

(51) International Patent Classification:
H04L 9/32 (2006.01)

(21) International Application Number:

PCT/EP2017/084603

(22) International Filing Date:

27 December 2017 (27.12.2017)

(25) Filing Language:

English

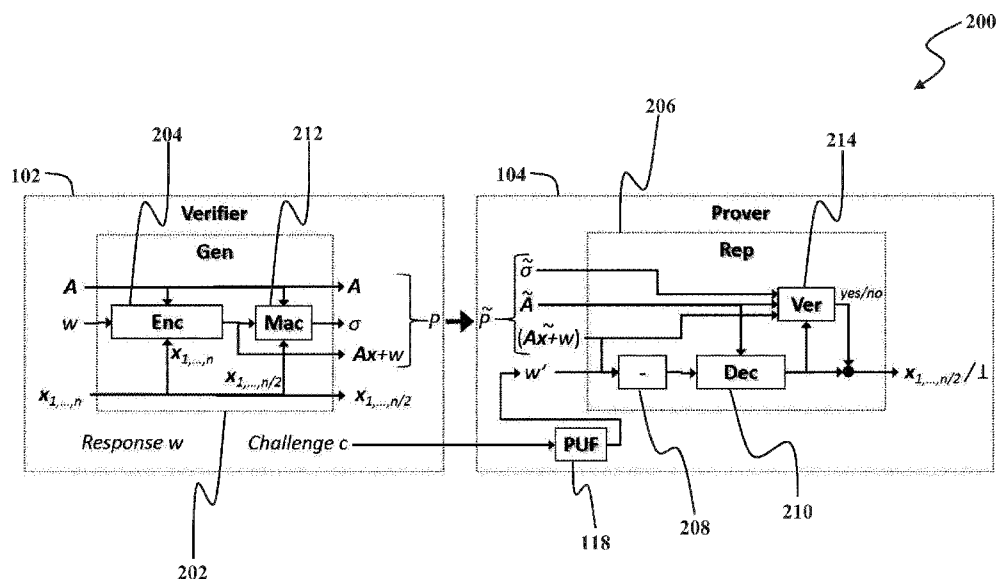
(26) Publication Language:

English

(30) Priority Data:

62/440728

30 December 2016 (30.12.2016) US

(71) Applicant: **ROBERT BOSCH GMBH** [DE/DE]; Postfach
30 02 20, 70442 Stuttgart (DE).(72) Inventors: **GUAJARDO MERCHAN, Jorge**; 813 North-
ridge Drive, Pittsburgh, Pennsylvania 15216 (US). **DU-**PLYS, Paulius; Bruckmuehle 5, 71706 Markgroeningen
(DE). **HUTH, Christopher**; Albertus-Magnus Strasse 13,
71229 Leonberg (DE).(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, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,
KR, KW, 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,(54) Title: **ROBUST COMPUTATIONAL FUZZY EXTRACTOR AND METHOD FOR AUTHENTICATION****FIG. 2**

(57) **Abstract:** A method and system for authenticating a device is disclosed. The method includes the steps of: receiving a helper bit string and a first MAC; measuring a first response bit string of a physical unclonable function of the device with respect to a challenge bit string; subtracting the first response bit string from the helper bit string; decoding a result of the subtraction using a uniformly distributed random matrix, the shared secret bit string being provided from the decoding if the helper bit string was encoded using a previously measured second response bit string that is within a threshold level of similarity to the first response bit string, the decoding outputting an error value otherwise; determining a second MAC based on the shared secret bit string, the uniformly distributed random matrix, and the helper bit string; and determining whether the second MAC matches the first MAC.



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).

Published:

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

ROBUST COMPUTATIONAL FUZZY EXTRACTOR AND METHOD FOR AUTHENTICATION

[0001] This application claims the benefit of priority of U.S. provisional application serial no. 62/440,728, filed on December 30, 2016, the disclosure of which is herein incorporated by reference in its entirety.

FIELD

[0002] The method and devices disclosed in this document relates to cryptography and, more particularly, to use of physical unclonable functions and fuzzy extractors for device authentication.

BACKGROUND

[0003] In some applications, physical unclonable functions (PUFs) are used to authenticate devices. However, the raw response of any PUF is noisy and has to be post-processed to derive e.g. an identical key every time the PUF is queried. Fuzzy extractors can be constructed with error correcting codes and used to remove the noise from the PUF's raw response using so-called helper data, which is publicly available. Due to the nature of the error correcting codes (and the redundancy information present in the helper data), traditional information-theoretic fuzzy extractors do not preserve the entire entropy present in the PUF's raw response. For example, if the raw response of a PUF has n bits of entropy, only m bits can be used after the raw response of the PUF is processed by a traditional information-theoretic fuzzy extractor, where $m < n$. Such a loss of entropy is particularly bad in situations where the PUF's raw response has a limited amount of

entropy. As an example, preliminary experiments indicate that the entropy of a MEMS-PUF's raw response amount to little less than 90 bits. However, this is not sufficient in terms of security if the PUF response is used to derive a symmetric key. For instance, the German BSI technical guideline TR-02102-1 "Kryptographische Verfahren: Empfehlungen und Schlüssellängen" (version 2015-01, 10th February 2015) requires a symmetric secret key length of at least 128 bits.

[0004] Accordingly, it would be advantageous to provide an alternative to the traditional fuzzy extractor in which the full entropy of a PUF can be preserved and from which longer keys can be derived. Additionally, it would further be advantageous if the method is secure against both passive eavesdroppers and active attackers.

SUMMARY

[0005] A method for authenticating a first device is disclosed. The method includes the steps of: receiving, with a transceiver of the first device, a helper bit string and a first message authentication code tag from a second device that is remote from the first device; measuring a first response bit string of a physical unclonable function of the first device with respect to a challenge bit string, the physical unclonable function being provided by one of the processor of the first device and a further physical component of the first device; subtracting, with the processor of the first device, the first response bit string from the helper bit string; decoding, with the processor of the first device, a result of the subtraction using a uniformly distributed random matrix, the shared secret bit string being provided from the decoding if the helper bit string was encoded using a previously measured second response bit string that is within a threshold level of

similarity to the first response bit string, the decoding outputting an error value otherwise; determining, with the processor of the first device, a second message authentication code tag based on the shared secret bit string, the uniformly distributed random matrix, and the helper bit string; and determining, with the processor of the first device, whether the second message authentication code tag matches the first message authentication code tag.

[0006] A further method for authenticating a first device is disclosed. The further method includes the steps of: measuring a first response bit string of a physical unclonable function of the first device with respect to a challenge bit string, the physical unclonable function being provided by a component of the first device; deriving, with a processor of the first device, a shared secret bit string from a uniformly distributed random vector; encoding, with the processor of the first device, a helper bit string by multiplying a uniformly distributed random matrix with the uniformly distributed random vector and adding the first response bit string to a result of the multiplication; determining, with the processor of the first device, a first message authentication code tag based on the shared secret bit string, the uniformly distributed random matrix, and the helper bit string; and transmitting, with a transceiver of the first device, the helper bit string and the first message authentication code tag to a second device that is remote from the first device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The foregoing aspects and other features of the methods and devices are explained in the following description, taken in connection with the accompanying drawings.

[0008] FIG. 1 shows a system comprising a proving device and a verifying device.

[0009] FIG. 2 shows a block diagram of a robust computational fuzzy extractor.

[0010] FIG. 3 shows a method for authenticating a device using the robust computational fuzzy extractor.

[0011] FIG. 4 shows a block diagram of a reverse robust computational fuzzy extractor.

[0012] FIG. 5 shows a method for authenticating a device using the reverse robust computational fuzzy extractor.

DETAILED DESCRIPTION

[0013] For the purposes of promoting an understanding of the principles of the disclosure, reference will now be made to the embodiments illustrated in the drawings and described in the following written specification. It is understood that no limitation to the scope of the disclosure is thereby intended. It is further understood that the present disclosure includes any alterations and modifications to the illustrated embodiments and includes further applications of the principles of the disclosure as would normally occur to one skilled in the art which this disclosure pertains.

[0014] FIG. 1 shows a system 100 comprising a proving device 102 (also referred to herein as the “prover”) and a verifying device 104 (also referred to herein as the

“verifier”). The proving device 102 and the verifying device 104 communicate with one another via an insecure communication channel, such as a network 106 (e.g., the Internet, a wireless local area network, or a wireless mesh network) or a direct communication channel (e.g., radio frequency identification (RFID) or near-field-communication (NFC)). Given the insecurity of the communication channel, the verifying device 104 and the proving device 102 are configured to perform an authentication process at least to verify the identity and authenticity of the proving device 102. In some embodiments, the authentication process is a mutual authentication process in which the identities and authenticity of both devices 102 and 104 are verified.

[0015] In some embodiments, the proving device 102 may comprise a security token, a smart card, a hardware security module, a machine, a controller, an actuator, a sensor, a tablet computer, a smartphone, a laptop, or any other device configured for communication with a host system or another device. In at least some embodiments, the proving device is a lightweight device with relatively limited processing capability and memory, such as an Internet of Things (IoT) device.

[0016] In the embodiment shown, the proving device 102 comprises a processor 108, memory 110, and a transceiver 112. The proving device 102 may also comprise many additional components which are operably connected to the processor 108 and configured to provide one or more services or functions, such as sensing elements, actuators, interfaces, displays, etc. (not shown). The memory 110 is configured to store program instructions that, when executed by the processor 108, enable the proving device 102 to perform one or more services or functions. In addition to program instructions for implementing the primary services or functions of the proving device 102, the program

instructions at least include an authentication program 114 for proving the identity and authenticity of the proving device 102 to the verifying device 104. The memory 110 is also configured to store data 116, which may include data utilized by the authentication program 126.

[0017] The memory 110 may be of any type of device capable of storing information accessible by the processor 108, such as a memory card, ROM, RAM, write-capable memories, read-only memories, hard drives, discs, flash memory, or any of various other computer-readable medium serving as data storage devices as will be recognized by those of ordinary skill in the art. Additionally, although the memory 110 is shown monolithically in F, the memory 110 may comprise several discrete memories of different types which are used for different purposes.

[0018] The processor 108 may include a system with a central processor, multiple processors, dedicated circuitry for achieving functionality, or other systems. Furthermore, it will be recognized by those of ordinary skill in the art that a "processor" includes any hardware system, hardware mechanism or hardware component that processes data, signals or other information.

[0019] The transceiver 112 may be any of various devices configured for communication with other electronic devices, including the ability to send communication signals and receive communication signals. The transceiver 112 may include different types of transceivers configured to communicate with different networks and systems. The transceiver 112 is at least configured to exchange data between the proving device 102 and the verifying device 104, but may also be configured to communicate with additional devices on the network 106. In one embodiment, the

transceiver 112 is configured to exchange data using a protocol such as Wi-Fi, Bluetooth, RFID, NFC, ZigBee, Z-Wave, or Ethernet.

[0020] The proving device 102 also has a physical unclonable function (PUF) 118, which can be utilized by the processor 108, configured to receive an input (e.g., a challenge bit string) and produce a unique output (e.g., a response bit string). The output response of the PUF 118 depends on the unique physical structure of at least one component of the proving device 102 and serves as a digital fingerprint for the proving device 102. In at least one embodiment, a semiconductor device, such as the processor 108, the memory 110, the transceiver 112, or a MEMS-sensor (not shown) of the proving device 102 provides the basis for the PUF 118. In other embodiments, the proving device 102 may include a dedicated semiconductor device (not shown) configured only to provide the PUF 118.

[0021] The microstructure of the semiconductor device, or other component, which provides the PUF 118 includes random physical variations that are naturally introduced by during manufacture and that are not feasibly controlled or replicated, even by the manufacturer. Additionally, in some types of PUF-enabled semiconductor devices, variations in environmental conditions, such as temperature, supply voltage, and electromagnetic interference also introduce randomness and unpredictability into the operation of the device. As a result, the PUF 118 has a unique and unpredictable way of generating a response to a particular input. Furthermore, for a given challenge input string, the PUF 118 does not necessarily reliably produce exactly the same response string each time. Instead, for a given challenge input string, the PUF 118 may generate

reliably similar but not identical responses each time it is used. In this way, the PUF 118 can be considered to have a noisy response.

[0022] In contrast to the proving device 102, in some embodiments, the verifying device 104 is a host system such as a remote server, a local control hub (e.g., as used in home automation systems), a payment kiosk, or any other device which must verify the identity and authenticity of connected devices. Additionally, the verifying device 104 generally has more processing capability and more memory than the proving device 102 and is better suited to bear any computationally or memory intensive aspects of the authentication process.

[0023] In the embodiment shown, the verifying device 104 comprises a processor 120, memory 122, and a transceiver 124. The memory 122 is configured to store program instructions that, when executed by the processor 120, enable the verifying device 104 to perform one or more services or functions. In addition to program instructions for implementing the primary services or functions of the verifying device 104, the program instructions at least include an authentication program 126 for verifying the identity and authenticity of the proving device 102. The memory 122 is also configured to store data 128, which may include data utilized by the authentication program 126, such as pairs of challenges and measured responses of the PUF 118 of the proving device 102.

[0024] The memory 122 may be of any type of device capable of storing information accessible by the processor 120, such as a memory card, ROM, RAM, write-capable memories, read-only memories, hard drives, discs, flash memory, or any of various other computer-readable medium serving as data storage devices as will be recognized by those of ordinary skill in the art. Additionally, although the memory 122 is shown

monolithically in the figure, the memory 122 may comprise several discrete memories of different types which are used for different purposes.

[0025] The processor 120 may include a system with a central processor, multiple processors, dedicated circuitry for achieving functionality, or other systems. Furthermore, it will be recognized by those of ordinary skill in the art that a "processor" includes any hardware system, hardware mechanism or hardware component that processes data, signals or other information.

[0026] The transceiver 124 may be any of various devices configured for communication with other electronic devices, including the ability to send communication signals and receive communication signals. The transceiver 124 may include different types of transceivers configured to communicate with different networks and systems. The transceiver 124 is at least configured to exchange data between the verifying device 104 and the proving device 102, but may also be configured to communicate with additional devices on the network 106. In one embodiment, the transceiver 124 is configured to exchange data using a protocol such as Wi-Fi, Bluetooth, RFID, NFC, ZigBee, Z-Wave, or Ethernet.

[0027] FIG. 2 shows a block diagram of a robust computational fuzzy extractor 200. Unlike traditional fuzzy extractors, which are defined as information-theoretic objects, the robust computational fuzzy extractor 200 leverages computational security based on Learning with Errors (LWE), which is post-quantum secure, to preserve the full entropy of the PUF 118 and obtain longer cryptographic keys. Unlike many traditional fuzzy extractors, the cryptographic key is not extracted from the PUF response. Instead, a random linear code is used to derive a longer key using LWE assumptions and the PUF

response is used to “encrypt” the key in a manner that is decryptable with knowledge of a sufficiently similar PUF response.

[0028] The robust computational fuzzy extractor 200 includes a generate function 202. The generate function 202 receives as an input a previously measured response w of the PUF 118. Additionally, the generate function 202 receives as inputs a uniformly distributed random matrix A and a uniformly distributed random vector $x_{1,\dots,n}$. In at least one embodiment, the vector $x_{1,\dots,n}$ is a random linear code. In one exemplary embodiment, the matrix A and the vector $x_{1,\dots,n}$ are uniformly sampled over a finite field $\mathbb{F}_q$ according to $A \in \mathbb{F}_q^{m \times n}$ and $x \in \mathbb{F}_q^n$, where n is a security parameter, $m \geq n$, and q is prime, as described in *Computational Fuzzy Extractors* (Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. In *Advances in Cryptology-ASIACRYPT 2013*, pages 174-193. Springer, 2013), in which m , n , and q are selected so as to maintain the full entropy of the source which provides the input string w . The generate function 202 includes an encode function 204 that computes the vector $Ax + w$ using matrix multiplication and addition, which can be considered an encryption of $x_{1,\dots,n}$ where decryption works from any close w . Furthermore, the generate function 202 derives a shared secret $x_{1,\dots,n/2}$ from the random vector $x_{1,\dots,n}$. In one embodiment, the shared secret $x_{1,\dots,n/2}$ is a bit string comprising the first through $\frac{n}{2}$ elements of the random vector $x_{1,\dots,n}$. In one embodiment, the shared secret $x_{1,\dots,n/2}$ is the same as the random vector $x_{1,\dots,n}$. Finally, the generate function 202 includes a message authentication code (MAC) function 212 that computes a MAC tag σ , which can be considered a keyed cryptographic hash, based on the matrix A , the vector $Ax + w$, and the shared secret $x_{1,\dots,n/2}$. The outputs of the generate function

202 are the shared secret $x_{1,...,n/2}$ and helper data $P = (A, \sigma, Ax + w)$. The helper data P is considered public data and may be known to an adversary, whereas the shared secret $x_{1,...,n/2}$ may be used as a cryptographic key or for authentication.

[0029] The robust computational fuzzy extractor 200 further includes a reproduce function 206. The reproduce function 206 receives as an input a measured response w' of the PUF 118. Additionally, the reproduce function 206 receives as inputs helper data $\tilde{P} = (\tilde{A}, \tilde{\sigma}, \widetilde{Ax + w})$, which may be the same as the helper data P , but may also include one or more elements that have been modified by an active attacker. The reproduce function 206 includes a subtract function 208 that computes $\tilde{b} = \widetilde{Ax + w} - w' = \widetilde{Ax} + \delta$. Additionally, the reproduce function 206 includes a decode function 210 that decodes the result of the subtraction $\widetilde{Ax} + \delta$ and is able to output the shared secret $x_{1,...,n/2}$ if the response w' is sufficiently close and/or similar to the response w according to some predefined metric (e.g., Hamming distance between w' and w is less than t). Otherwise, if the response w' is not sufficiently close and/or similar to the response w (e.g., Hamming distance between w' and w is greater than t), the decode function 210 fails to resolve and outputs an error value $\perp$. One embodiment of such a decoding algorithm is known in *Computational Fuzzy Extractors* (Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. In *Advances in Cryptology-ASIACRYPT 2013*, pages 174-193. Springer, 2013), in which the decoding algorithm (1) randomly selects rows without replacement $i_1, ..., i_{2n} \leftarrow [1, m]$; (2) restricts A, b to rows $i_1, ..., i_{2n}$ and denotes these $A_{i_1, ..., i_{2n}}, b_{i_1, ..., i_{2n}}$; (3) finds n rows of $A_{i_1, ..., i_{2n}}$ that are linearly independent and, if no such rows exist, outputs $\perp$ and stops; (4) denotes by A', b' the restriction of $A_{i_1, ..., i_{2n}}, b_{i_1, ..., i_{2n}}$ (respectively) to these rows and

computes $x' = (A')^{-1}b'$; (5) returns to step (1) if $b - Ax'$ has more than t nonzero coordinates; and (6) outputs x' .

[0030] The reproduce function 206 further includes a verify function 214 that computes a MAC tag σ'' based on the possibly modified matrix $\tilde{A}$, the possibly modified vector $\widetilde{Ax + w}$, and the shared secret $x_{1,\dots,n/2}$ that was decoded by the decode function 210. The verify function 214 compares the computed MAC tag σ'' with the possibly modified MAC tag $\tilde{\sigma}$. The verify function 214 outputs an acceptance if the possibly modified MAC tag $\tilde{\sigma}$ matches the computed MAC tag σ'' . Otherwise, if the MAC tags do not match, then the verify function 214 outputs a rejection or the error value $\perp$. If both the decode function 210 and the verify function 212 are successful, the output of the reproduce function 206 is the shared secret $x_{1,\dots,n/2}$. If either the decode function 210 or the verify function 212 fails, the output of the reproduce function 206 is the error value $\perp$.

[0031] In some embodiments, the generate function 202 and reproduce function 206 of the robust computational fuzzy extractor 200 are implemented in a silicon blocks in the processor 108 and/or the processor 120 and are accessible as functions in the processors. In other embodiments, the generate function 202 and reproduce function 206 of the fuzzy extractor 200 are implemented using a combination of hardware and software, or purely software (e.g., the authentication programs 114 and 126), preferably using an area of the memory 110 and/or the memory 122 that stores software instructions but cannot be easily modified to ensure that the software instructions for the fuzzy extractor are not altered.

[0032] As discussed in greater detail below, during an authentication process, the generate function 202 generates the helper data P and the shared secret $x_{1,\dots,n/2}$, based on the previously measured response w , the matrix A , and the vector $x_{1,\dots,n}$. The proving

device 104 receives a challenge c (e.g., a bit-string) and possibly modified helper data $\tilde{P}$. The challenge string c is provided to the PUF 118, which outputs a noisy response w' . The reproduce function 206 outputs either the shared secret $x_{1,\dots,n/2}$ or the error value $\perp$. If the proving device 102 is authentic (i.e. includes the unique PUF device 118) and the helper data $\tilde{P}$ has not been modified by an active attacker, then it will successfully reproduce the shared secret $x_{1,\dots,n/2}$ and provide a matching MAC σ'' . However, if the proving device 102 is not authentic (i.e., does not include the unique PUF device 118), then it cannot successfully reproduce the shared secret $x_{1,\dots,n/2}$. Additionally, if the helper data $\tilde{P}$ has been modified, the proving device 102 will not compute a matching MAC σ'' and will know that the source device is an active attacker, rather than the verifying device 104.

[0033] FIG. 3 shows a detailed method 300 for authenticating a device using the robust computational fuzzy extractor 200. In the description of the method, statements that the method is performing some task or function refers to a controller or general purpose processor executing programmed instructions stored in non-transitory computer readable storage media operatively connected to the controller or processor to manipulate data or to operate one or more components in the system 100 to perform the task or function. The processor 108 of the proving device 102 or the processor 120 of the verifying device 104 above may be such a controller or processor and the executed program instructions (e.g., the authentication programs 114 and 126) may be stored in the memories 110 and 122. Additionally, the steps of the methods may be performed in any feasible chronological order, regardless of the order shown in the figures or the order in which the steps are described.

[0034] The method 300 begins with steps of exchanging authentication requests and authentication request acknowledgements between a verifying device and a proving device to prepare for authentication (blocks 302 and 304). Particularly, with reference to the particular embodiments discussed in detail herein, as a preliminary step, the proving device 102 and the verifying device 104 are configured to exchange authentication request and authentication request acknowledgement messages to prepare for authentication of the proving device 102. In one embodiment, the processor 108 of the proving device 102 is configured to operate the transceiver 112 to transmit an authentication request $auth_{Req}$ to the verifying device 104. In at least one embodiment, the authentication request $auth_{Req}$ includes an identifier ID that uniquely identifies the proving device 102. The verifying device 104 receives the authentication request $auth_{Req}$ with the transceiver 124. The processor 120 of the verifying device 104 is configured to compare the received identifier ID with a list of known identifiers stored in the database 128. If the identifier ID is a known identifier, the processor 120 is configured to operate the transceiver 124 to transmit an authentication request acknowledgement $auth_{Ack}$ to the proving device 102. It is noted that in some embodiments, the verifying device 102 is configured to transmit an authentication request $auth_{Req}$ to the proving device 102 and the proving device 102 is configured to transmit an authentication request acknowledgement $auth_{Ack}$, with the identifier ID , to the verifying device 104.

[0035] The method 300 continues with a step of reading a previously measured response bit string of the physical unclonable function of the proving device to the challenge bit string from a memory of the verifying device (block 306). Particularly, in some embodiments, for each known identifier ID , the database 128 of the verifying

device 104 stores a plurality of challenge-response bit string pairs (c_i, w_i) , where each response bit string w_i is a measured response of the PUF 118 (which corresponds to a respective identifier ID) to a different challenge bit string c_i . In at least one embodiment, the pairs generated at a time of manufacture of the proving device 102, at a time of manufacture of the particular component which comprises the PUF 118, or otherwise prior to the authentication process. After or in response to receiving the authentication request $auth_{Req}$ from the proving device 102, the processor 120 of the verifying device 104 is configured to read a previously measured response bit string w . In one embodiment, the processor 120 is configured to select the measured response bit string w from the plurality of measured response bit strings w_i using a time table or other rule set for deciding which measured response bit string w is to be utilized.

[0036] The method 300 continues with a step of generating a shared secret bit string, public helper data, and a MAC tag using the generate function of the robust computational fuzzy extractor (block 308). Particularly, the processor 120 of the verifying device 104 is configured to derive a shared secret bit string $x_{1,...,n/2}$ from a uniformly distributed random vector $x_{1,...,n}$, using the generate function 202 of the robust computation fuzzy extractor 200 as discussed above with respect to FIG. 2. Furthermore, the processor 120 is configured to encode a helper bit string $Ax + w$ by multiplying a uniformly distributed random matrix A with the uniformly distributed random vector $x_{1,...,n}$ and adding the previously measured response bit string w to a result of the multiplication, using the encode function 204 of the generate function 202 of the robust computation fuzzy extractor 200 as discussed above with respect to FIG. 2. Finally, the processor 120 is configured to compute a MAC tag σ based on the shared secret bit

string $x_{1,\dots,n/2}$, the uniformly distributed random matrix A , and the helper bit string $Ax + w$, using the MAC function 212 of the generate function 202 of the robust computation fuzzy extractor 200 as discussed above with respect to FIG. 2. In one embodiment, the processor 120 is configured to generate the uniformly distributed random vector $x_{1,\dots,n}$ at the time of authentication. In one embodiment, the processor 120 is configured to generate the uniformly distributed random matrix A , which is considered part of the public helper data P , at the time of manufacture of the proving device 102 or at some other prior time. However, in some alternative embodiments, the verifying device receives the uniformly distributed random matrix A from the proving device 102 at the time of authentication or at some other prior time.

[0037] The method 300 continues with steps of transmitting the helper bit string and the MAC tag from the verifying device (block 310) and receiving the helper bit string and the MAC tag at the proving device (block 312). Particularly, the processor 120 of the verifying device 104 operates the transceiver 124 to transmit at least the helper bit string $Ax + w$ and the MAC tag σ to the proving device 102. The proving device 102 is configured to receive the possibly modified helper bit string $\widetilde{Ax + w}$ and the possibly modified MAC tag $\tilde{\sigma}$ with the transceiver 112. As noted above, the helper bit string $\widetilde{Ax + w}$ and the MAC tag $\tilde{\sigma}$ may be the same as the helper bit string $Ax + w$ and the MAC tag σ , but may also have been modified by an active attacker. In some embodiments, the verifying device 104 transmits all of the public helper data P , including the uniformly distributed random matrix A , to the proving device 102.

[0038] The method 300 continues with steps of transmitting a challenge bit string from the verifying device (block 314) and receiving the challenge bit string at the proving

device (block 316). Particularly, as discussed above, the database 128 of the verifying device 104 stores a plurality of challenge-response bit string pairs (c_i, w_i) , where each response bit string w_i is a measured response of the PUF 118 to a different challenge bit string c_i . The processor 120 of the verifying device 104 is configured to operate the transceiver 124 to transmit, to the proving device 102, the challenge bit string c which corresponds to the response bit string w that was used by the verifying device 104 to generate the helper bit string $Ax + w$. The proving device 102 is configured to receive the challenge bit string c with the transceiver 112.

[0039] In some alternative embodiments, the challenge bit string c may be installed on onto the memory 110 of the proving device 102 at a time of manufacture. In such embodiments, the step of transmitting the challenge bit string c from the verifying devices 104 (block 314) may be omitted. Instead, the processor 108 of the proving device 102 is configured to read the challenge bit string c from the memory 110. In one embodiment, the processor 108 reads the challenge bit string c in response to receiving the helper bit string $\widetilde{Ax + w}$ and the MAC tag $\tilde{\sigma}$. In some embodiments a plurality of challenge bit strings c_i are stored in the memory 110. In one embodiment, the processor 108 is configured to select a challenge bit string c from the plurality of challenge bit strings c_i using a time table or other rule set for deciding which challenge bit string c is to be utilized.

[0040] The method 300 continues with a step of measuring a response bit string of a physical unclonable function of the proving device to the challenge bit string (block 318). Particularly, the processor 108 of the proving device 102 is configured to provide the

challenge bit string c as an input to the PUF 118. The processor 108 measures, receives, or otherwise determines a noisy response w' of the PUF 118 to the challenge bit string c .

[0041] The method 300 continues with a step of reproducing the shared secret bit string using the reproduce function of the robust computational fuzzy extractor (block 320). Particularly, the processor 108 of the proving device 102 is configured to subtract the noisy response w' from the possibly modified helper bit string $\widetilde{Ax + w}$, using the subtract function 208 of the reproduce function 206 of the robust computation fuzzy extractor 200 as discussed above with respect to FIG. 2. Furthermore, the processor 108 is configured to attempt to derive the shared secret $x_{1,\dots,n/2}$ by decoding the result of the subtraction $\widetilde{Ax + \delta}$, using the decode function 210 of the reproduce function 206 of the robust computation fuzzy extractor 200 as discussed above with respect to FIG. 2. As discussed above, the decode function 210 is able to output the shared secret $x_{1,\dots,n/2}$ if the response w' is sufficiently close and/or similar to the response w according to some predefined metric (e.g., Hamming distance between w' and w is less than t). Otherwise, if the response w' is not sufficiently close and/or similar to the response w (e.g., Hamming distance between w' and w is greater than t), the decode function 210 fails to resolve and outputs an error value $\perp$. In one embodiment, the processor 108 is configured to receive a possibly modified uniformly distributed random matrix $\tilde{A}$, which is considered part of the public helper data $\tilde{P}$, from the verifying device 104 at the time of authentication, alongside the helper bit string $\widetilde{Ax + w}$ and the MAC tag $\tilde{\sigma}$, or at some other prior time. However, in many embodiments, the uniformly distributed random matrix A is installed on the memory 110 at a time of manufacture and is reused during different authentications. After the initial deployment of the proving device 102, the verifying

device 104 is configured to only transmit a new uniformly distributed random matrix A if necessary for security-related reasons. In alternative embodiments, the processor 108 of the proving device 102 may generate the uniformly distributed random matrix A and provide it to the verifying device 104 at some prior time.

[0042] Finally, the processor 108 is configured to compute a MAC tag σ'' based on the possibly modified matrix $\tilde{A}$ (or the matrix A is installed on the memory 110 at a time of manufacture), the possibly modified vector $\widetilde{Ax + w}$, and the shared secret $x_{1,\dots,n/2}$ that was decoded by the decode function 210 and compare the computed MAC tag σ'' with the possibly modified MAC tag $\tilde{\sigma}$, using the verify function 214 of the reproduce function 206 of the robust computation fuzzy extractor 200 as discussed above with respect to FIG. 2. As discussed above, the verify function 214 outputs an acceptance if the possibly modified MAC tag $\tilde{\sigma}$ matches the computed MAC tag σ'' . Otherwise, if the MAC tags do not match, then the verify function 214 outputs a rejection or the error value $\perp$.

[0043] The method 300 continues with steps of transmitting, from the proving device, one of (i) a confirmation message indicating that authentication was successful and (ii) a rejection message indicating that authentication was unsuccessful (block 322) and receiving the one of (i) the confirmation message and (ii) the rejection message at the verifying device (block 324). Particularly, in response to the shared secret $x_{1,\dots,n/2}$ being successfully derived from the decoding process and MAC tags matching in the verifying process, the processor 108 of the proving device 102 is configured to operate the transceiver 112 to transmit an authentication confirmation message $auth_{conf}$ to the verifying device 104, which indicates that the proving device 102 and/or the verifying device 104 was successfully authenticated. In response to the decoding process failing to

resolve and outputting the error value $\perp$ and/or the MAC tags not matching in the verifying process and outputting the error value $\perp$, the processor 108 is configured to operate the transceiver 112 to transmit an authentication rejection message $auth_{rej}$ to the verifying device 104, which indicates that the proving device 102 and/or the verifying device 104 was not successfully authenticated.

[0044] FIG. 4 shows a block diagram of a reverse robust computational fuzzy extractor 400. The reverse robust computational fuzzy extractor 400 is similar to the robust computational fuzzy extractor 200, except that the functions performed on the proving device 102 and the verifying device 104 are reversed. Particularly, as discussed in greater detail below, during an authentication process, the proving device 102 receives a challenge c (e.g., a bit-string) from the verifying device 104. The challenge string c is provided to the PUF 118, which outputs a noisy response w' . The generate function 202 outputs the helper data P and the shared secret $x_{1,...,n/2}$, based on the noisy response w' , the matrix A , and the vector $x_{1,...,n}$. Possibly modified helper data $\tilde{P}$ is provided to the verifying device 104 and the reproduce function 206 outputs either the shared secret $x_{1,...,n/2}$ or the error value $\perp$, based a previously measured response w and the possibly modified helper data $\tilde{P}$. If the proving device 102 is authentic (i.e. includes the unique PUF device 118) and the helper data $\tilde{P}$ has not been modified by an active attacker, then verifying device 104 will successfully reproduce the shared secret $x_{1,...,n/2}$ and provide a matching MAC σ'' . However, if the proving device 102 is not authentic (i.e., does not include the unique PUF device 118), then verifying device 104 cannot successfully reproduce the shared secret $x_{1,...,n/2}$. Additionally, if the helper data $\tilde{P}$ has

been modified, the verifying device 104 will not compute a matching MAC σ'' and will know that the source device is an active attacker, rather than the proving device 102.

[0045] FIG. 5 shows a detailed method 500 for authenticating a device using the reverse robust computational fuzzy extractor 400. In the description of the method, statements that the method is performing some task or function refers to a controller or general purpose processor executing programmed instructions stored in non-transitory computer readable storage media operatively connected to the controller or processor to manipulate data or to operate one or more components in the system 100 to perform the task or function. The processor 108 of the proving device 102 or the processor 120 of the verifying device 104 above may be such a controller or processor and the executed program instructions (e.g., the authentication programs 114 and 126) may be stored in the memories 110 and 122. Additionally, the steps of the methods may be performed in any feasible chronological order, regardless of the order shown in the figures or the order in which the steps are described.

[0046] The method 500 begins with steps of exchanging authentication requests and authentication request acknowledgements between a verifying device and a proving device to prepare for authentication (blocks 502 and 504), which are essentially similar to the steps 302 and 304 of the method 300 and are not described again in detail.

[0047] The method 500 continues with steps of transmitting a challenge bit string from the verifying device (block 506) and receiving the challenge bit string at the proving device (block 508). Particularly, in some embodiments, the processor 120 of the verifying device 104 operates the transceiver 124 to transmit a challenge bit string c to the proving device 102. In some embodiments, for each known identifier ID , the database 128 of the

verifying device 104 stores a plurality of challenge-response bit string pairs (c_i, w_i) , where each response bit string w_i is a measured response of the PUF 118 (which corresponds to a respective identifier ID) to a different challenge bit string c_i . In at least one embodiment, the pairs are generated at a time of manufacture of the proving device 102, at a time of manufacture of the particular component which comprises the PUF 118, or otherwise prior to the authentication process. After or in response to receiving the authentication request $auth_{Req}$ from the proving device 102, the processor 120 of the verifying device 104 is configured to select a challenge bit string c from the database 128 and operate the transceiver 124 to transmit the challenge bit string c to the proving device 102. In one embodiment, the processor 120 is configured to select the challenge bit string c from the plurality of challenge bit strings c_i using a time table or other rule set for deciding which challenge bit string c is to be utilized. The proving device 102 is configured to receive the challenge bit string c with the transceiver 112.

[0048] In some alternative embodiments, the challenge bit string c may be installed on onto the memory 110 of the proving device 102 at a time of manufacture. In such embodiments, the step of transmitting the challenge bit string c from the verifying devices 104 (block 506) may be omitted. Instead, the processor 108 of the proving device 102 is configured to read the challenge bit string c from the memory 110. In one embodiment, the processor 108 reads the challenge bit string c in response to receiving the authentication request acknowledgement $auth_{Ack}$. In some embodiments a plurality of challenge bit strings c_i are stored in the memory 110. In one embodiment, the processor 108 is configured to select a challenge bit string c from the plurality of challenge bit

strings c_i using a time table or other rule set for deciding which challenge bit string c is to be utilized.

[0049] The method 500 continues with a step of measuring a response bit string of a physical unclonable function of the proving device to the challenge bit string (block 510). Particularly, the processor 108 of the proving device 102 is configured to provide the challenge bit string c as an input to the PUF 118. The processor 108 measures, receives, or otherwise determines a noisy response w' of the PUF 118 to the challenge bit string c .

[0050] The method 500 continues with a step of generating a shared secret bit string, public helper data, and a MAC tag using the generate function of the reverse computational fuzzy extractor (block 512). Particularly, the processor 108 of the proving device 102 is configured to derive a shared secret bit string $x_{1,...,n/2}$ from a uniformly distributed random vector $x_{1,...,n}$, using the generate function 202 of the reverse robust computation fuzzy extractor 400 as discussed above with respect to FIG. 4. Furthermore, the processor 108 is configured to encode a helper bit string $Ax + w'$ by multiplying a uniformly distributed random matrix A with the uniformly distributed random vector $x_{1,...,n}$ and adding the noisy response bit string w' to a result of the multiplication, using the encode function 204 of the generate function 202 of the reverse robust computation fuzzy extractor 400 as discussed above with respect to FIG. 4. Finally, the processor 108 is configured to compute a MAC tag σ based on the shared secret bit string $x_{1,...,n/2}$, the uniformly distributed random matrix A , and the helper bit string $Ax + w'$, using the MAC function 212 of the generate function 202 of the reverse robust computation fuzzy extractor 400 as discussed above with respect to FIG. 4. In one embodiment, the processor 108 is configured to generate the uniformly distributed random vector $x_{1,...,n}$ at

the time of authentication. In one embodiment, the processor 108 is configured to receive the uniformly distributed random matrix A , which is considered part of the public helper data p , from the verifying device 104 at the time of authentication, alongside the challenge bit string c or at some other prior time. However, in many embodiments, the uniformly distributed random matrix A is installed on the memory 110 at a time of manufacture and is reused during different authentications. After the initial deployment of the proving device 102, the verifying device 104 is configured to only transmit a new uniformly distributed random matrix A if necessary for security-related reasons. In alternative embodiments, the processor 108 of the proving device 102 may generate the uniformly distributed random matrix A and provide it to the verifying device 104.

[0051] The method 500 continues with steps of transmitting the helper bit string and the MAC tag from the proving device (block 514) and receiving the helper bit string and the MAC tag at the verifying device (block 516). Particularly, the processor 108 of the proving device 102 operates the transceiver 112 to transmit at least the helper bit string $Ax + w'$ and the MAC tag σ to the verifying device 104. The verifying device 104 is configured to receive a possibly modified helper bit string $\widetilde{Ax + w}$ and a possibly modified MAC tag $\tilde{\sigma}$ with the transceiver 124. As noted above, the helper bit string $\widetilde{Ax + w}$ and the MAC tag $\tilde{\sigma}$ may be the same as the helper bit string $Ax + w$ and the MAC tag σ , but may also have been modified by an active attacker. In some embodiments, the proving device 102 transmits all of the public helper data P , including the uniformly distributed random matrix A , to the verifying device 104.

[0052] The method 500 continues with a step of reading a previously measured response bit string of the physical unclonable function of the proving device to the

challenge bit string from a memory of the verifying device (block 518). Particularly, as discussed above, the database 128 of the verifying device 104 stores a plurality of challenge-response bit string pairs (c_i, w_i) , where each response bit string w_i is a measured response of the PUF 118 to a different challenge bit string c_i . After receiving the helper bit string $\widetilde{Ax + w'}$, the processor 120 of the verifying device 104 is configured to read the previously measured response bit string w which corresponds to the challenge bit string c that was used by the proving device 102 to generate the helper bit string $\widetilde{Ax + w'}$.

[0053] The method 500 continues with a step of reproducing the shared secret bit string using the reproduce function of the reverse computational fuzzy extractor (block 520). Particularly, the processor 120 of the verifying device 104 is configured to subtract the previously measured response bit string w from the possibly modified helper bit string $\widetilde{Ax + w'}$, using the subtract function 208 of the reproduce function 206 of the reverse robust computation fuzzy extractor 400 as discussed above with respect to FIG. 4. Furthermore, the processor 120 is configured to attempt to derive the shared secret $x_{1,...,n/2}$ by decoding the result of the subtraction $\widetilde{Ax + \delta}$, using the decode function 210 of the reproduce function 206 of the reverse robust computation fuzzy extractor 400 as discussed above with respect to FIG. 4. As discussed above, the decode function 210 is able to output the shared secret $x_{1,...,n/2}$ if the response w' is sufficiently close and/or similar to the response w according to some predefined metric (e.g., Hamming distance between w' and w is less than t). Otherwise, if the response w' is not sufficiently close and/or similar to the response w (e.g., Hamming distance between w' and w is greater than t), the decode function 210 fails to resolve and outputs an error value $\perp$.

[0054] Finally, the processor 120 is configured to compute a MAC tag σ'' based on the possibly modified matrix $\tilde{A}$ (or the matrix A is installed on the memory 122), the possibly modified vector $\widetilde{Ax + w'}$, and the shared secret $x_{1,\dots,n/2}$ that was decoded by the decode function 210 and compare the computed MAC tag σ'' with the possibly modified MAC tag $\tilde{\sigma}$, using the verify function 214 of the reproduce function 206 of the reverse robust computation fuzzy extractor 400 as discussed above with respect to FIG. 4. As discussed above, the verify function 214 outputs an acceptance if the possibly modified MAC tag $\tilde{\sigma}$ matches the computed MAC tag σ'' . Otherwise, if the MAC tags do not match, then the verify function 214 outputs a rejection or the error value $\perp$.

[0055] The method 500 continues with steps of transmitting, from the verifying device, one of (i) a confirmation message indicating that authentication was successful and (ii) a rejection message indicating that authentication was unsuccessful (block 522) and receiving the one of (i) the confirmation message and (ii) the rejection message at the proving device (block 524). Particularly, in response to the shared secret $x_{1,\dots,n/2}$ being successfully derived from the decoding process and MAC tags matching in the verifying process, the processor 120 of the verifying device 104 is configured to operate the transceiver 124 to transmit an authentication confirmation message $auth_{conf}$ to the proving device 102, which indicates that the proving device 102 and/or the verifying device 104 was successfully authenticated. In response to the decoding process failing to resolve and outputting the error value $\perp$ and/or the MAC tags not matching in the verifying process and outputting the error value $\perp$, the processor 120 is configured to operate the transceiver 124 to transmit an authentication rejection message $auth_{rej}$ to proving device

102, which indicates that the proving device 102 and/or the verifying device 104 was not successfully authenticated.

[0056] The herein described methods (e.g., the methods 300, 500 and/or the robust computational fuzzy extractors 200, 400) improve the functioning of the proving device 102, and the verifying device 104, respectively or in combination by enabling it/them to operate more securely to authenticate the proving device 102. Particularly, by including the processes of generating a MAC tag on the verifying device 104 and using the MAC tag to verify the authenticity and integrity of transmitted helper data on the proving device, the system is more secure against active attackers that may modify the helper data. Additionally, the MAC tag is efficiently generated and verified because the uniformly distributed random matrix A is reused for hashing and signing. In this way, the method can advantageously be implemented with minimal or no increase in memory or processing capability. Furthermore, in the case of the reverse robust fuzzy extractor 400, the devices operate more efficiently to authenticate the proving device 102. Particularly, in one embodiment, the generate function 202 runs in $O(n^2)$ and the reproduce function 206 runs in $O(n^4)$. In the reverse robust fuzzy extractor 400, the computationally expensive reproduce function 206 is implemented on the verifying device 104, and the less computationally expensive generate function 202 is implemented on the proving device 102. In this way, the method can advantageously be implemented on light-weight proving devices 102. Additionally, due the uniform random distribution of the matrix A and the vector $x_{1,...,n}$, any statistical bias in the response of the PUF 118 is masked, without any additional steps required, thereby minimizing information leakage and

improving the security of the system 100. Finally, the full entropy of the PUF 118 is retained and longer cryptographic keys can be obtained.

[0057] While the disclosure has been illustrated and described in detail in the drawings and foregoing description, the same should be considered as illustrative and not restrictive in character. It is understood that only the preferred embodiments have been presented and that all changes, modifications and further applications that come within the spirit of the disclosure are desired to be protected.

What is claimed is:

1. A method of authenticating a first device, the method comprising:

receiving, with a transceiver of the first device, a helper bit string and a first message authentication code tag from a second device that is remote from the first device;

measuring a first response bit string of a physical unclonable function of the first device with respect to a challenge bit string, the physical unclonable function being provided by a component of the first device;

subtracting, with a processor of the first device, the first response bit string from the helper bit string;

decoding, with the processor of the first device, a result of the subtraction using a uniformly distributed random matrix, the shared secret bit string being provided from the decoding if the helper bit string was encoded using a previously measured second response bit string that is within a threshold level of similarity to the first response bit string, the decoding outputting an error value otherwise;

determining, with the processor of the first device, a second message authentication code tag based on the shared secret bit string, the uniformly distributed random matrix, and the helper bit string; and

determining, with the processor of the first device, whether the second message authentication code tag matches the first message authentication code tag.

2. The method of claim 1 further comprising:

transmitting, with the transceiver of the first device, one of (i) a confirmation message indicating that authentication was successful and (ii) a rejection message indicating that authentication was unsuccessful to the second device, the confirmation message being transmitted only if shared secret bit string was provided from the decoding and the second message authentication code tag matches the first message authentication code tag, the rejection message being transmitted otherwise.

3. The method of claim 1 further comprising:

reading, with a processor of the second device, the second response bit string from a memory storage of the second device, the second response bit string being previously measured and stored at a time of manufacture of the first device using the physical unclonable function of the first device;

deriving, with the processor of the second device, the shared secret bit string from a uniformly distributed random vector;

encoding, with the processor of the second device, the helper bit string by multiplying the uniformly distributed random matrix with the uniformly distributed random vector and adding the second response bit string to a result of the multiplication;

determining, with the processor of the second device, the first message authentication code tag based on the shared secret bit string, the uniformly distributed random matrix, and the helper bit string; and

transmitting, with a transceiver of the second device, the helper bit string and the first message authentication code tag to the first device.

4. The method of claim 1 further comprising:

receiving, with the transceiver of the first device, the challenge bit string from the second device.

5. The method of claim 1 further comprising:

reading the challenge bit string from a memory storage of the first device.

6. The method of claim 1 further comprising:

reading, with the processor of the first device, the uniformly distributed random matrix from a memory storage of the first device, the uniformly distributed random matrix being stored in the memory storage at a time of manufacture of the first device.

7. The method of claim 1 further comprising:

receiving, with the transmitter of the first device, the uniformly distributed random matrix from the second device; and

storing, with the processor of the first device, the received uniformly distributed random matrix in a memory storage of the first device, the received uniformly distributed random matrix replacing any currently stored version of the uniformly distributed random matrix in the memory storage of the first device.

8. The method of claim 1 further comprising:

transmitting, with the transmitter of the first device, an authentication request message, the authentication request message including an identifier of the first device.

9. The method of claim 1, wherein the physical unclonable function is provided by one of the processor of the first device and a memory storage of the first device.

10. The method of claim 1, wherein the physical unclonable function is provided by one of a semiconductor component of the first device and a microelectromechanical system of the first device that is configured to provide the physical unclonable function.

11. A method of authenticating a first device, the method comprising:

measuring a first response bit string of a physical unclonable function of the first device with respect to a challenge bit string, the physical unclonable function being provided by a component of the first device;

deriving, with a processor of the first device, a shared secret bit string from a uniformly distributed random vector;

encoding, with the processor of the first device, a helper bit string by multiplying a uniformly distributed random matrix with the uniformly distributed random vector and adding the first response bit string to a result of the multiplication;

determining, with the processor of the first device, a first message authentication code tag based on the shared secret bit string, the uniformly distributed random matrix, and the helper bit string; and

transmitting, with a transceiver of the first device, the helper bit string and the first message authentication code tag to a second device that is remote from the first device.

12. The method of claim 11 further comprising:

receiving, with a transceiver of the second device, the helper bit string and the first message authentication code tag from the first device;

reading, with a processor of the second device, a previously measured second response bit string of the physical unclonable function of the first device with respect to the challenge bit string from a memory storage of the second device, the second response bit string being measured and stored at a time of manufacture of the first device;

subtracting, with the processor of the second device, the second response bit string from the helper bit string;

decoding, with the processor of the second device, a result of the subtraction using the uniformly distributed random matrix, the shared secret bit string being provided from the decoding if the first response bit string is within a threshold level of similarity to the second response bit string, the decoding outputting an error value otherwise;

determining, with the processor of the second device, a second message authentication code tag based on the shared secret bit string, the uniformly distributed random matrix, and the helper bit string; and

determining, with the processor of the second device, whether the second message authentication code tag matches the first message authentication code tag; and

transmitting, with the transceiver of the second device, one of (i) a confirmation message indicating that authentication was successful and (ii) a rejection message indicating that authentication was unsuccessful to the second device, the confirmation message being transmitted only if shared secret bit string was provided from the decoding

and the second message authentication code tag matches the first message authentication code tag, the rejection message being transmitted otherwise.

13. The method of claim 12 further comprising:

receiving, with the transceiver of the first device, the one of (i) the confirmation message and (ii) the rejection message from the second device.

14. The method of claim 11 further comprising:

receiving, with the transceiver of the first device, the challenge bit string from the second device.

15. The method of claim 11 further comprising:

reading the challenge bit string from a memory storage of the first device.

16. The method of claim 11 further comprising:

reading, with the processor of the first device, the uniformly distributed random matrix from a memory storage of the first device, the uniformly distributed random matrix being stored in the memory storage at a time of manufacture of the first device.

17. The method of claim 11 further comprising:

receiving, with the transmitter of the first device, the uniformly distributed random matrix from the second device; and

storing, with the processor of the first device, the received uniformly distributed random matrix in a memory storage of the first device, the received uniformly distributed random matrix replacing any currently stored version of the uniformly distributed random matrix in the memory storage of the first device.

18. The method of claim 11 further comprising:

transmitting, with the transmitter of the first device, an authentication request message, the authentication request message including an identifier of the first device.

19. The method of claim 11, wherein the physical unclonable function is provided by one of the processor of the first device and a memory storage of the first device.

20. The method of claim 11, wherein the physical unclonable function is provided by a microelectromechanical system of the first device that is configured to provide the physical unclonable function.

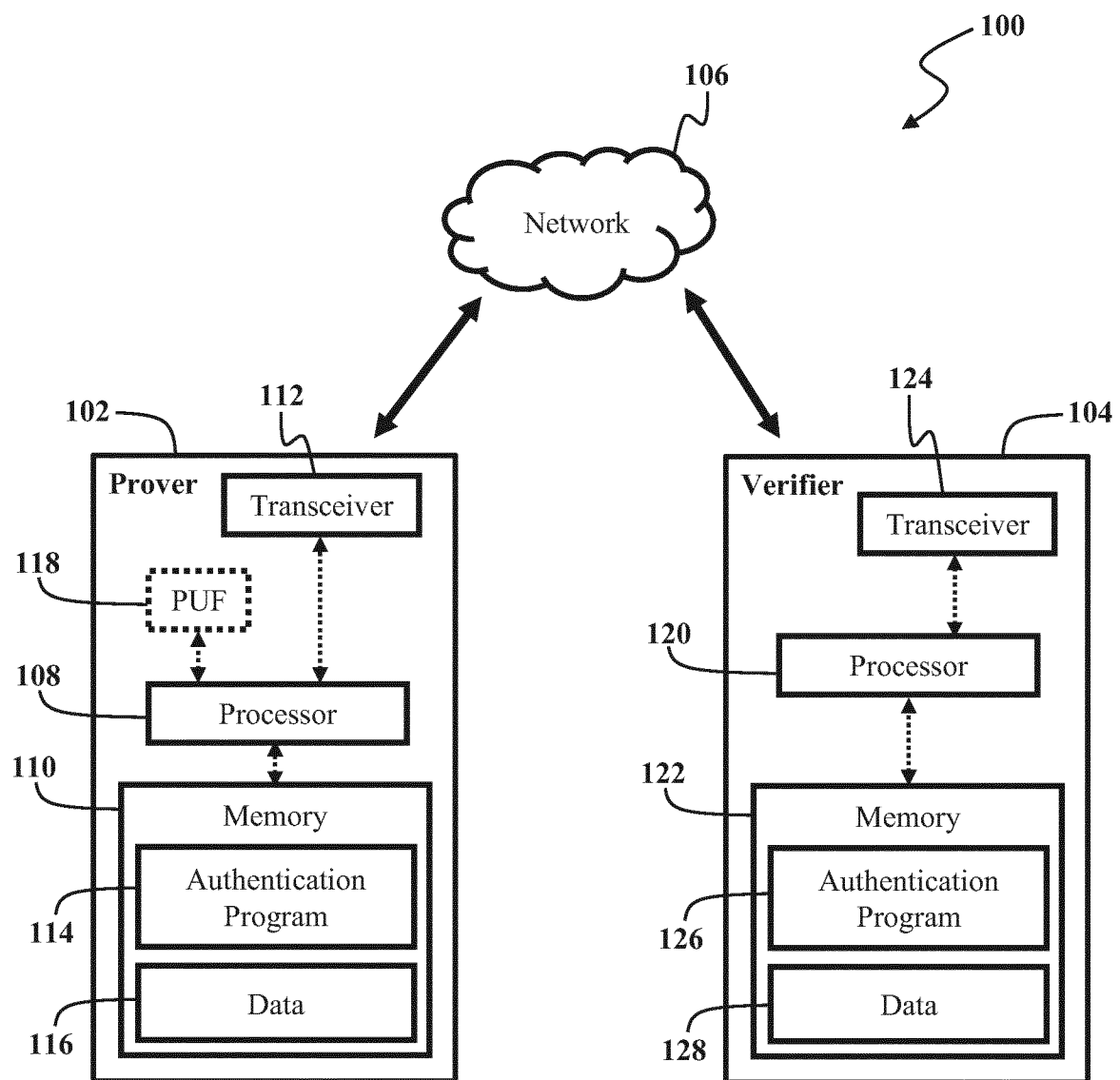


FIG. 1

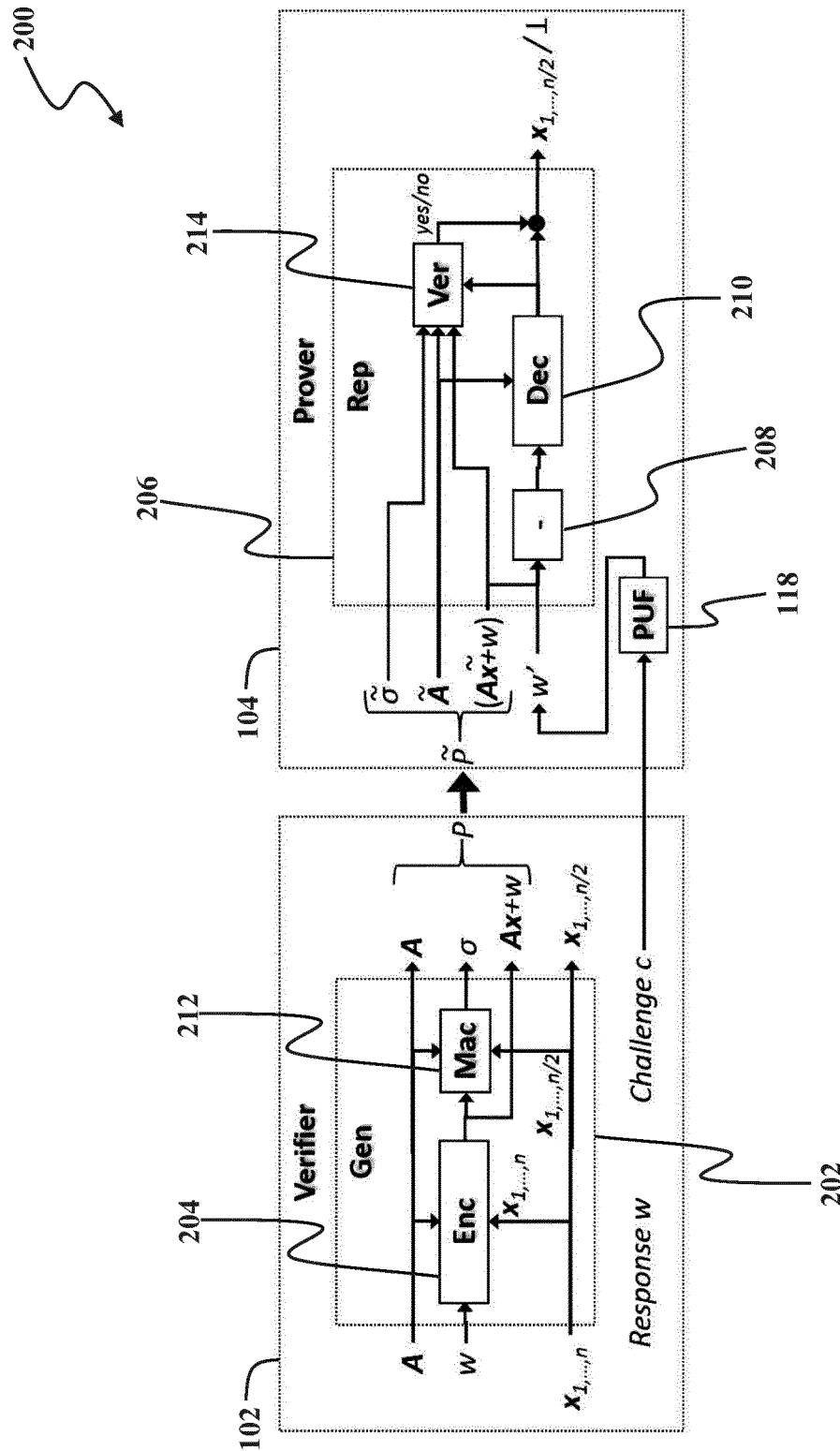


FIG. 2

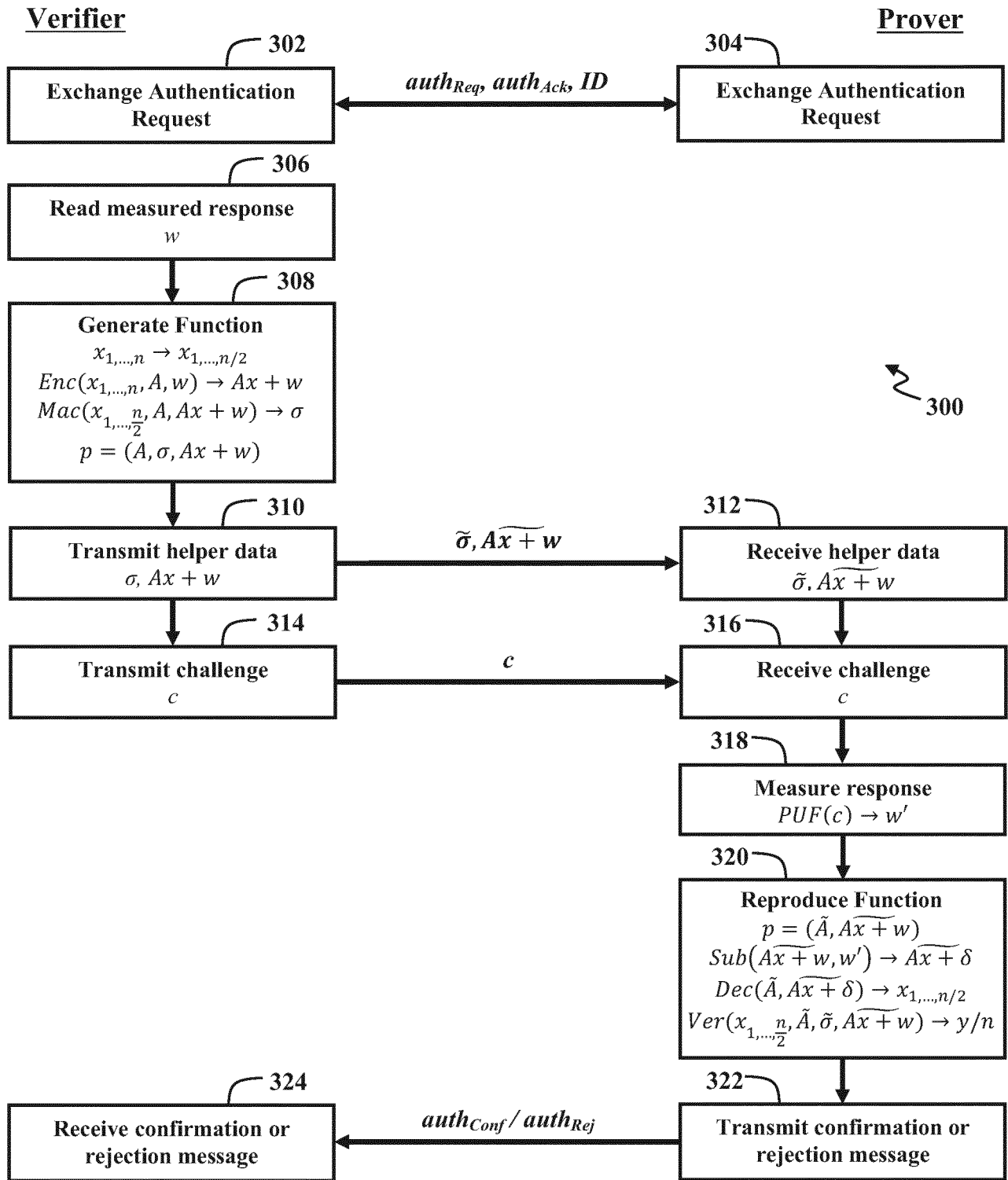


FIG. 3

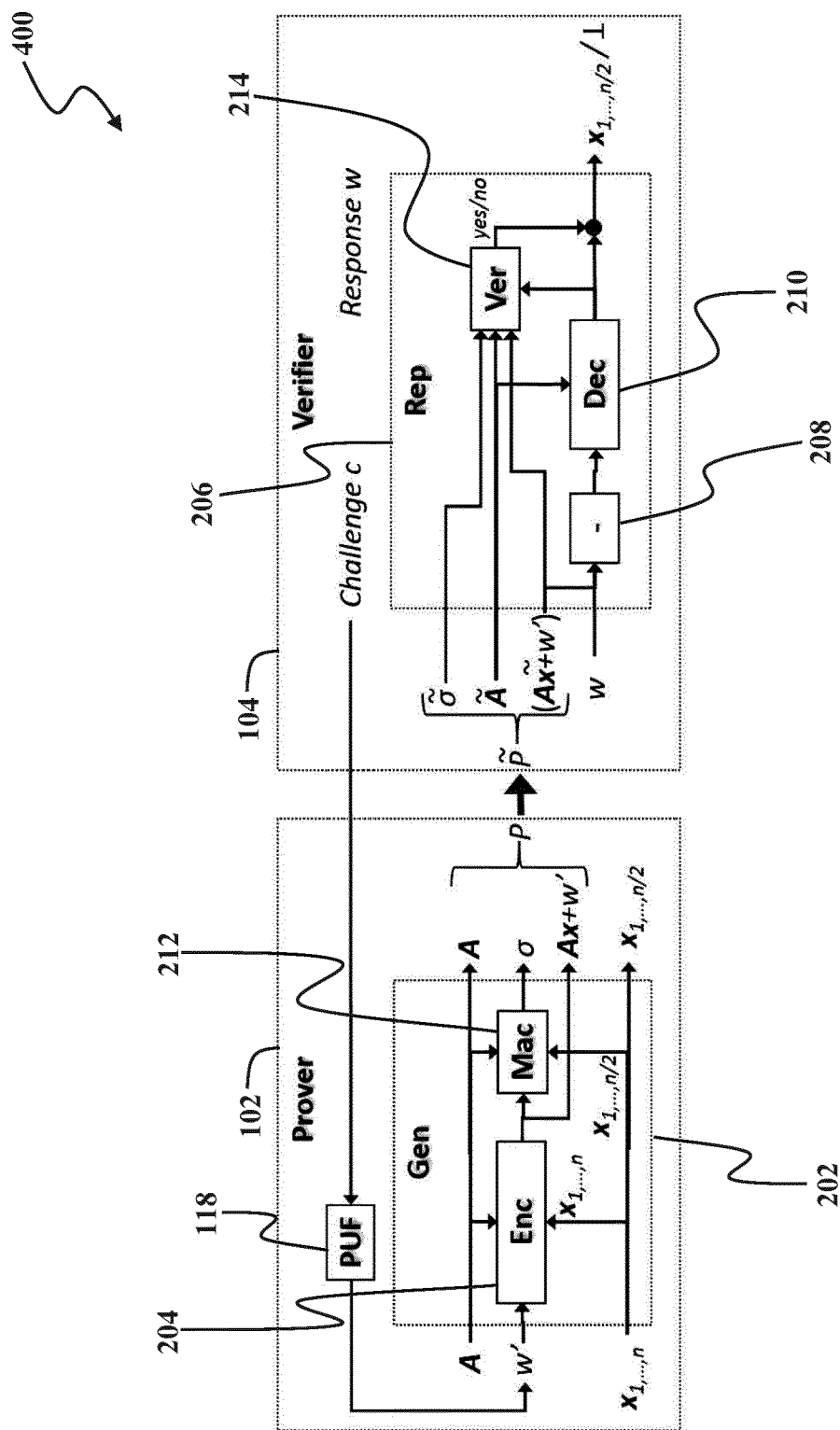


FIG. 4

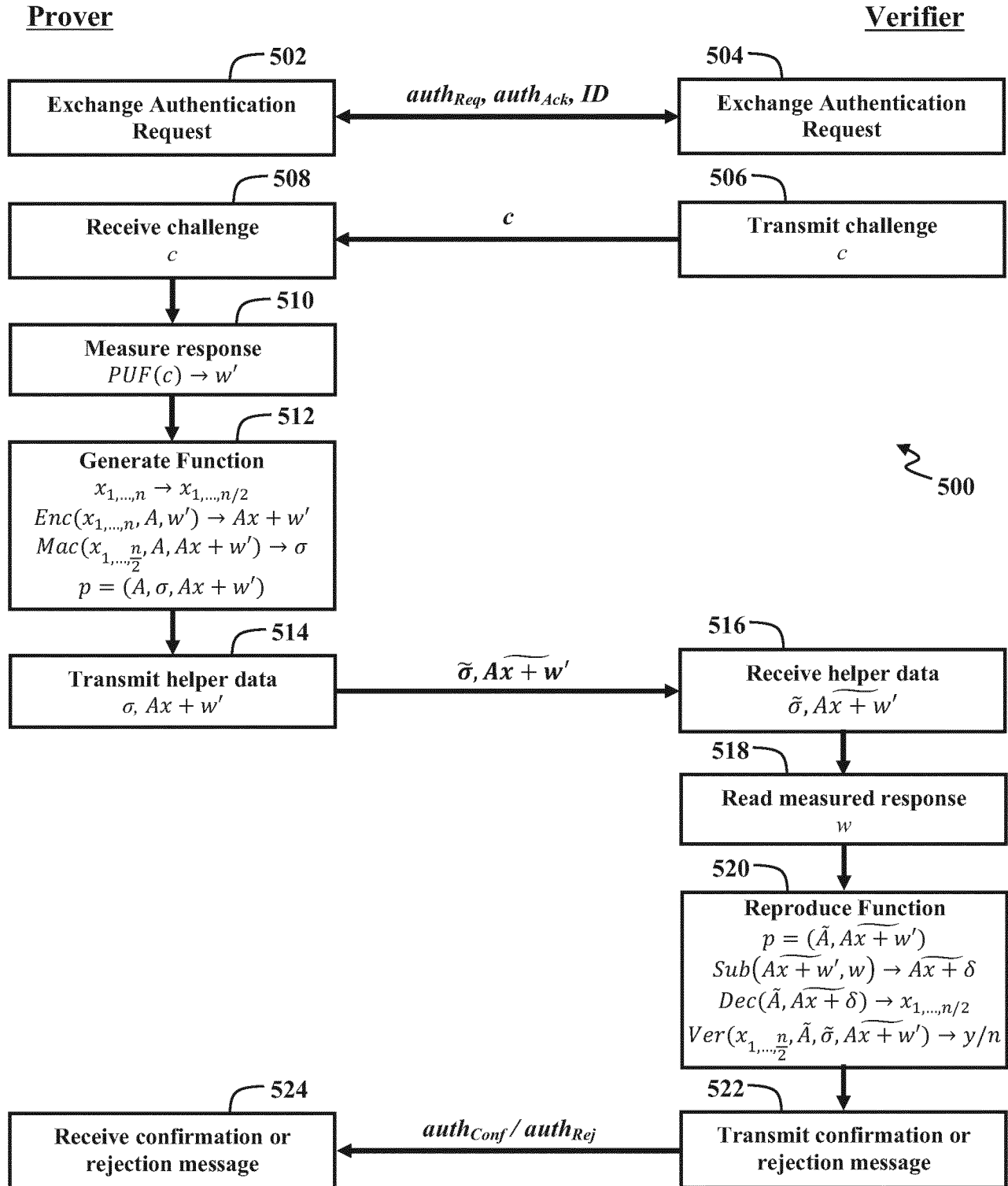


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2017/084603

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/32
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHRISTOPHER HUTH ET AL: "Securing Systems with Scarce Entropy: LWE-Based Lossless Computational Fuzzy Extractor for the IoT", INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH,, vol. 20161015:190825, 11 October 2016 (2016-10-11), pages 1-33, XP061021932, [retrieved on 2016-10-11]	1-20
Y	sections 3 and 4; figures 3,4 ----- -/--	1-10



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

14 March 2018

Date of mailing of the international search report

22/03/2018

Name and mailing address of the ISA/

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

Authorized officer

Billet, Olivier

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2017/084603

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	ANTHONY VAN HERREWEGE ET AL: "Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs", 2 March 2012 (2012-03-02), FINANCIAL CRYPTOGRAPHY AND DATA SECURITY, SPRINGER BERLIN HEIDELBERG, BERLIN, HEIDELBERG, PAGE(S) 374 - 389, XP047013844, ISBN: 978-3-642-32945-6 sections 1 and 3; figure 1 -----	1-10
T	FULLER BENJAMIN ET AL: "Computational Fuzzy Extractors", 1 December 2013 (2013-12-01), ECCV 2016 CONFERENCE; [LECTURE NOTES IN COMPUTER SCIENCE; LECT.NOTES COMPUTER], SPRINGER INTERNATIONAL PUBLISHING, CHAM, PAGE(S) 174 - 193, XP047452656, ISSN: 0302-9743 ISBN: 978-3-642-33485-6 sections 1 and 4 -----	1-20