, the communication interface 252 may be configured for wired communication with a device or network.

The communication interface 252 sends requests to the server 270 for position data relating to specific mobile tags 210 as specified in the request. Upon receipt of such data, the terminal device 250 causes display of the position of the mobiles tag 210.

The terminal device 250 may be a mobile phone, a smart phone, a tablet computer, a laptop computer, a camera, an mp3-player, or equipment integrated within vehicles, etc.

Returning to Figure 2, the access point 280 is a Bluetooth transceiver and is in communication with the server 270. The transceiver is a discrete device that may be physically located within the Bluetooth range of the mobile tag 210 and is capable of receiving a composite BTLE message from the mobile tag 210. The transceiver is additionally connected to the server 270 by a data link (e.g. a wired Ethernet cable or a WiFi connection). Thus, subsequent to receiving the composite BTLE message from the mobile tag 210, the transceiver forwards the data contained in the composite BTLE message to the server 270.

The access point 280 has an antenna 288 and is configured to receive composite BTLE messages from the mobile tag 210. The access point 280 then forwards the composite BTLE messages to the server 270. Although only one access point 280 is shown coupled to the server 270 in Figure 2, it should be noted that it is within the scope of the present disclosure that several such access points 280 may be coupled to the same server 270.

As BTLE messages are used to calculate the position of the mobile tag 210, the use of a wired network is not required. Further, as the calculation is done within the server 270, the computing burden on the processor 211 of the mobile tag 210 is significantly reduced, thereby extending the battery life of the power source 219 of any of the tags (such as the mobile tag 210).

BTLE messages may be transmitted by each of the fixed tags 220 and 230 periodically, for instance at 4Hz (250 millisecond intervals) or at intervals defined by some component within the system. They may alternatively be transmitted on request of some component within the system.

In this specification, the terms ‘message’ and ‘packet’ are used interchangeably since they are intrinsically linked.

The server 270 calculates the position of the mobile tag 210 with the same periodicity with which the composite BTLE messages are received from the mobile tag 210. The composite BTLE messages are transmitted by the mobile tag 210 periodically, for example every second. However this periodicity increases if the mobile tag 210 is moving. Motion of the mobile tag 210 is detected by the sensor 213.

Upon receipt of the composite BTLE message from the access point 280, the server 270 uses the secure identifiers IDp*i*_t*n* and the truncated secure identifiers S_IDp*i*_t*n* relating to the fixed tags 220, 230 at any time instance t*n* to identify the respective fixed tag, and to retrieve position data relating to the identified fixed tag. This position data is held in the look-up table 272 contained within the ROM 274 of the server 270.

Such a look-up table contains at least the identifiers IDp*i* of all fixed tags contained within the area covered by the server 270, the secure identifiers IDp*i*_t*n* and the truncated secure identifiers S_IDp*i*_t*n* relating to each fixed tag at any time instant t*n*, and their corresponding position data. The secure identifiers and the truncated secure identifiers may be stored in hexadecimal bits. The position data may take any form, such as Cartesian coordinates, Polar coordinates, or simply may just associate each fixed tag with its latitude *x*, longitude *y* and floor number *z*. The data contained in the look-up table 272 may be pre-programmed into the server ROM 274 by external means.

An exemplary look up table is shown in Table 1 which shows position data corresponding to four fixed tags with identifiers IDp1, IDp2, IDp3 and IDp4 and their respective secure identifiers and truncated secure identifiers obtained at three time instances t1, t2 and t3.

Time	Secure Identifier	Truncated Secure Identifier	Identifier/ Counter	Latitude	Longitude	Floor no
t1	FFFF12F409E451ACB67F	FF7F	IDp1/ FFFF	7.063	319.14	1
t2	FFFFF4193E6B35108B2E	FF2E				
t3	FFFF13279691406337E8	FFE8				
t1	4B8FF8125FF1B2840E3A	4B3A	IDp2/ 4B8F	7.178	318.83	1
t2	4B8FC14A084C4BAF2A62	4B62				
t3	4B8F095ECB4817ED568F	4B8F				
t1	D5001D333CBCD2E8F424	D524	IDp3/ D500	6.931	319.64	2
t2	D50051B41EDD3770B0E2	D5E2				
t3	D5006C8059B6A0496156	D556				
t1	BBD845FCDEA50D887900	BB00	IDp4/ BBD8	6.897	319.96	1
t2	BBD84D6C96E52F0EF55D	BB5D				
t3	BBD8D4325FB4213B301C	BB1C				
t*n*	IDp*i*_t*n*	S_IDp*i*_t*n*	IDp*i*/CNT*i*	*x*	*y*	*z*

Table 1

The processor 271 then uses the position data associated with each of the fixed tags 220 and 230 identified in the composite BTLE message, and the respective measured RSSI values, to calculate the position of the mobile tag 210 using the methods as previously described. The processor 271 may optionally incorporate the features of the area covered by the server 270 to further narrow the calculated locus of the position of the mobile tag 210.

If there are more than two fixed tags, detection of motion of the mobile tag 210 using the BTLE messages received from the fixed tags 220, 230 is made, regardless of whether the mobile tag 210 has a motion sensor 213. If the RSSI measurements in all BTLE messages received from the more than two fixed tags 220, 230 are the same or substantially the same at each period except for the messages from one of the more than two fixed tags 220, 230, the mobile tag 210 is determined to be not moving. The difference in message from the one fixed tag of the more than two fixed tags may be due to attenuation of the signal from that fixed tag (due to, for example, a person walking through the path between the fixed tag in question and the mobile tag).

The computer program instructions 215 in the mobile tag 210 may provide the logic and routines that enables the mobile tag 210 to perform the functionality described. The computer program instructions 215 may be pre-programmed into the mobile tag 210.

The processing circuitry 211, 221, 271 may be any type of processing circuitry. For example, the processing circuitry may be a programmable processor that interprets computer program instructions and processes data. The processing circuitry may include plural programmable processors. Alternatively, the processing circuitry may be, for example, programmable hardware with embedded firmware. The processing circuitry or processor 211, 221, 271 may be termed processing means.

Typically, the BTLE modules 212 and 222 each comprise a processor connected to both volatile memory and non-volatile memory. The computer program is stored in the non-volatile memory and is executed by the processor using the volatile memory for temporary storage of data or data and instructions.

The term 'memory' when used in this specification is intended to relate primarily to memory comprising both non-volatile memory and volatile memory unless the context implies otherwise, although the term may also cover one or more volatile memories only, one or more non-volatile memories only, or one or more volatile memories and one or more non-volatile memories. Examples of volatile memory include RAM, DRAM, SDRAM etc. Examples of non-volatile memory include ROM, PROM, EEPROM, flash memory, optical storage, magnetic storage, etc.

Each BTLE module 212, 222 may be a single integrated circuit. Each may alternatively be provided as a set of integrated circuits (i.e. a chipset). The BTLE modules 212, 222, may alternatively be hardwired, application-specific integrated circuits (ASIC).

The fixed tags 220, 230 are distributed around a building or premises covered by server 270. For instance these devices may be located at various points in a hospital or warehouse so as to provide sufficient coverage of the premises. While the above description details two fixed tags 220, 230, this is a minimum number of fixed tags required and any number of such devices can be employed.

Embodiments of the present disclosure provide a scheme whereby a BTLE mobile tag 210 is located by a terminal device 250, with the aid of at least two fixed tags 220, 230, without the need for a network. The terminal device 250 causes a display of the mobile tag 210 relative to the position of the terminal device 250, so as to inform or guide a user. In relation to the following description, it will be understood that there may be more than one mobile tag 210 present.

A method 300 according to the present disclosure will now be described with respect to Figure 3 and mostly with reference to the fixed tag 220. The steps carried out by the fixed tag 220 are performed by the processor 221 using the RAM 226 under control of the software application 225 stored in the ROM 224.

The aim of method 300 is to encode the identity of the fixed tag by generating a series of secure identifiers that are unique to the fixed tag. The secure identifiers are each generated using a dynamic variable α stored within the fixed tag, and a constant β specific to that particular fixed tag. The dynamic variable may be time dependent. A function *f* then maps an instant value of the dynamic variable α, taken at time t1, for example, and the constant β to the secure identifier IDp1_t1. Thus,

$$\text{IDp1\_t1} = f(\alpha(t1), \beta).$$

While it has been described that the dynamic variable α is time dependent, it will be appreciated that this dependency may be based on any other variable γ maintained within the fixed tag that does not have a constant value. For example, α may be dependent on a random number generator, an internal counter, an internal clock value, a timestamp TS1, etc. In the case where α is a timestamp TS1, this value may be obtained by the fixed tag using a Network Time Protocol. Thus the value of α obtained at one instant of γ ($\gamma1$) is different to the value of α obtained at another instant of γ ($\gamma2$), i.e. $\alpha(\gamma1) \neq \alpha(\gamma2)$.

The constant β may take on a single value within the fixed tag, or may comprise several constant values. In an exemplary embodiment, the constant β may comprise a static counter value CNT1 and a fixed tag identifier IDp1, i.e. β(CNT1, IDp1).

The function *f* is any function that maps dynamic variables and constants to a further dynamic value. Function *f* may be any one-to-one function. In an exemplary embodiment, the function *f* may comprise, in part, an encryption function based on the Advanced Encryption Standard.

The method 300 starts at step S1 where the processor 221 obtains a timestamp TS1 at time t1 from its internal clock. The processor 221 may obtain such a timestamp using a Network Time Protocol.

At step S2, the processor 221 then generates a secure identifier IDp1_t1 using the device identifier IDp1 of the fixed tag 220, a counter CNT1 associated with the fixed tag 220, and the timestamp TS1. The counter CNT1 is a fixed number for each fixed tag 220. In the example shown in Table 1, the counter CNT1 has the hexadecimal value **FFFF** for the fixed tag with device identifier IDp1. The secure identifier IDp1_t1 is generated using a time dependent function in which:

$$\text{IDp1\_t1} = \text{CNT1} \mid\mid \text{AES}(\text{TS1}, \text{IDp1}),$$

where AES is an Advanced Encryption Standard number generator based on TS1 and ID1, and || denotes the concatenation of CNT1 and the number generated by the AES. The secure identifiers of the fixed tags are each 48 bits in length.

In the example shown in Table 1, the secure identifier IDp1_t1 for the fixed tag with device identifier IDp1 at time t1 is generated by concatenating a counter value CNT1 of **FFFF** with the output **12F409E451ACB67F** of an AES function using TS1 and IDp1. As the dynamic variable TS1 generates a unique value at each time instant t, the output of the AES function will also be unique at each time instant t. Thus the secure identifiers generated at each time instance are unique and distinct to those generated at another time instance. This can be seen in Table 1 where the generated secure identifiers at time instances t1, t2 and t3 are distinct from each other.

The method then progresses to step S3 in which the processor 221 uses the generated secure identifier IDp1_t1 to create a BTLE message. This BTLE message takes the form as shown in Figure 1(a) and may contain other data in addition to the secure identifier IDp1_t1. The BTLE message is transmitted by the BTLE module 222 of the fixed tag 220 via antenna 228. Transmission of the BTLE messages by the BTLE module 222 may be performed at any suitable frequency.

At step S4, the method 300 determines if the fixed tag 220 has been turned off. If the tag 220 has not been turned off, the method 300 loops back to step S1 and the process repeats itself at a new time t2 (thereby generating another secure identifier IDp1_t2). If the fixed tag 220 has been turned off, the device stops transmitting BTLE messages and the method ends.

The method 300 is a two-fold method of encoding the fixed tag identifier, and therefore eliminates replay attacks from third parties since the (varying) time stamp TS1 is used in the encryption of the fixed tag identifier IDp1; this encrypted identifier is then further encoded by concatenation with a counter CNT1 specific to the fixed tag. As a two-fold method is used, this stops rogue devices having knowledge of the correct time stamp TS1 from deciphering the content of the BTLE message transmitted by the fixed tag.

A second method 400 according to the present disclosure will now be described with respect to Figure 4, and mostly with reference to the mobile tag 210. The steps carried out by the mobile tag 210 are performed by the processor 211 using the RAM 216 under control of the software application 215 stored in the ROM 214. Steps performed by the fixed tags 220 and 230 involve the corresponding components.

The method 400 starts at step S1. Here the mobile tag 210 scans for BTLE messages that are transmitted by any of the fixed tags 220 and 230 as described above. The software application 215 stored in ROM 214 of the mobile tag 210 specifies the manner in which the scanning operation takes place, the frequency of scanning, and the width of the scanning window. Within the scanning window, the processor 211 instructs the BTLE module 212 in the mobile tag 210 to detect BTLE messages that impinge upon the antenna 218.

At step S2, the BTLE module 212 receives BTLE messages that are transmitted by BTLE devices within the scanning window. The processor 112 then determines if the received messages were transmitted by any of the fixed tags 220 and 230. Here the BTLE module 212 in the mobile tag 210 determines the value of the secure identifier contained in a received BTLE message by checking its header section. The BTLE module 212 then passes this value to the processor 211. The processor 211, using software application 215, compares this value with parameters set by the software application 215. Only BTLE messages transmitted with secure identifiers that correspond to ‘allowed’ fixed tags 220 and 230 are received and subsequently stored in the RAM 216.

At step S3, when the processor 211 confirms that a received BTLE message has been transmitted by an 'allowed' fixed tag 220 and 230, it uses a function g to map a secure identifier IDp1_t1 contained in the received BTLE message to a further secure identifier S_IDp1_t1. Thus,

$$S\_IDp1\_t1 = g(IDp1\_t1)$$

where the function g may be any one-to-one or many-to-one function. The function g may be a scrambling function, a concatenation function, a truncation function, a masking function, an obfuscating function, etc. The function g may also be a combination of said functions. In this manner, the secure identifier IDp1_t1 is further encoded into the further secure identifier S_IDp1_t1, thereby providing multi-layered encoding.

In step S3, the processor 211 generates a truncated secure identifier S_IDp1_t1 at time instant t1 using the secure identifier IDp1_t1 contained in the received BTLE message. In an exemplary embodiment, the truncated secure identifier S_IDp1_t1 is generated using a predetermined number p of the most significant bits of the secure identifier IDp1_t1 concatenated with a further predetermined number q of the least significant bits of the secure identifier IDp1_t1, i.e.:

$$S\_IDp1\_t1 = \mathrm{LEFT}\ (IDp1\_t1, p)\ ||\ \mathrm{RIGHT}\ (IDp1\_t1, q),$$

where LEFT (IDp1_t1, p) gives the p most significant bits of the secure identifier IDp1_t1, RIGHT (IDp1_t1, q) gives the q least significant bits of the secure identifier IDp1_t1, and || denotes the concatenation of the aforementioned bits.

Predetermined numbers p and q have a value of 2, however these numbers may take on any value. In the example shown in Table 1, the secure identifier IDp1_t1 has a hexadecimal value of **FFFF12F409E451ACB67F** for the fixed tag with device identifier IDp1 at time t1. Concatenating the two most significant bits and the two least significant bits of IDp1_t1 gives a truncated secure identifier S_IDp1_t1 of **FF7F**.

It must be mentioned that the secure identifiers IDp1_t1 and the truncated secure identifiers S_IDp1_t1 for each fixed tag is unique to that specific fixed tag. Thus, in the example given in Table 1, values in the group of IDp1_t1 and S_IDp1_t1 for the fixed tag with identifier IDp1 are not repeated for any other fixed tag. Understandably in a system with a large number of fixed tags 220, 230, p and q may each take on values in excess of 2 to provide for a larger set of resulting truncated secure identifiers. For example a system containing a large number of fixed tags may be implemented in which p may have a value of 2 while q may have a value of 4 or 5. Such a system would therefore provide for a larger array of secure identifiers S_IDpi_tn that do not overlap.

At step S4, the processor 211 instructs the BTLE module 212 to record a radio parameter associated with a received BTLE message. This parameter is the RSSI value. The received BTLE message, the associated RSSI value, and the truncated secure identifier corresponding to the fixed tag from which the BTLE message was transmitted are stored in the RAM 216 of the mobile tag 210.

The processor 211 instructs the BTLE module 212 to scan for BTLE messages transmitted from at least three fixed tags 220 and 230.

At step S5, the processor 211 uses the truncated secure identifiers obtained from the fixed tags 220 and 230, and the RSSI values associated with the BTLE messages received from the fixed tags 220 and 230 to create a composite BTLE message. The identifier of the mobile tag 210 may also be transmitted in the composite BTLE message. This composite BTLE message takes the form as shown in Figure 1(b) and is transmitted by the BTLE module 212 of the mobile tag 210 via antenna 218. Transmission of the composite BTLE messages by the BTLE module 212 may be performed at any suitable frequency.

At step S6, the method 400 determines if the mobile 210 has been turned off. If the tag 210 has not been turned off, the method 400 loops back to step S1 and the process repeats itself at a new time t2 (thereby generating another truncated secure identifier S_IDp1_t2 for the fixed tag with identifier IDp1). If the mobile tag 210 has been turned off, the device stops transmitting composite BTLE messages and the method ends.

A third method 600 according to the present disclosure will now be described with respect to Figures 5 and 6, and mostly with reference to the system 500 as shown in Figure 5. The steps carried out by the receiver device 290 are performed by the server 270 and the access point 280; in the server 270, the steps are carried out by the processor 271 using the RAM 276 under control of a software application 275 stored in the ROM 274. Steps performed by the mobile tag 210 and by fixed tags 220 and 230 involve the corresponding components.

System 500 in Figure 5 shows two fixed tags 220, 230 transmitting BTLE messages BTLE1 and BTLE2, respectively, which are received by a mobile tag 210 and by a receiver device 290. BTLE1 contains the secure identifier IDp1_t*n* while BTLE2 contains secure identifier IDp2_t*n*. The mobile tag 210 transmits a composite BTLE message BTLE3 which is received by the receiver device 290. BTLE3 contains truncated secure identifiers S_IDp1_t*n* and S_IDp2_t*n* corresponding to fixed tags 220 and 230, and RSSI values RSSI1 and RSSI2 associated with BTLE1 and BTLE2 as received by the mobile tag 210. While the system only shows two fixed tags, it will be appreciated that any number of fixed tags in excess of two may be used, each transmitting their secure identifiers in a BTLE message according to the methods as previously described. The composite BTLE message transmitted by the mobile tag would therefore be longer and contain truncated secure identifiers corresponding to the number of fixed tags in the system. Further, while the system 500 only shows one mobile tag, it will be appreciated that any number of mobile tags may be present, each operating according to the methods previously described.

Method 600 begins at step S1. As with previously described method 400, the receiver device 290 scans for BTLE messages that are transmitted by BTLE devices. These devices could be any of the mobile tag 210 and the fixed tags 220 and 230. The software application 275 stored in ROM 274 of the receiver device 290 specifies the manner in which the scanning operation takes place, the frequency of scanning, and the width of the scanning window. Within the scanning window, the processor 271 within the server 270 accesses the access point 280 to detect any BTLE messages that impinge upon the antenna 288.

The server 270 receives BTLE messages that are generally transmitted by BTLE devices within the scanning window. The processor 271 then determines the source of the BTLE messages. The processor 271 determines the value of an identifier contained in a received BTLE message by checking its header section. The processor 271, using software application 275, compares this value with parameters set by the software application 275. Only BTLE messages transmitted with secure identifiers or truncated secure identifiers that correspond to 'allowed' BTLE devices are received and subsequently stored in the RAM 276.

At step S2, the BTLE messages BTLE1 and BTLE2 transmitted by the fixed tags 220, 230 are received by the access point 280 via the antenna 288. These BTLE messages are then transmitted to the server 270 for action by the processor 271.

Similarly, at step S3, the composite BTLE message BTLE3 transmitted by the mobile tag 210 is received by the access point 280 via the antenna 288. This composite BTLE message is also transmitted to the server 270 for action by the processor 271.

As BTLE messages BTLE1, BTLE2 and BTLE3 contain secure identifiers and truncated secure identifiers relating to fixed tags 220, 230 and mobile tag 210, any third parties monitoring BTLE messages transmitted by the fixed tags 220, 230 or the mobile tag 210 will not be able to duplicate such messages to spoof the position tracking system. This is because the identifiers transmitted in such BTLE messages are secured using multi-layered encoding as described above, thus making it difficult for the third party to decipher the data contained in the BTLE messages transmitted by any of the fixed or mobile tags as knowledge of at least TS1, CNT1 and ID1 are required to decipher the secure identifiers and the truncated identifiers.

At step S4, when the processor 271 confirms that a received BTLE message has been transmitted by an 'allowed' BTLE device (i.e. an allowed mobile tag 210 or an allowed fixed tag 220, 230), the processor 271 decodes the received BTLE messages by scanning for the secure identifiers (in the case of BTLE messages received from fixed tags 220, 230) or the truncated secure identifiers (in the case of composite BTLE messages received from mobile tags 210) that correspond to tags 210, 220 and 230 at regular time intervals, such as every minute. These values are saved in the RAM 276 of the server 270.

The processor 271 then identifies the respective fixed tags that have sent BTLE messages BTLE1 and BTLE2 to the mobile tag 210 in step S5. Here the processor 271 compares the truncated secure identifier S_IDp*i*_t*n* obtained from the composite BTLE message BTLE3 with the secured identifiers IDp*i*_t*n* obtained from BTLE messages BTLE1 and BTLE2 at any instant of time t*n*. This is done using the look-up table 272 stored in ROM 274 of the server 270. The look-up table 272 contains the values as shown in Table 1 above; in particular, the look-up table contains the tag identifier IDp*i* and counter value CNT*i* associated with each fixed tag 220, 230. The look-up table also contains historical data of IDp*i*_t*n* for past time intervals. For any one fixed tag 220, 230, the processor 271 identifies the fixed tag by matching the bits of S_IDp1_t1 with IDp1_t1 using the counter values CNT*i* in the look-up table.

In step S6, once the fixed tag has been identified, the processor 271 accesses the look-up table stored within the ROM 274 to obtain the position data associated with the identified fixed tag. This position data is then momentarily pushed to the RAM 276 of the server 270.

If there are truncated secure identifiers corresponding to more than two fixed tags 220, 230 in the composite BTLE message, the processor 271 then identifies these further fixed tags, fetches the corresponding position data from the look-up table 272, and pushes the position data corresponding to these fixed tags to the RAM 276.

At step S7, the processor 271 accesses the RAM 276 and uses the position data contained therein, together with the RSSI values transmitted in the composite BTLE message, to calculate the position of the mobile tag 210. This calculation is performed using any of the methods described above. If more than two fixed tags are present, the processor 271 only uses the position data in the RAM 276 that corresponds to the three fixed tags that have the highest RSSI values for the calculation.

If there is more than one mobile tag 210 transmitting composite BTLE messages which have been consequently received by access point 280 of the receiver device 290, the processor 271 continues to calculate the position of all the mobile tags 210.

The calculated position of each mobile tag 210, together with the mobile tag identifier, is stored in the RAM 276 of the receiver device 290.

The method then optionally proceeds to step S8. In this step, a terminal device 250 requests for position data corresponding to specific mobile tags 210 from the server 270 for display of the position of the mobile tags 210. This request is made by the communications interface 252 of the terminal device 250 when initiated by a user. The request is a two-step process in which the server 270 is initially queried to ascertain the quantity of position data held within RAM 276. This informs the user of the position data available, and may be presented to the user as a list of mobile tag identifiers (as previously described with reference to Figure 1(a)). The user then selects the mobile tag(s) as required and a further request is sent to the server 270, this time specifying the identifiers of the mobile tags 210 that the user is interested in. The processor 271 in server 270 then fetches the required position data from the RAM 276 and returns this data to the terminal device 250.

The terminal device 250 then calculates its position in the following manner.

The terminal device 250 scans for BTLE messages that are transmitted by the fixed tags 220 and 230. The software application 255 stored in ROM 254 of the terminal device 250 specifies the manner in which the scanning operation takes place, the frequency of scanning, and the width of the scanning window. Within the scanning window, the processor 251 instructs the BTLE module 540 to detect position advertising messages that impinge upon the antenna 542. The RAM 256 also contains a similar look-up table as that contained in server 270; this look-up table however only relates to the fixed tags 220, 230. An exemplary look-up table contained in RAM 256 of the terminal device 250 is shown in Table 2. Such a table is maintained by the software application 255, and may be updated with new values by external means.

Time	Secure Identifier	Identifier/ Counter	Latitude	Longitude	Floor no
t1	**FFFF12F409E451ACB67F**				
t2	**FFFFF4193E6B35108B2E**	IDp1/ **FFFF**	7.063	319.14	1
t3	**FFFF13279691406337E8**				
t1	**4B8FF8125FF1B2840E3A**				
t2	**4B8FC14A084C4BAF2A62**	IDp2/ **4B8F**	7.178	318.83	1
t3	**4B8F095ECB4817ED568F**				
t1	**D5001D333CBCD2E8F424**				
t2	**D50051B41EDD3770B0E2**	IDp3/ **D500**	6.931	319.64	2
t3	**D5006C8059B6A0496156**				
t1	**BBD845FCDEA50D887900**				
t2	**BBD84D6C96E52F0EF55D**	IDp4/ **BBD8**	6.897	319.96	1
t3	**BBD8D4325FB4213B301C**				
t*n*	IDp*i*_t*n*	IDp*i*/CNT*i*	*x*	*y*	*z*

Table 2

The BTLE module 540 receives messages that are transmitted the fixed tags 220 and 230. The BTLE module 540 in the terminal device 250 determines the value of a secure identifier contained in a received BTLE message by checking its header section. The BTLE module 540 then looks-up the identity of the fixed tag from which the BTLE message has been received, and retrieves its position data. The processor 251, using software application 255, compares the identity of the fixed tag with parameters set by the software application 255. Only messages from ‘allowed’ BTLE fixed tags are received and subsequently stored in the RAM 256.

Once the processor 251 confirms that a received BTLE message has been transmitted by an ‘allowed’ BTLE fixed tag, it instructs the BTLE module 540 to record a radio parameter associated with said received message. This radio parameter is the RSSI value of the received message. The retrieved position data, the associated RSSI value, and the unique secure identifier of the fixed tag from which the message was transmitted are stored in the RAM 256 of the terminal device 250.

While the foregoing description relates to two fixed tags, any number of fixed tags may be present in an area. In general, at least three fixed tags will be present in an area covered by the server.

The processor 251 uses the retrieved position data and the measured RSSI values corresponding to each received BTLE message, to calculate the position of the terminal device 250 relative to the fixed tags from which the BTLE messages originated.

If there are more than three fixed tags transmitting messages which have been consequently received by the BTLE module 540 of the terminal device 250, the processor 251 may only use the position data stored in the RAM 256 that corresponds to the three BTLE messages that have the highest RSSI values for the calculation.

The data corresponding to the position of the terminal device 250 is stored in the RAM 256 of the terminal device 250.

The terminal device 250 then uses the position data for the mobile tag(s) transmitted by the server 270 to create image data representative of the position of the mobile tag(s) 210 relative to the terminal device 250. The terminal device 250 uses this image data, and the data corresponding to the position of the terminal device 250, to cause display of the position of the mobile tag(s) 210 relative to the position of the terminal device 250.

Figure 7 shows several areas 520 (zone A) and 530 (zone B), these areas being mapped by a terminal device 250. The terminal device 250 displays the location of one of more mobile tags in each of the areas 520 and 530. The display of the mobile tags in these locations may be done by any method as previously discussed. The server 270 calculates the position of each of mobile tags 210A and 210B relative to the respective fixed tags using the methods as previously described: the position of mobile tag 210A is calculated using data corresponding to fixed tags 220A, 230B and 240C; and the position of mobile tag 210B is calculated using data corresponding to fixed tags 220B, 230B and 240B.

The terminal device 250 displays the position of mobile tag 210A relative to the position of the terminal device 250A in area 520. In addition to this, terminal device 250A may be configured to additionally display the position of mobile tags 210B in area 530.

To do this, the terminal device 250 sends a request for mobile tag data to the server 270. The server queries its RAM 276 and returns a list of mobile tags, the calculated position data of which is presently stored in the RAM 276; this position data is stored in the RAM 276 as position vectors. The user then selects from the list the mobile tag(s) that it wishes to track. The communication interface 252 of the terminal device 250 then sends a request to the server 270 to obtain the position vectors corresponding to a selected list of mobile tags.

The processor 251 uses the data associated with each position vector to enable the display of the mobile tag(s) relative to the position of the terminal device 250. This may be in the form of image data that is output from the processor 251 to a display 258. The image data therefore contains information relating to the location of the terminal device 250 and the relative locations of the mobile tag(s) 210 at that instant. This image data is updated at regular intervals where any new position data corresponding to the position of the mobile tag 210 or the position of the terminal device 250 is re-calculated using the methods described above, and updated image data is then output to the display 258. The software application 255 specifies the frequency with which the updating occurs.

Figure 8(a) shows an exemplary terminal device 250 with a display 258. The terminal device 250 may be a tablet computer, a laptop computer, a smart phone, or equipment integrated within vehicles. The display 258 comprises a screen 610 and a plurality of input buttons 620.

The display 258 is configured to receive the image data generated by the processor 251. The display 258 graphically represents the received image data. This enables a user to view the location of one or more mobile tags 210 relative to the terminal device 250.

This graphical representation may be in the form of an indicated general direction of the mobile tag(s) 210 from the terminal device 250, or specific coordinates from the reference point. These coordinates may be Cartesian coordinates or Polar coordinates.

The simplest form of indication involves the display 258 presenting the user with an arrow 630 to indicate the direction in which a mobile tag 210 is located relative to the terminal device 250, as shown in Figure 8(a). Here the image data received by the display 258 is representative of an area 650 containing a terminal device 250 and one mobile tag 670. The area 650 is a physical space which contains the mobile tag(s) and the terminal device. The arrow 630 only indicates the direction of the mobile tag 670 in relation the terminal device 250. This graphical representation shows the bearing of the mobile tag 670 from the terminal device 250; no distance is indicated. When the display 258 is rotated (such as in Figures 8(b) and 8(c)), the angular placement of the arrow 630 relative to the terminal device 250 does not change. This assumes that the mobile tag 670 has not changed position as the display 258 is rotated as in Figures 8(b) and 8(c).

If more than one mobile tag is present in the area, the display either displays two arrows on its display, as shown in Figure 9(a), or plays an animation of a first arrow followed by a second arrow, as shown in Figures 9(b) and 9(c). Here the imagining data received by display 258 is representative of area 750 containing a terminal device 250 and two mobile tags 770 and 775. In Figure 9(a), the display 258 displays two arrows 730 and 735. Arrow 730 indicates the direction of the mobile tag 770 in relation to the terminal device 250, while arrow 735 indicates the direction of the mobile tag 775 in relation to the terminal device 250.

Alternatively, the display may display the two arrows sequentially. This is shown in Figures 9(b) and 9(c) in relation to the area 750 as described above. Here display 258 first displays arrow 730, in a first instance, as shown in Figure 9(b). Arrow 730 indicates the direction of the mobile tag 770 in relation to the terminal device 250. The display 258 then sequentially displays arrow 735, in a second instance, as shown in Figure 9(c). Arrow 735 indicates the direction of the mobile tag 775 in relation to the terminal device 250.

The graphical representation on the display could also take the form of specific coordinates. Here both the direction and distance of the mobile tag(s) from the terminal device 250 are indicated on the display 258. Figure 10(a) shows the display of two mobile tags 210 represented as crosses 820 and 830 and a terminal device 250 represented as a square 810 on a display 258 in a Cartesian coordinate system. In Cartesian coordinates, the latitude and longitude of the mobile tag(s) relative to the terminal device 250 is visually presented on the screen 805 of the display 258. The square 810 indicates the position of the terminal device 250, while the crosses 820 and 830 indicate the relative position of the respective mobile tags 210 with respect to the position of the terminal device 250. A user holding a terminal device 250 is therefore instantly graphically provided with location information of the two mobile tags 210 represented by crosses 820 and 830 in relation to the user's position (i.e. the position of the terminal device 250), thereby enabling the user to find his or her direction within an area.

Figure 10(b) illustrates a display of the location of two mobile tags 210 represented as arrows 860 and 870, and the location of the terminal device 250 represented as a reference marker 880, on the display 258 in a Polar coordinate system. The reference marker 880 in this case is a compass point indicating North as determined from a magnetometer 253 in the terminal device 250. The compass North is a direction from which all bearings are measured. Thus in this case, the bearing of a mobile tag 210 as indicated by arrow 860 is measured from the compass North, as shown by the angle α. The length of arrow 860 indicates the distance between the terminal device 250 and the mobile tag 210. Similarly, the bearing of a further mobile tag 210 as indicated by arrow 870 is measured from the compass North and is shown by the angle β. The length of arrow 870 indicates the distance between the terminal device 250 and the further mobile tag 210. Therefore, in this particular example, the mobile tag 210 represented by arrow 860 is closer to the terminal 250 than the mobile tag 210 represented by arrow 870, as seen by the respective lengths of arrows 860 and 870.

In various embodiments of the present disclosure, the floorplan of a building may be additionally displayed by the display in relation the previously described display techniques. This floorplan will be of an area of the building within which the BTLE devices are located, and would be within the area covered by the server. The points corresponding to the relative location of the mobile tag(s) are superimposed onto the display of the floorplan. This allows a user having a terminal device 250 to navigate his or her way around the building with ease when tracking a mobile tag 210.

Alternatively, the data relating to the position vectors stored in the RAM 276 of the server 270 could be output to a display 258 of a terminal device 250 which displays this data as text to a user. Such text may present the user with the numerical coordinates of the mobile tag(s) 210 in relation to the position of the terminal device 250.

The information contained in the header portion of each AD structure of the BTLE messages can be used to selectively display position data according to various criteria. These criteria may be based on an identifier value contained in the data packets. In relation to the example provided in Figure 1(a), the position data may be selected according to the device name, the manufacturer ID or the service, for example. In this manner, the display will only display the position of mobile tags that meet the selection criteria (i.e. mobile tags which have position data that contains the selected identifier value).

Figure 11(a) illustrates an exemplary display 258 of a terminal device 250. The display 258 comprises at least a screen 910. The screen 910 displays the position of the terminal device 250 as point 920, points corresponding to mobile tags 210 of a first type 930, 931, 932, and 933, and points corresponding to mobile tags of a second type 940 and 941. The first type of mobile tag may have a different identifier value to the second type of mobile tag. For example the first identifier type may be for mobile tags which are attached to patients in a hospital, while the second identifier type may be for mobile tags which are attached to nurses in the same hospital. In the example shown in Figure 11(a), mobile tags of the first type are shown as triangles, while mobile tags of the second type are shown as crosses. The position of the terminal device 250 is shown as a square.

In Figure 11(a), the screen 910 also displays a selector menu 950. The selector menu 950 enables the user to display points having identifiers that correspond to a selected value. This alters the display on screen 910 of the display 258. In Figure 11(b), 'sel_1' has been selected from the selector menu; thus the screen 910 only displays points that correspond to mobile tags 210 that have an identifier value of 'sel_1' in their BTLE messages. These points are the points identified by the triangle icon, and represent mobile tags 210 having the 'sel_1' identifier. To use the above example of a hospital, 'sel_1' could be a value associated with patients.

Conversely, if 'sel_2' has been selected by a user, the screen 910 only displays points that correspond to mobile tags 210 that have an identifier value of 'sel_2'. These points are the points identified by the cross icon, and represent mobile tags 210 having the 'sel_2' identifier in their BTLE messages. To use the above example of a hospital, 'sel_2' could be a value associated with nurses.

A user can therefore filter the displayed position of mobile tags 210 according to their identifiers. This is especially useful when the number of mobile tags 210 is large, which, in turn, clutters a display of the locations of these devices within an area. After filtering the points, the user is able to get a clear idea of the location of mobile tags 210 (having a selected identifier) in relation to the terminal device 250. This may be especially useful for tracking patients or expensive pieces of hospital equipment, for example. As with the examples provided above, a map showing the features of a building or complex within the area may be provided on the screen 810 to enhance the information provided to the user.

Returning back to Figure 7, mobile tags 210 present in two different areas 520 (zone A) and 530 (zone B) can be displayed together by a single terminal device 250.

Figure 12 illustrates an exemplary display 258 of terminal device 250. The display 258 comprises at least a screen 1005. The screen 1005 displays the position of the terminal device 250 as point 1010 and points corresponding to mobile tags 210 of a various types, shown collectively as 1020, in an area covered by the server 270 (area 520 (zone A) in Figure 7). The position of the terminal device 250 is shown as a square. A selector menu 950 is also shown for the purposes as discussed above in relation to Figure 11(a).

Additionally, the display 258 also shows a section 1030 containing point 1040 which relates to mobile device 210B in an area external to area 520. From Figure 7, mobile tag 210B is located in area 530 (zone B). The position of this externally located mobile tag is shown as text in Figure 10, but it will be understood that the positions of these tags may be displayed in accordance with any of the embodiments as discussed in relation to Figures 9 and 10. For example, a map of area 530 (zone B) may be shown in an inset in the display 258 to pictorially represent the position of mobile tag 210B in area 530.

Embodiments of the present disclosure may be applied to an augmented reality environment. Here the terminal device 250 may additionally be equipped with a camera which is able to capture images of the area covered by the terminal device 250. The terminal device 250 overlays these images with the position data obtained from the respective mobile tags 210 available in the area. This presents the user with a stream of images as the terminal device 250 is held by the user, the stream of images showing the position of mobile tags 210 in the area. It will be understood that the positions of these tags may be displayed in the augmented reality environment in accordance with any of the embodiments as discussed in relation to Figures 8 to 12.

Embodiments of the present disclosure may be implemented in software, hardware, application logic or a combination of software, hardware and application logic. The software, application logic and/or hardware may reside on memory, or any computer media. In an example embodiment, the application logic, software or an instruction set is maintained on any one of various conventional computer-readable media. In the context of this document, a “computer-readable medium” may be any media or means that can contain, store, communicate, propagate or transport the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer.

A computer-readable medium may comprise a computer-readable storage medium that may be any tangible media or means that can contain or store the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer as defined previously.

According to various embodiments of the previous aspect of the present disclosure, the computer program according to any of the above aspects, may be implemented in a computer program product comprising a tangible computer-readable medium bearing computer program code embodied therein which can be used with the processor for the implementation of the functions described above.

Reference to “computer-readable storage medium”, “computer program product”, “tangibly embodied computer program” etc., or a “processor” or “processing circuit” etc. should be understood to encompass not only computers having differing architectures such as single/multi processor architectures and sequencers/parallel architectures, but also specialised circuits such as field programmable gate arrays FPGA, application specify circuits ASIC, signal processing devices and other devices. References to computer program, instructions, code etc. should be understood to express software for a programmable processor firmware such as the programmable content of a hardware device as instructions for a processor or configured or configuration settings for a fixed function device, gate array, programmable logic device, etc.

By way of example, and not limitation, such “computer-readable storage medium” may mean a non-transitory computer-readable storage medium which may comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. An exemplary non-transitory computer-readable storage medium 1100 is shown in Figure 13, in the form of an optical storage disk such as a CD. Also, any connection is properly termed a "computer-readable medium". For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that "computer-readable storage medium" and data storage media do not include connections, carrier waves, signals, or other transient media, but are instead directed to non-transient, tangible storage media. Disk and disc, as used herein, include compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray disc, where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of "computer-readable medium".

Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules. Also, the techniques could be fully implemented in one or more circuits or logic elements.

If desired, the different steps discussed herein may be performed in a different order and/or concurrently with each other. Furthermore, if desired, one or more of the above-described steps may be optional or may be combined.

Although various aspects of the present disclosure are set out in the independent claims, other aspects of the present disclosure comprise other combinations of features from the described embodiments and/or the dependent claims with the features of the independent claims, and not solely the combinations explicitly set out in the claims.

Claims

1. A method comprising:

a fixed tag obtaining an instant value from a dynamic number generator within said fixed tag;

the fixed tag using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and

the fixed tag transmitting a first Bluetooth Low Energy message including said first secure identifier.

2. The method of claim 1, wherein the tag identifier comprises a first identifier and a second identifier.

3. The method according to claim 2, wherein the first function comprises:

the fixed tag generating an encrypted value using said instant value and said first identifier; and

the fixed tag concatenating said encrypted value with said second identifier.

4. The method according to any of claims 2 and 3 further comprising:

the fixed tag using the first function to map said first and second identifiers and a new instant value obtained from said number generator to a further first secure identifier; and

the fixed tag transmitting a further first Bluetooth Low Energy message including said further first secure identifier.

5. The method according to any of claims 3 and 4, wherein said encrypted value is generated using an advanced encryption standard cipher.

6. The method according to any of the preceding claims, wherein said dynamic number is generated using a network time protocol.

7. The method according to any preceding claim, wherein said instant value is a timestamp.

8. The method according to any of the preceding claims, wherein the radio parameter is received signal strength.

9. A method comprising:

a mobile tag receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags, each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag;

the mobile tag using a second function to map each first secure identifier to a respective second secure identifier;

the mobile tag measuring a radio parameter for each of the first Bluetooth Low Energy messages; and

the mobile tag transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages, said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

10. The method according to claim 9 wherein the second function comprises:

the mobile tag truncating each first secure identifier to generate said second secure identifier.

11. The method according to any of claims 9 and 10, wherein the second Bluetooth Low Energy message includes an identifier corresponding to the mobile tag.

12. The method according to any of claims 9 to 11, wherein the radio parameter is received signal strength.

13. The method according to any of claims 9 to 12, comprising the mobile tag receiving at least one first Bluetooth Low Energy message from each of at least three fixed tags.

14. A method comprising:

a Bluetooth receiver device receiving at least one first Bluetooth Low Energy message from each of at least two fixed tags;

the Bluetooth receiver device receiving at least one second Bluetooth Low Energy message from a mobile tag;

the Bluetooth receiver device decoding from each first Bluetooth Low Energy message a first secure identifier corresponding to a respective fixed tag;

the Bluetooth receiver device decoding from each second Bluetooth Low Energy message at least one second secure identifier corresponding to each fixed tag, said second secure identifier being derived from said first secure identifier;

the Bluetooth receiver device decoding from each second Bluetooth Low Energy message measured radio parameters associated with each of said first Bluetooth Low Energy messages;

sever apparatus using said first and second secure identifiers to identify each fixed tag;

the server apparatus retrieving position data corresponding to each fixed tag; and

the server apparatus using the position data and the radio parameters to calculate a position of the mobile tag.

15. The method according to claim 14 further comprising:

the Bluetooth receiver device decoding from the second Bluetooth Low Energy message an identifier corresponding to the mobile tag.

16. The method according to any of claims 14 and 15, comprising the server apparatus detecting whether the mobile tag is moving.

17. The method according to claim 16, comprising the server apparatus performing position calculation more frequently in response to detecting that the mobile tag transitions to a moving state.

18. The method according to any of claims 14 to 17, comprising the server apparatus using first Bluetooth Low Energy messages from a predetermined number of fixed tags from which strongest signals are received for calculating the position of the mobile tag if signals are received from the predetermined number of fixed tags.

19. The method according to claim 18, wherein the predetermined number is at least three.

20. The method according to any of claims 14 to 19, wherein the position data contains the latitude, longitude and floor number of the respective device.

21. The method according to any of claims 14 to 20, wherein said radio parameter is received signal strength.

22. The method according to any of claims 14 to 21, comprising the server apparatus calculating the position of more than one mobile tag.

23. The method according to any of claims 14 to 22, comprising the server apparatus using the first Bluetooth Low Energy messages from at least two fixed tags with the highest values of the radio parameter for calculating the position of the mobile tag.

24. The method according to any of claims 14 to 23, comprising the server apparatus storing the position data corresponding to each of the fixed tags, together with their corresponding first and second secure identifiers, in a memory device.

25. The method according to claim 24, wherein the memory device is a non-volatile memory, preferably a read only memory.

26. The method according to any of claims 14 to 25, wherein the Bluetooth receiver device comprises a transceiver device.

27. The method according to any of claims 14 to 26, wherein the Bluetooth receiver device has a fixed position.

28. The method according to any of claims 14 to 27, wherein the Bluetooth receiver device is capable of communication with the server apparatus via a data link.

29. The method according to claim 28, wherein the data link comprises any of: a WiFi connection and a wired connection.

30. The method according to any of claims 14 to 29, wherein the Bluetooth receiver device comprises a fixed tag selected from any of said at least two fixed tags.

31. Apparatus comprising a fixed tag, the fixed tag comprising at least one processor, at least one memory, and computer-readable code stored on the at least one

memory, wherein the computer-readable code when executed controls the at least one processor to perform a method comprising:

obtaining an instant value from a dynamic number generator within said fixed tag;

using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and

transmitting a first Bluetooth Low Energy message including said first secure identifier.

32. The apparatus according to claim 31, wherein the tag identifier comprises a first identifier and a second identifier.

33. The apparatus according to claim 32, wherein the first function comprises:

the fixed tag generating an encrypted value using said instant value and said first identifier; and

the fixed tag concatenating said encrypted value with said second identifier.

34. Apparatus comprising a mobile tag, the mobile tag comprising at least one processor, at least one memory, and computer-readable code stored on the at least one memory, wherein the computer-readable code when executed controls the at least one processor to perform a method comprising:

receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags, each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag;

using a second function to map each first secure identifier to a respective second secure identifier;

measuring a radio parameter for each of the first Bluetooth Low Energy messages; and

transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages, said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

35. The apparatus according to claim 34, wherein the second function comprises:

the mobile tag truncating each first secure identifier to generate said second secure identifier.

36. A computer program comprising computer executable instructions, which, when executed by a fixed tag, causes the fixed tag to perform a method according to any of claims 1 to 8.

37. A computer program comprising computer executable instructions, which, when executed by a mobile tag, causes the mobile tag to perform a method according to any of claims 9 to 13.

38. A non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by computing apparatus causes the computing apparatus to perform a method comprising:

a fixed tag obtaining an instant value from a dynamic number generator within said fixed tag;

the fixed tag using a first function to map said instant value and a tag identifier corresponding to said fixed tag to a first secure identifier; and

the fixed tag transmitting a first Bluetooth Low Energy message including said first secure identifier.

39. The non-transitory computer-readable storage medium according to claim 38, wherein the tag identifier comprises a first identifier and a second identifier.

40. The non-transitory computer-readable storage medium according to claim 39, wherein the first function comprises:

the fixed tag generating an encrypted value using said instant value and said first identifier; and

the fixed tag concatenating said encrypted value with said second identifier.

41. A non-transitory computer-readable storage medium having stored thereon computer-readable code, which, when executed by computing apparatus causes the computing apparatus to perform a method comprising:

a mobile tag receiving at least one first Bluetooth Low Energy message transmitted from each of at least two fixed tags, each first Bluetooth Low Energy message including at least a first secure identifier corresponding to the respective fixed tag;

the mobile tag using a second function to map each first secure identifier to a respective second secure identifier;

the mobile tag measuring a radio parameter for each of the first Bluetooth Low Energy messages; and

the mobile tag transmitting at least one second Bluetooth Low Energy message including the second secure identifier of each of the fixed tags and the measured radio parameters associated with each of the received first Bluetooth Low Energy messages, said second Bluetooth Low Energy message enabling the calculation of a position of the mobile tag.

42. The non-transitory computer-readable storage medium according to claim 41, wherein the second function comprises:

the mobile tag truncating each first secure identifier to generate said second secure identifier.

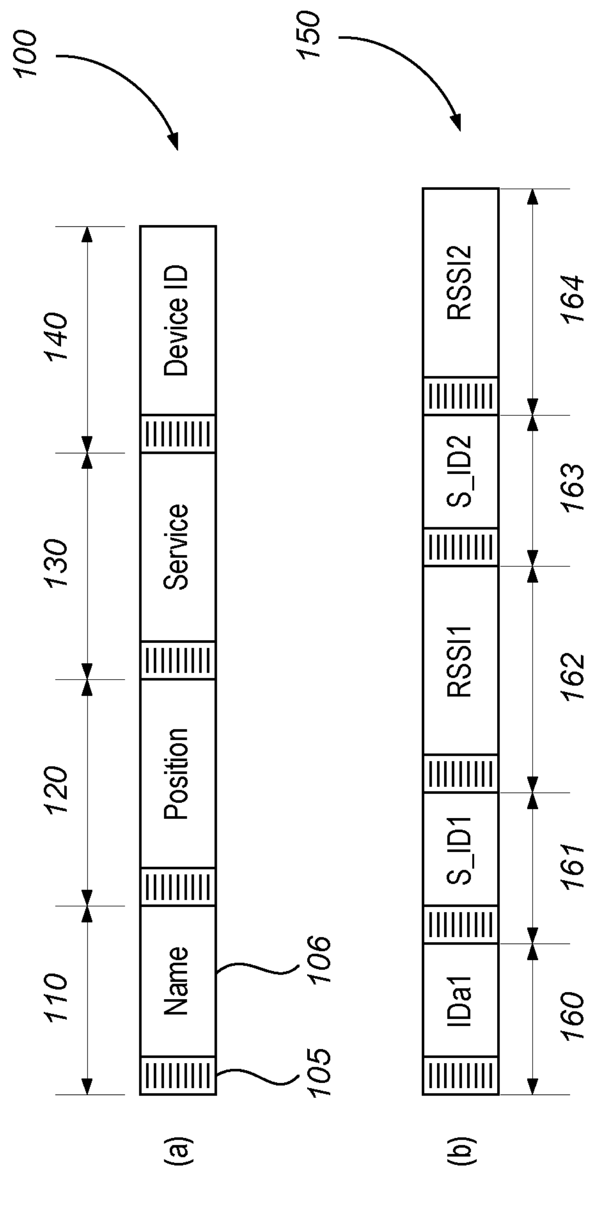
100
150
Device ID
Service
Position
Name
140
130
120
110
106
105
RSSI2
S_ID2
RSSI1
S_ID1
IDa1
164
163
162
161
160
(a)
(b)

FIG. 1

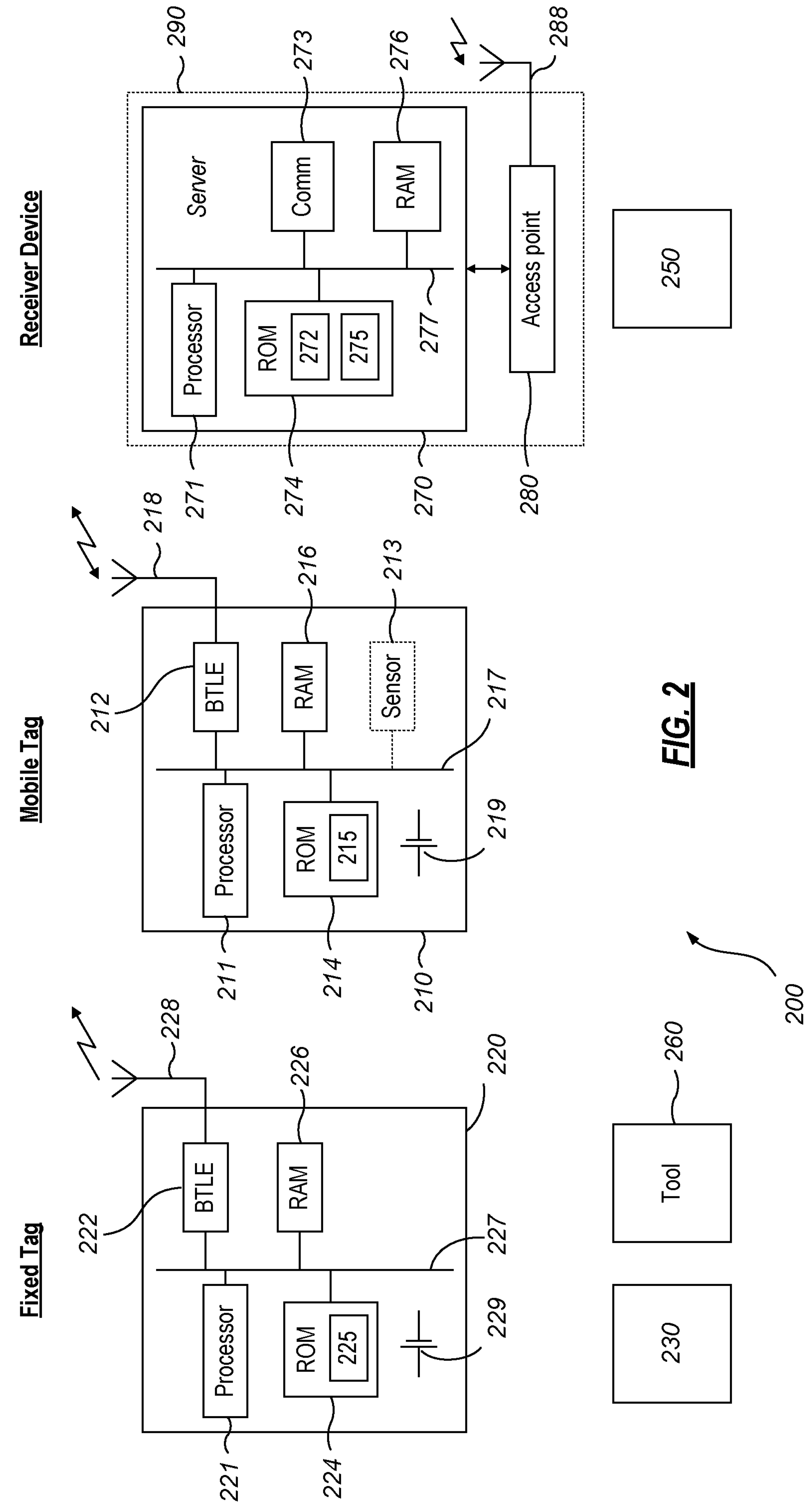
Receiver Device
Mobile Tag
Fixed Tag
Server
Comm
RAM
Processor
ROM
272
275
Access point
250
290
273
276
288
271
274
270
280
277
BTLE
RAM
Sensor
Processor
ROM
215
212
216
213
217
218
219
211
214
210
BTLE
RAM
Processor
ROM
225
222
226
228
220
227
229
221
224
Tool
260
230
200

FIG. 2

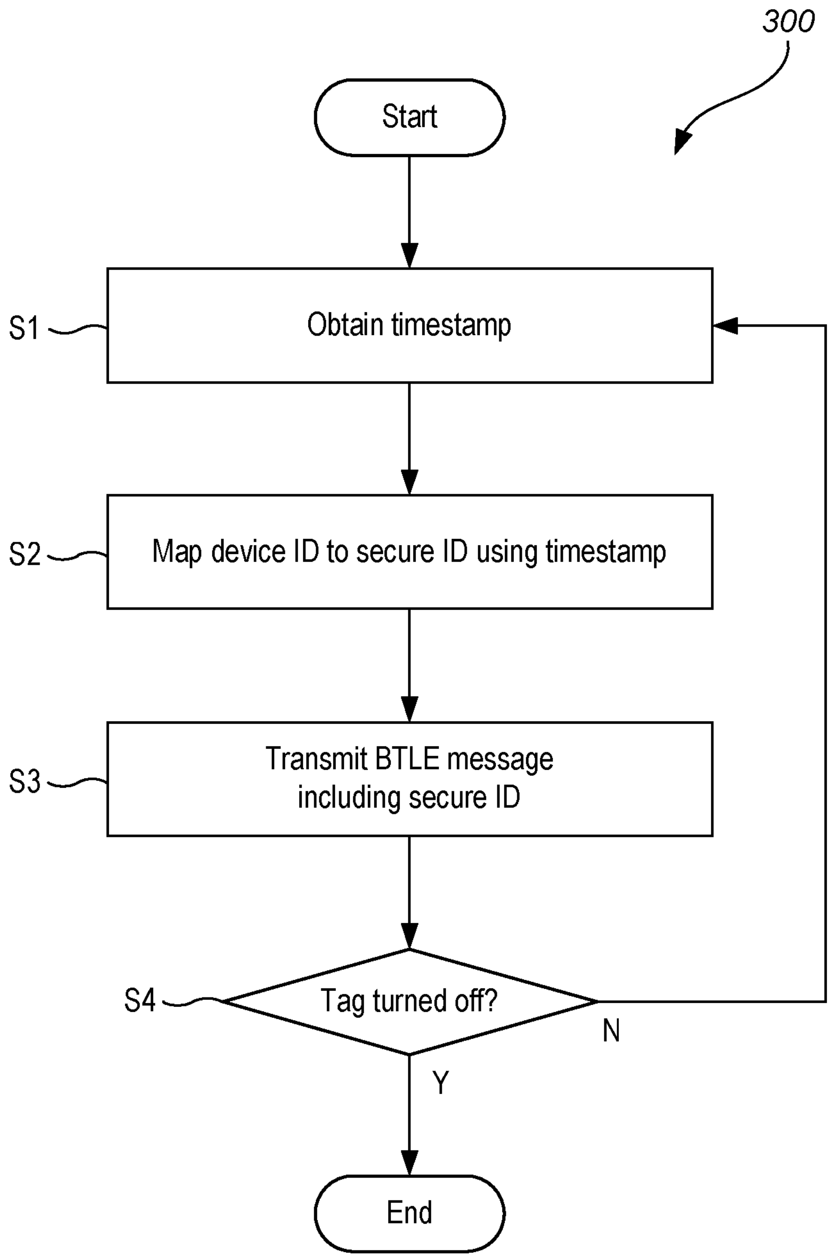
300
Start
S1
Obtain timestamp
S2
Map device ID to secure ID using timestamp
S3
Transmit BTLE message including secure ID
S4
Tag turned off?
N
Y
End

FIG. 3

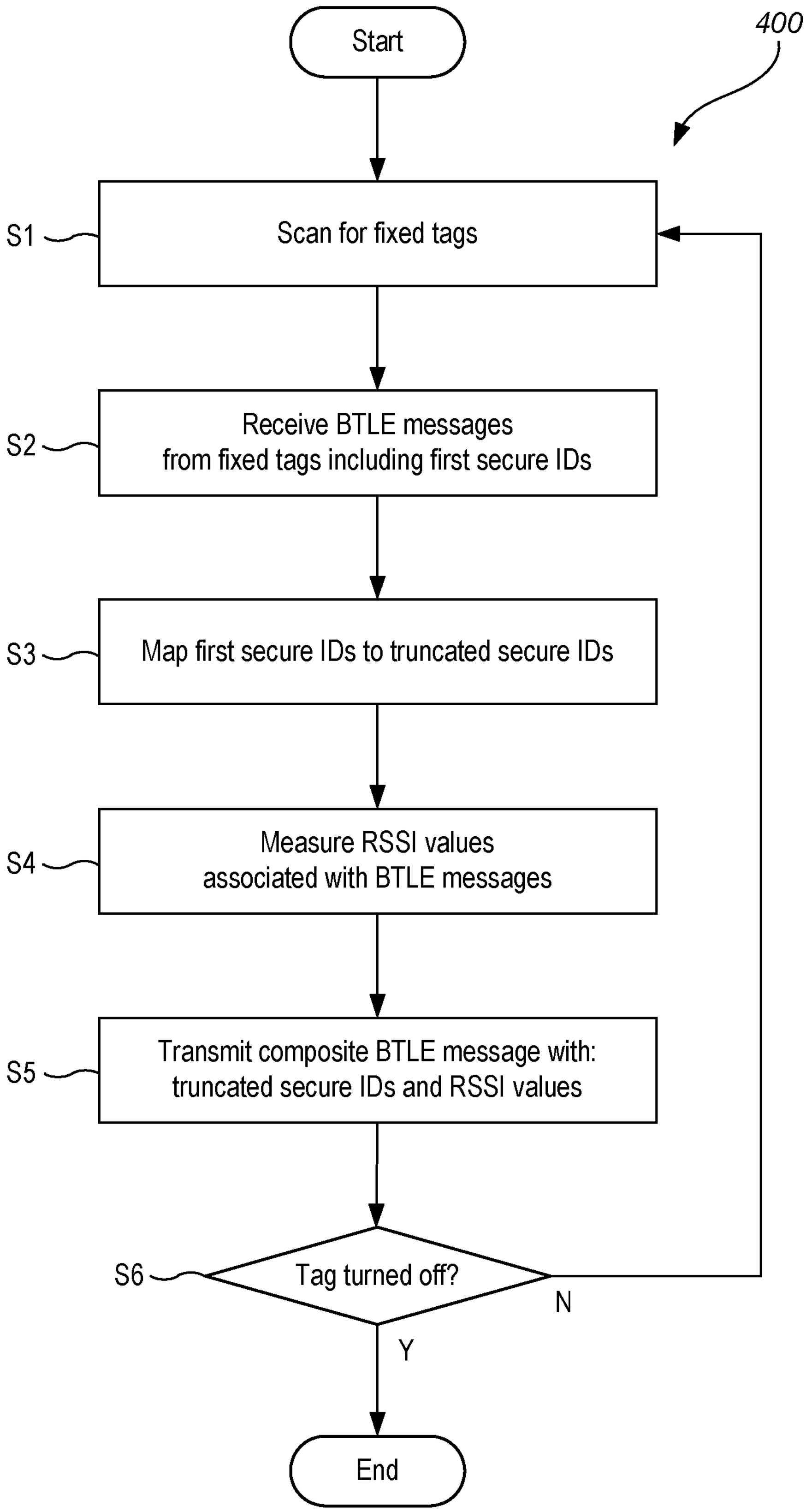
400
Start
S1
Scan for fixed tags
S2
Receive BTLE messages
from fixed tags including first secure IDs
S3
Map first secure IDs to truncated secure IDs
S4
Measure RSSI values
associated with BTLE messages
S5
Transmit composite BTLE message with:
truncated secure IDs and RSSI values
S6
Tag turned off?
N
Y
End

FIG. 4

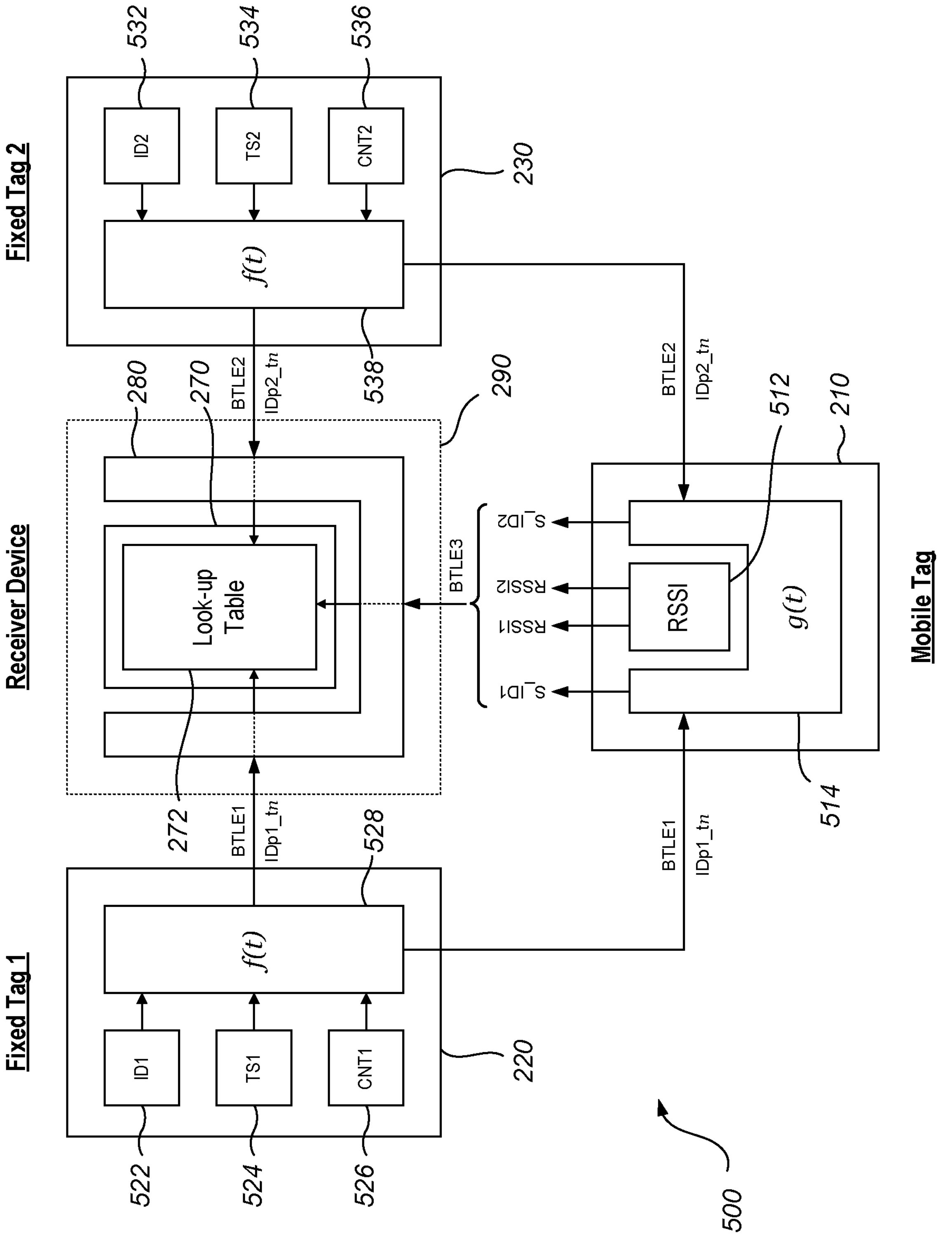
Fixed Tag 1
Receiver Device
Fixed Tag 2
ID1
TS1
CNT1
f(t)
522
524
526
528
220
ID2
TS2
CNT2
f(t)
532
534
536
538
230
Look-up Table
280
270
272
290
BTLE1
IDp1_tn
BTLE2
IDp2_tn
BTLE3
S_ID1
RSSI1
RSSI2
S_ID2
RSSI
g(t)
512
514
210
Mobile Tag
500

FIG. 5

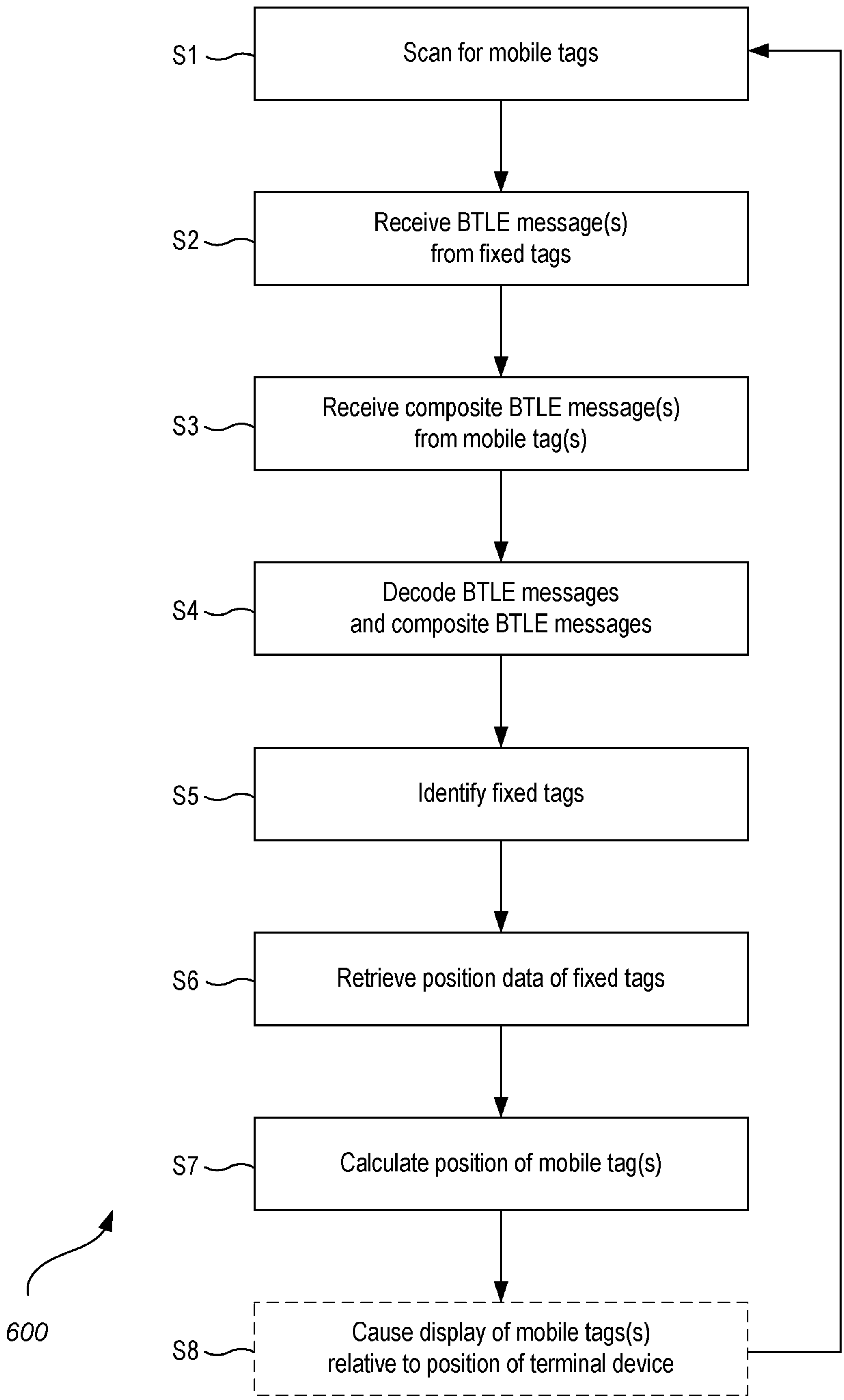
S1
Scan for mobile tags
S2
Receive BTLE message(s) from fixed tags
S3
Receive composite BTLE message(s) from mobile tag(s)
S4
Decode BTLE messages and composite BTLE messages
S5
Identify fixed tags
S6
Retrieve position data of fixed tags
S7
Calculate position of mobile tag(s)
S8
Cause display of mobile tags(s) relative to position of terminal device
600

FIG. 6

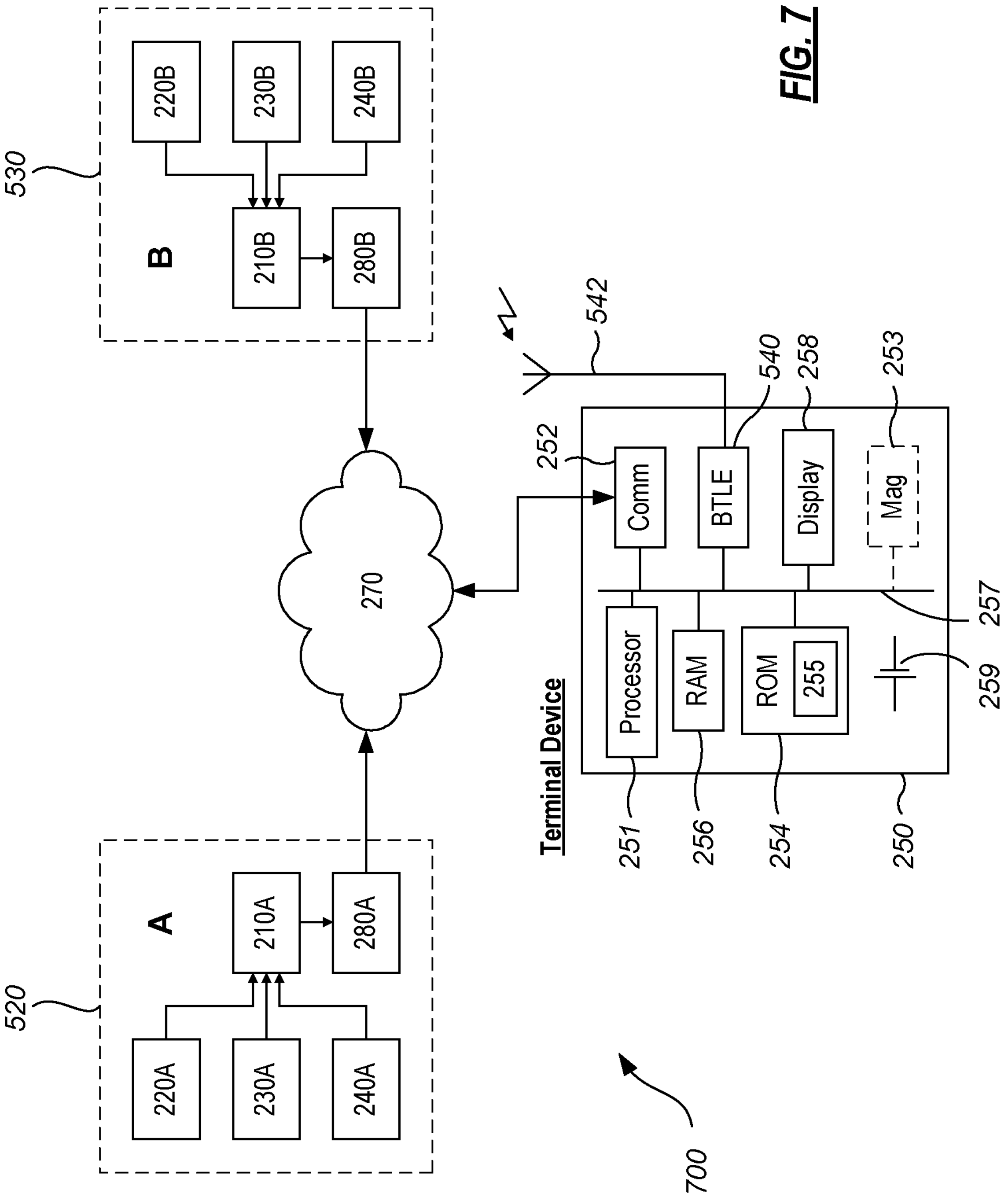
530
B
220B
230B
240B
210B
280B
520
A
220A
230A
240A
210A
280A
270
542
540
258
253
252
Comm
BTLE
Display
Mag
257
Processor
RAM
ROM
255
259
Terminal Device
251
256
254
250
700

FIG. 7

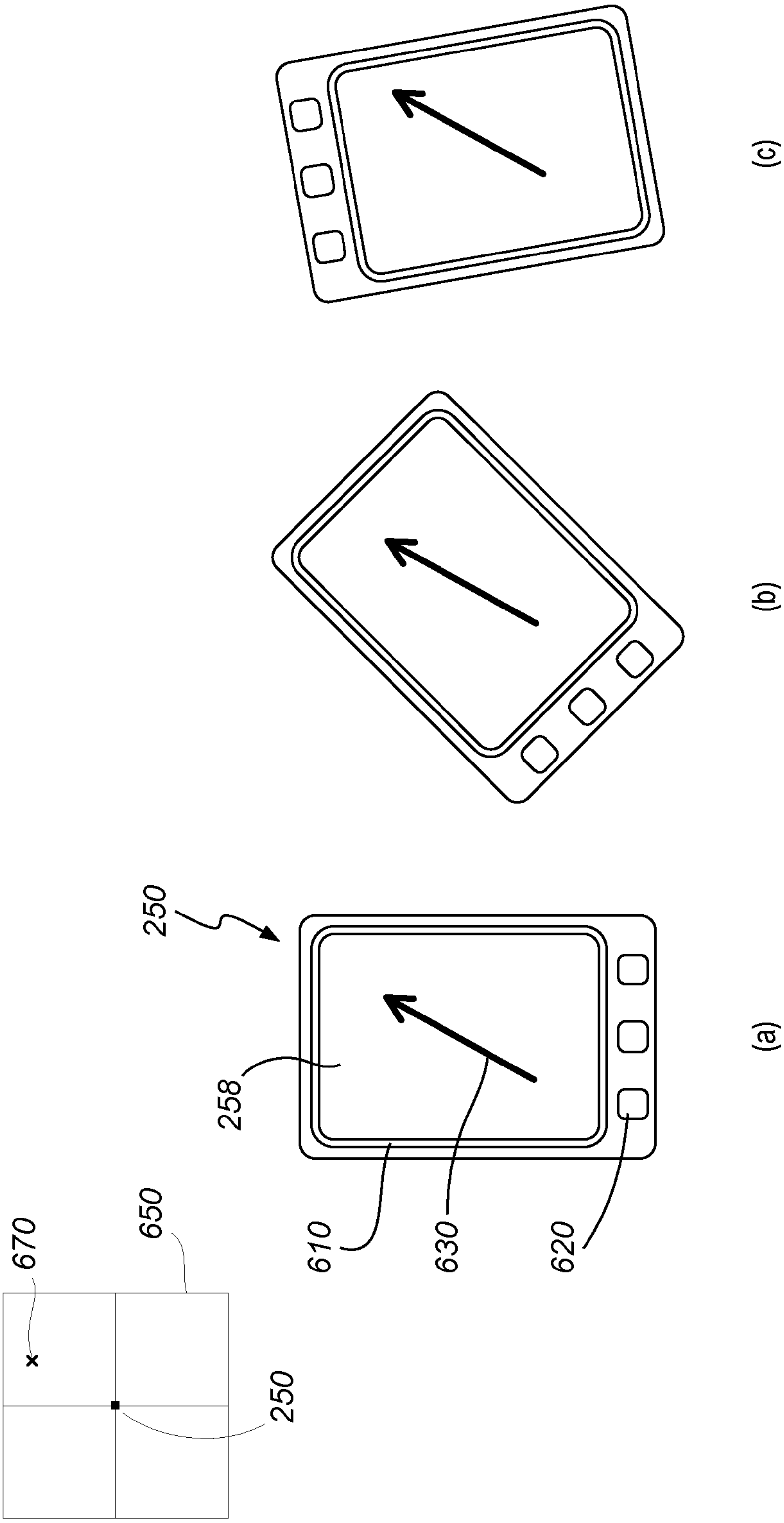

FIG. 8

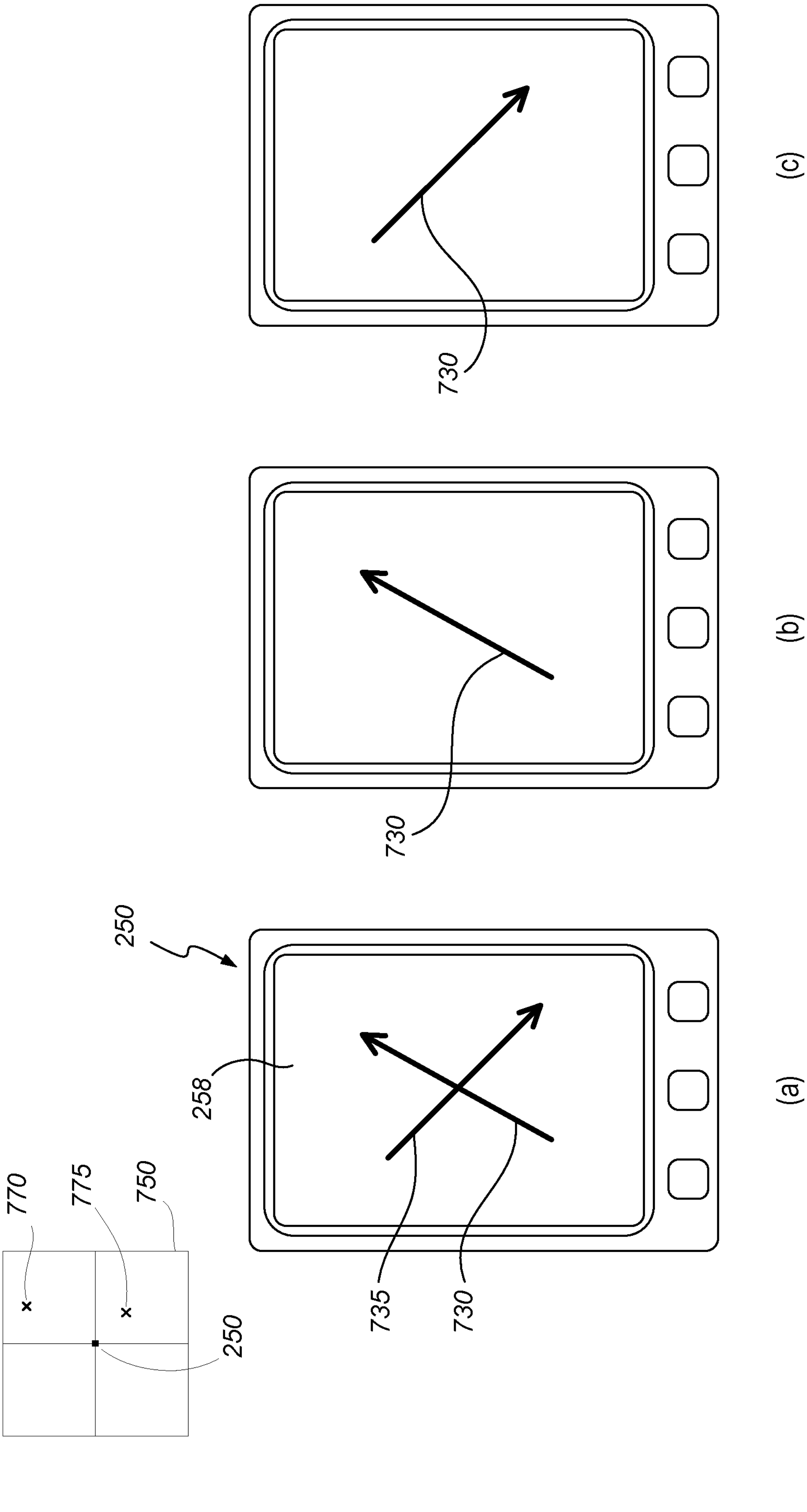

FIG. 9

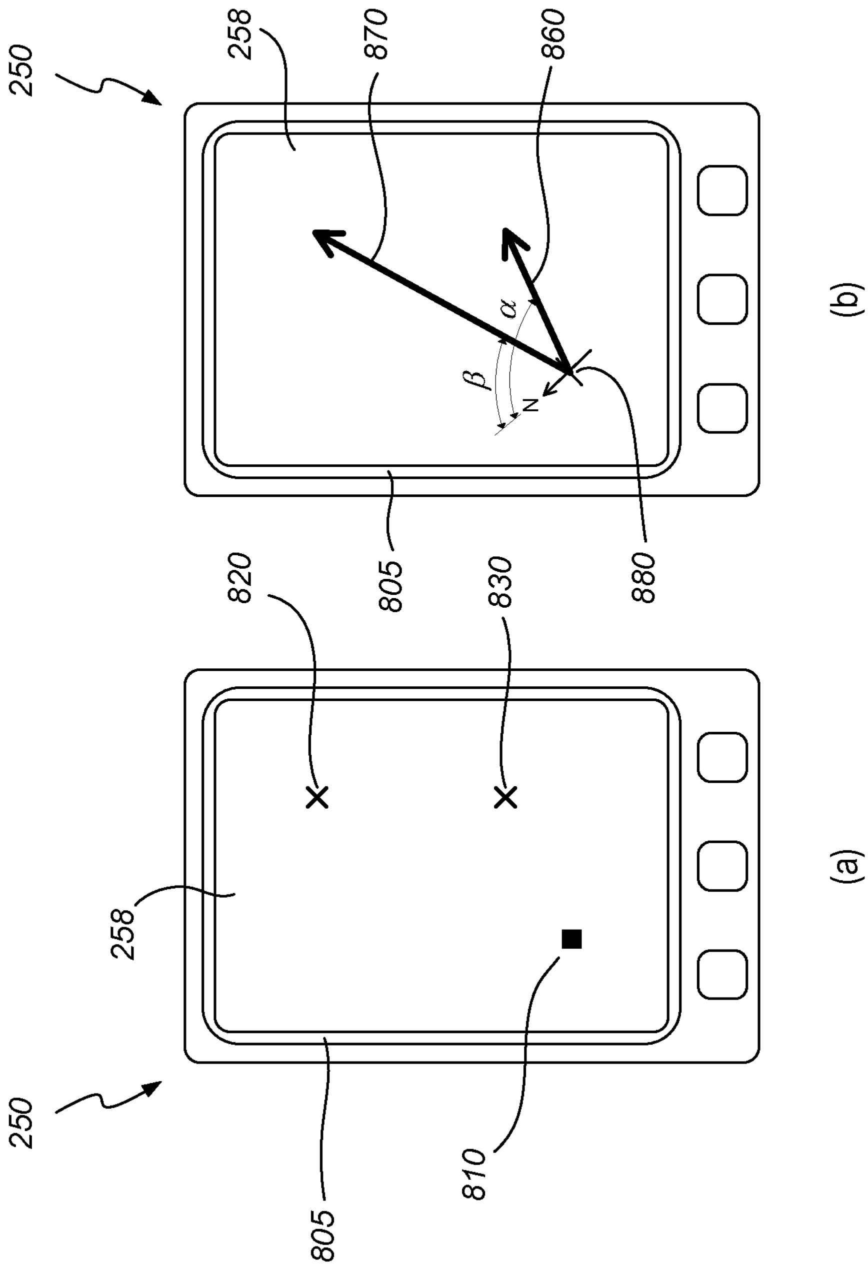

FIG. 10

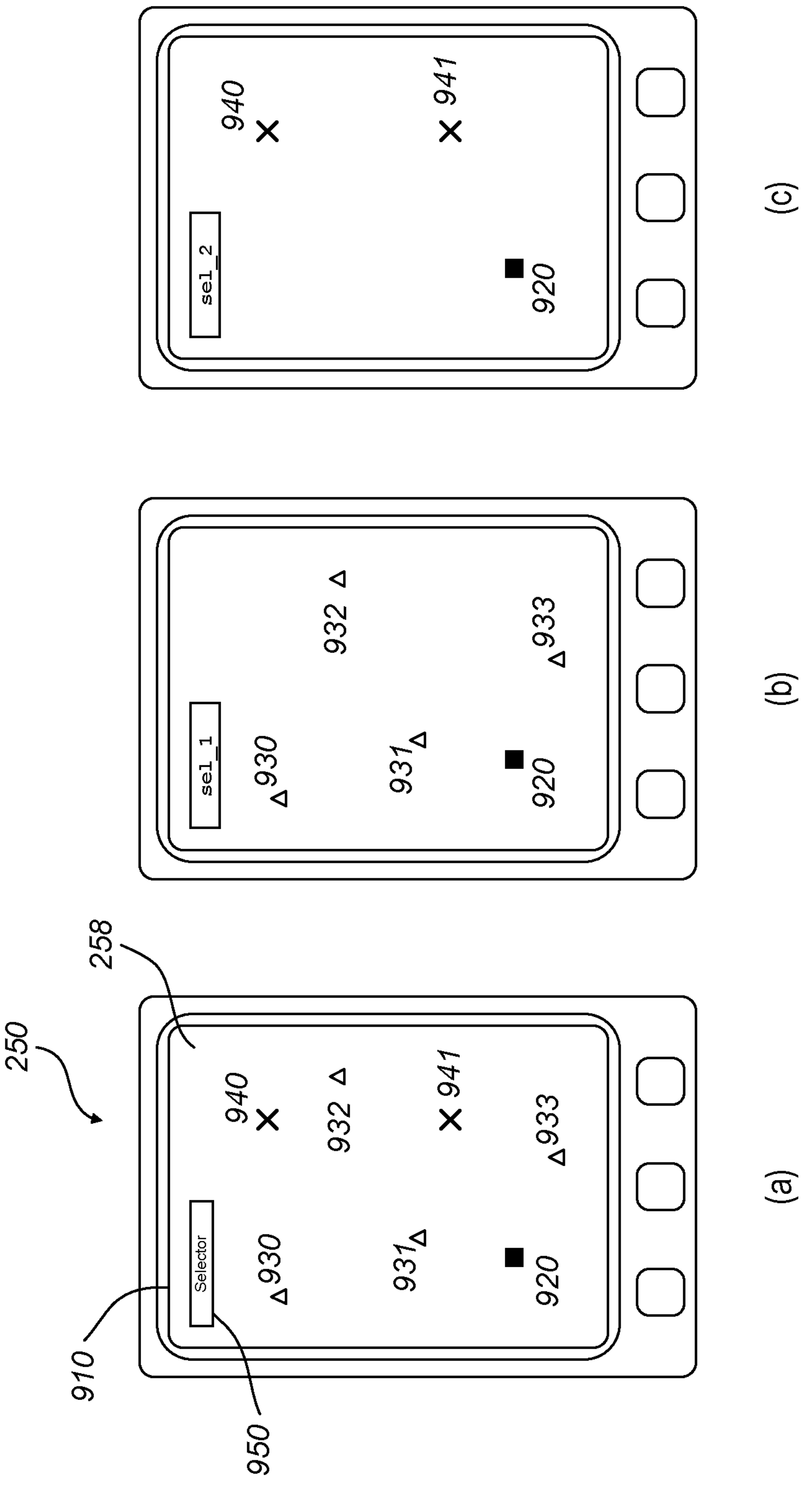
250
258
910
950
Selector
930
931
932
933
940
941
920
(a)
sel 1_
(b)
sel 2_
(c)

FIG. 11

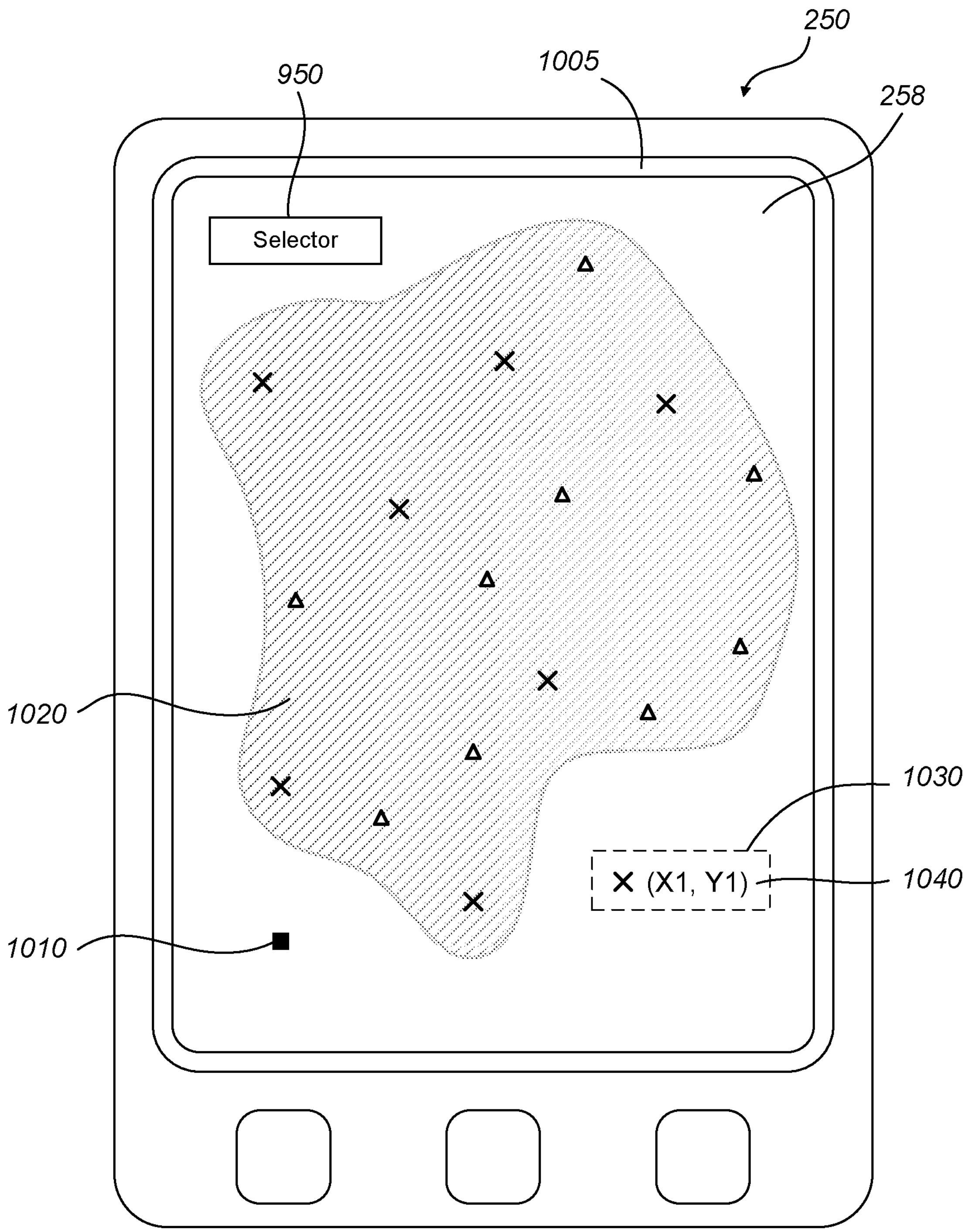

FIG. 12

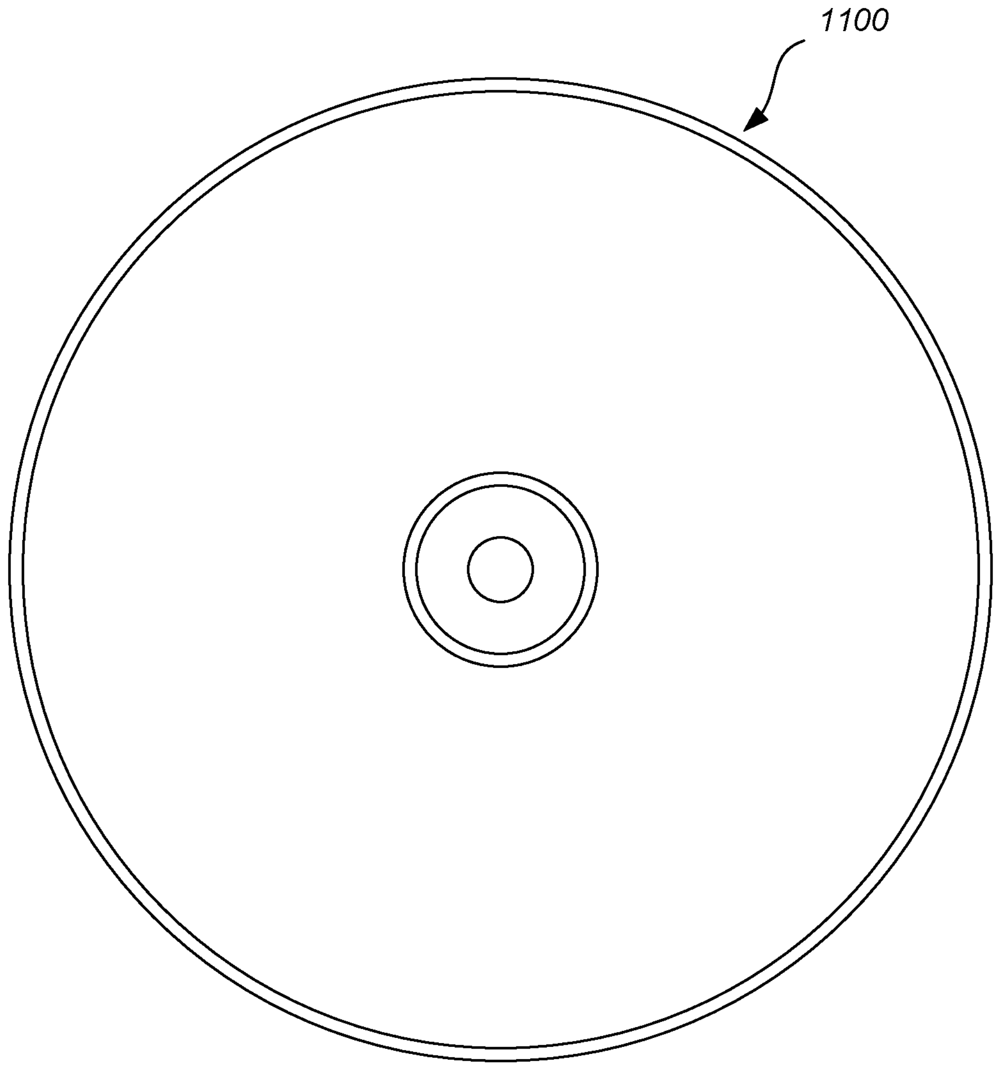

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2014/051038

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

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)

IPC: G01S, G01C, H04W, H04L

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

FI, SE, NO, DK

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

EPO-Internal & translations from Thomson Reuters and IPPH/SIPO, WPI, PRH-Internal, XP3GPP, XPAIP, XPESP, XPETSI, XPI3E, XPIEE, XPIETF, XPIOP, XPIPCOM, XPJPEG, XPMISC, XPOAC, XPRD, XPTK, COMPDX, INSPEC, TDB, NPL

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007247367 A1 (ANJUM F. [US] et al.) 25 October 2007 (25.10.2007) abstract; par. [0010], [0036]-[0037], [0044]-[0046]	1-42
X	ANJUM F. et al., "Secure localization in sensor networks using transmission range variation", Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems, 7-10 November, 2005, Washington D.C., United States, pp. 195-203. abstract; chapters I, III	1-42
A	WO 2013126759 A2 (QUALCOMM INC [US]) 29 August 2013 (29.08.2013) abstract	
A	WO 2007146364 A2 (TELCORDIA TECH INC [US]) 21 December 2007 (21.12.2007) abstract; page 21, par. 5 - page 22, par. 1	

☒ 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	Date of mailing of the international search report
03 August 2015 (03.08.2015)	17 August 2015 (17.08.2015)

Name and mailing address of the ISA/FI
Finnish Patent and Registration Office
P.O. Box 1160, FI-00101 HELSINKI, Finland
Facsimile No. +358 9 6939 5328

Authorized officer
Patrik Pousi

Telephone No. +358 9 6939 500

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2013084030 A1 (NOKIA CORP [FI]) 13 June 2013 (13.06.2013) abstract	
A	US 2009088183 A1 (PIERSOL K. W. [US] et al.) 02 April 2009 (02.04.2009) abstract	
A	US 2007219711 A1 (KALDEWEY T. [DE] et al.) 20 September 2007 (20.09.2007) abstract	

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2014/051038

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2007247367 A1	25/10/2007	US 7576694 B2	18/08/2009
		CA 2649527 A1	01/11/2007
		CA 2649527 C	16/06/2015
		EP 2018786 A1	28/01/2009
		JP 2008536340 A	04/09/2008
		WO 2007123260 A1	01/11/2007
WO 2013126759 A2	29/08/2013	CN 104246529 A	24/12/2014
		CN 104247369 A	24/12/2014
		CN 104365122 A	18/02/2015
		CN 104428806 A	18/03/2015
		CN 104584589 A	29/04/2015
		EP 2817651 A2	31/12/2014
		EP 2817937 A2	31/12/2014
		EP 2842295 A1	04/03/2015
		EP 2845030 A2	11/03/2015
		JP 2015510743 A	09/04/2015
		JP 2015513838 A	14/05/2015
		JP 2015515080 A	21/05/2015
		KR 20140144684 A	19/12/2014
		KR 20140146080 A	24/12/2014
		US 2013214909 A1	22/08/2013
		US 2013217332 A1	22/08/2013
		US 2013217333 A1	22/08/2013
		US 2013282438 A1	24/10/2013
		US 2013297422 A1	07/11/2013
		US 2014133656 A1	15/05/2014
		US 2014254466 A1	11/09/2014
		US 2014370879 A1	18/12/2014
		US 2015077229 A1	19/03/2015
		WO 2013126747 A2	29/08/2013
		WO 2013163326 A1	31/10/2013
		WO 2013163333 A2	31/10/2013
		WO 2013163334 A2	31/10/2013
		WO 2013163338 A2	31/10/2013
		WO 2015042065 A1	26/03/2015
		WO 2015042618 A1	26/03/2015
WO 2007146364 A2	21/12/2007	CA 2655415 A1	21/12/2007
		CA 2655415 C	05/11/2013
		EP 2039177 A2	25/03/2009
		US 2007291682 A1	20/12/2007

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
		US 7508788 B2	24/03/2009
WO 2013084030 A1	13/06/2013	CN 104221413 A	17/12/2014
		EP 2789181 A1	15/10/2014
		US 2015189619 A1	02/07/2015
US 2009088183 A1	02/04/2009	US 8711034 B2	29/04/2014
		EP 2046084 A1	08/04/2009
		JP 2009089395 A	23/04/2009
		JP 5223575 B2	26/06/2013
		JP 2009089396 A	23/04/2009
		JP 5298742 B2	25/09/2013
		US 2009085806 A1	02/04/2009
		US 8089405 B2	03/01/2012
		US 2009088182 A1	02/04/2009
		US 8265652 B2	11/09/2012
		US 2012162013 A1	28/06/2012
US 2007219711 A1	20/09/2007	US 7587274 B2	08/09/2009
		US 2009292460 A1	26/11/2009
		US 8346472 B2	01/01/2013
		US 2009292465 A1	26/11/2009
		US 8352176 B2	08/01/2013

CLASSIFICATION OF SUBJECT MATTER

IPC
G01S 1/02 (2010.01)
G01C 21/20 (2006.01)
H04L 9/08 (2006.01)
G01S 5/02 (2010.01)
H04W 12/00 (2009.01)
H04L 29/06 (2006.01)