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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 September 2010 (23.09.2010)

PCT

(10) International Publication Number
WO 2010/107902 A2

(51) International Patent Classification:
H04L 9/14 (2006.01) *G06Q 50/00* (2006.01)

(21) International Application Number:
PCT/US2010/027646

(22) International Filing Date:
17 March 2010 (17.03.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/161,304 18 March 2009 (18.03.2009) US

(71) Applicant *(for all designated States except US)*: **SZREK2SOLUTIONS, LLC** [US/US]; 60 Spencer Avenue, East Greenwich, Rhode Island 02818 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **SZREK, Walter** [US/US]; c/o Szrek2Solutions, LLC, 60 Spencer Avenue, East Greenwich, Rhode Island 02818 (US). **SZREK, Irene** [US/US]; c/o Szrek2Solutions, LLC, 60 Spencer Avenue, East Greenwich, Rhode Island 02818 (US).

(74) Agents: **JONES, Kevin, K.** et al.; Armstrong Teasdale, LLP, One Metropolitan Square, 26th Floor, St. Louis, Missouri 63102 (US).

(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, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) **Title**: SECURE PROVISIONING OF RANDOM NUMBERS TO REMOTE CLIENTS

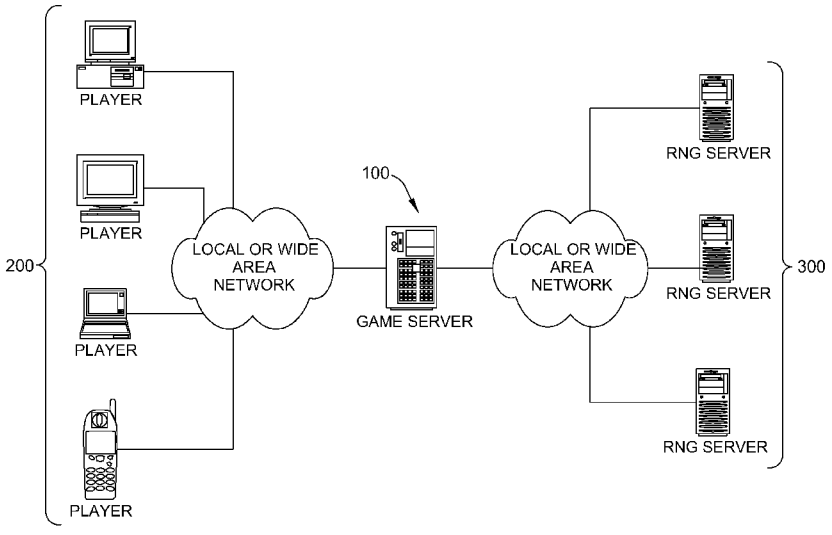


FIG. 1

(57) **Abstract**: The present disclosure provides a method for protecting the "identity" of random numbers used in multiplayer games, which prevents fraud related to access to random game elements by insiders. The method enables a distribution of random numbers by a game server in a way that the game server does not know the "identity" of random numbers but the game server may independently verify these random numbers. The method includes receiving transformation information, generating random numbers, transforming random numbers according to the transformation information, sending the transformed random numbers to a game server, securing, via the game server, the transformed random numbers, sending the transformed random numbers from the game server to a remote client, and transforming the transformed random numbers to recreate the generated random numbers.

WO 2010/107902 A2

SECURE PROVISIONING OF RANDOM NUMBERS TO REMOTE CLIENTS

BACKGROUND

[0001] This patent application claims the benefit of U.S. Provisional Patent Application Serial No. 61/161304, filed on March 18, 2009, the entirety of which is hereby incorporated by reference herein.

[0002] Game servers require random numbers for variety of reasons, for example, to draw winning numbers, to generate game outcomes, and to distribute cards. To prevent attacks on random numbers generated by game servers, conventional gaming regulators may require external (e.g., independent) random number generator (RNG) servers be used for random number generation. Security guidelines/standards set by, for example, the World Lottery Association, recommend that external RNG systems are used for random number generation for, for example, draws or instant winnings generation. As a result of required system redundancy, multiple RNG servers are often used.

[0003] However, use of multiple RNG servers introduces issues for, for example, games that require a state of random numbers to be preserved, games that issue cards to different players from a same deck of cards, and/or games that issue random numbers for individual players where the issued random numbers are dependent upon previously generated random numbers. An exemplary issue for game servers is vulnerability to an attack by insiders illegally accessing a game server which makes it possible, for example, to learn what cards have been dealt to a particular player.

[0004] Therefore, a system and method are needed where random numbers can be generated on any RNG server and can not be revealed to persons having access to a game server. For example, what is needed is a system that includes external RNG servers where a random number generation process is provided such that access to a game server does not permit access to random numbers (e.g., cards) that have been issued to a player.

SUMMARY

[0005] In embodiments, described herein is a method for protecting random numbers. The method includes receiving encrypted transformation information, decrypting the encrypted transformation information, generating random numbers, transforming random numbers according to the decrypted transformation information, sending the transformed random numbers to a game server, protecting, via the game server, the transformed random numbers, sending the transformed random numbers from the game server to a remote client, and transforming the transformed random numbers to recreate the generated random numbers.

[0006] In further embodiments, described herein is a method for protecting random numbers using asymmetric encryption. The method includes receiving a public key of a remote client by a random number server, generating random numbers, encrypting the random numbers with the public key, sending the encrypted random numbers to a game server, protecting the encrypted random numbers via the game server, sending the encrypted random numbers from the game server to a remote client, decrypting the encrypted random numbers with a private key by a remote client, and recreating the generated random numbers.

[0007] In still further embodiments, described herein is a system for playing card games. The system includes a random number generation server, a remote client, and one or more processors. The one or more processors are programmed to receive transformed random numbers from the random number generation server, secure transformation information, and verify random numbers.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The embodiments described herein may be better understood by referring to the following description in conjunction with the accompanying drawings.

[0009] Figure 1 is an exemplary block diagram of a gaming environment.

[0010] Figure 2 is an exemplary block diagram for providing random numbers to a remote client from random number generator (RNG) servers without a game server knowing actual values of the provided random numbers.

[0011] Figure 3 is an exemplary block diagram for providing random numbers to player devices from RNG severs without knowledge from a game server of actual values of the provided random numbers.

[0012] Figures 4 and 5 are exemplary block diagrams of a verification of random numbers provided by a remote client to a game server.

[0013] Figure 6 is an exemplary flow chart illustrating a process for protecting random numbers.

DETAILED DESCRIPTION

[0014] The present disclosure provides systems and methods for protecting an "identity" of random numbers used in multiplayer games to, for example, prevent fraud related to access to random game elements by insiders. The systems and methods described herein enable a distribution of random numbers by a game server in a way that the game server does not know the "identity" of random numbers but the game server may independently verify these random numbers.

[0015] While the making and use of various embodiments of the present disclosure are discussed in detail below, the embodiments of the present disclosure provide many applicable inventive concepts that may be embodied in a wide variety of specific contexts. The specific embodiments discussed herein are merely illustrative of specific ways to make and use the invention and do not delimit the scope of the present disclosure.

[0016] To facilitate the understanding of the present disclosure, a number of terms are defined below. Terms defined herein have meanings as commonly understood by a person of ordinary skill in the areas relevant to the embodiments of the present disclosure. Terms such as "a," "an" and "the" are not intended to refer to only a singular entity, but include the general class of which a

specific example may be used for illustration. The terminology herein is used to describe specific embodiments of the present disclosure, but their usage does not delimit the present disclosure, except as outlined in the claims.

[0017] The order of execution or performance of the operations in embodiments of the present disclosure illustrated and described herein is not essential, unless otherwise specified. For example, it is contemplated that executing or performing a particular operation before, contemporaneously with, or after another operation is within the scope of aspects of the present disclosure. Embodiments of the present disclosure may include additional or fewer operations than those disclosed herein.

[0018] Provided herein are systems and methods for securing random data generated on one or more random number generator (RNG) servers (300). In embodiments, the random data generated on one or more RNG servers (300) are not provided to game server (100), and therefore, cannot be obtained by an attack on game server (100).

[0019] Figure 1 is an exemplary gaming environment. One or more remote clients (200) may communicate directly or indirectly with game server (100) via, for example, a network. Game server (100) communicates directly or indirectly with one or more RNG server(s) (300) via, for example, a network. In embodiments, a remote client (200) and RNG server(s) (300) use a transformation of random numbers (e.g., cards) that cannot be interpreted directly by a game server (100). In one embodiment, only transformed random numbers are passed between RNG server(s) (300) and remote client (200) while original random numbers are used by remote client (200). However, one of ordinary skill in the art guided by the teaching herein will appreciate that there may be multiple ways of verifying random numbers presented by remote client (200) to game server (100).

[0020] In embodiments, there are multiple different transformations of random numbers that are possible. For example, a transformation may be one or more of the following: mapping a set of numbers into another set of numbers,

standard symmetric or asymmetric encryption, algorithmic transformation, and a combination of the above transformations.

[0021] Following is an example of mapping a set of numbers into another set of numbers in a game using a deck of cards (e.g., a standard deck of 52 different cards). Mapping is defined by ordering cards in a certain described way. For example, the 2 of clubs may be assigned a first position, followed by the 3 of clubs in the second position, and so on up to the ace of clubs. Diamonds, hearts, and spades are ordered similarly from the 2 to the ace, respectively. Each card is represented by a number corresponding to a position/natural order of the card. The numbers/position assigned to each card range from, for example, 1-52. In this example, position number 1 represents the 2 of clubs, position number 2 represents the 3 of clubs . . . position number 14 represents the 2 of diamonds . . . position number 27 represents the 2 of hearts, and so on. Next, 52 uniform different random numbers are generated (e.g., 27, 23, 11, 3, 51…) from a range of 1-52. The generated random numbers are mapped to the assigned numbers representing each of the cards. This mapping is then used to encode random numbers received by remote client (200). That is, as the generated random numbers are received, the generated random numbers are assigned a new number according to the order in which they are received. For example, when a random number is received in position number 1 (e.g., random number 27) the value of original number 27 is used (e.g., the 2 of hearts) replacing the previous value of position number 1 (e.g., the 2 of clubs), and when a random number is received in position number 2 (e.g., random number 23) the value of original number 23 is used (e.g., the jack of diamonds) replacing the previous value of position number 2 (e.g., the 3 of clubs). This process of "mapping" is continued for each of the remaining 52 numbers. One of ordinary skill in the art guided by the teaching herein will appreciate that any other mapping method could be used, as long as the mapping method enables a representation of each random number.

[0022] When using encryption as a method of transformation, encryption with an encryption key provided by remote client (200) may be used by RNG server(s) (300). In case of symmetric encryption, such as 3DES or AES, a secret key may be used to encrypt generated random numbers. In this example, a

secret key is sent in a secure way between remote client (200) and RNG server(s) (300). RNG server(s) (300) use this secret key to encrypt random numbers sent back to remote client (200). Remote client (200) decrypts the received encrypted random numbers using the same secret key provided by remote client (200). In case of asymmetric encryption, (e.g. RSA encryption), a public key from remote client (200) is sent to RNG server(s) (300). RNG servers (300) use this public key to encrypt random numbers before they are sent back to remote client (200). Remote client (200) then uses the private key to decrypt received random numbers.

[0023] In some embodiments, more than one system can provide RNG server(s) (300) functionality. As described herein, RNG server(s) (300) functionality is provided by one or more systems and is explained using an example of a single RNG server (300). Examples of a transformation using encryption are also provided herein. Furthermore, symmetric encryption as a transformation method is similar to algorithmic transformation using a secret key, therefore, only the description of one method is provided herein as a description of one of the methods is sufficient for to both methods.

[0024] The sequence of events described in Figures 2 and 3 illustrate embodiments of a process that provide random numbers to a remote client application. In embodiments, a remote client (e.g., remote client (200)) may be, for example, a personal computer, a notebook, a netbook, a tablet computer, a mobile phone, a smart phone, a slot machine, an electronic poker table, a lottery terminal, a video lottery terminal, a POS device, a specialized gaming device, a server computer, a hand-held or laptop device, a multiprocessor system, a microprocessor-based system, a set top box, a programmable consumer electronics, a network PC, a minicomputer, a mainframe computer, or distributed computing environments that includes any of the above systems or devices, or any other device or application requiring random numbers. In one embodiment, each instance of remote client application may be treated as a separate remote client.

[0025] As shown in Figure 2, remote client (200) generates transformation information. For example, when mapping, remote client (200) may

generate a set of unique random numbers from a same or bigger range as requested random numbers (209). These remote client generated random numbers are used to map random numbers, received from game server (100), into original random numbers. In this example, symmetric encryption is used as a transformation, and thus, an encryption key may be generated. For asymmetric encryption, remote client's (200) public key may be used. For example, when using AES as a transformation algorithm, an AES key is generated. In embodiments, an encryption algorithm or a transformation type may be included in transformation information.

[0026] After transformation information is generated, transformation information is secured or encrypted (202) in such a way that only RNG server(s) (300) can understand it. In one embodiment, securing of the transformation information is accomplished using asymmetric encryption with a public key from RNG server(s) (300). In further embodiments, all RNG servers (300) may share a same private key. In still further embodiments, an external server may be used by RNG servers (300) to decode transformation information and a public key of the external server may be used by remote client (200) for encrypting transformation information. For the purpose of this disclosure, use of such an external server may be a part of RNG server(s) (300) functionality. In one embodiment, if remote client's (200) public key is used as transformation information, the public key of the remote client (200) does not need to be encrypted when sent to game server (100) and consequently to RNG server(s) (300).

[0027] Secured transformation (202) is sent with a request for one or more transformed random numbers (209) to game server (100). Game server (100) retrieves encrypted RNG state information (103) and thereafter stores and sends the retrieved encrypted RNG state information (103) with a request for transformed random numbers (109) and secured transformation (102) to at least one of the RNG servers (300). However, if the request for one or more transformed random numbers (209) is a first request for a set of random numbers (e.g., a game is just starting and the first cards are retrieved for a first player), game server (100) may send "dummy" RNG state information (103) or no RNG state information (103) at all. Once RNG server(s) (300) receive secured transformation (302/102/202) and encrypted RNG

state information (103/303), RNG server(s) (300) decrypt RNG state information (103/303) and retrieve remote client transformation (304). For example, if mapping is encrypted using an asymmetric algorithm or a symmetric algorithm, RNG server(s) (300) may decrypt the mapping using a corresponding algorithm and/or a corresponding key. Further, if a symmetric encryption is used for transformation, a secret key used for transformation is decrypted so that the decrypted secrete key can be used to encrypt random numbers sent back to game server (100).

[0028] In one embodiment, RNG server(s) (300) may generate requested random numbers (305). RNG server(s) (300) transform the generated requested random numbers (305) according to a transformation (306) used to create the transformed random numbers. For example, if mapping of random numbers is used, RNG server(s) (300) may map generated requested random numbers (305) according to mapping of remote client (200). Further, if encryption is used, RNG server(s) (300) may encrypt requested random numbers using an appropriate algorithm with a key provided by remote client (200), and, once this is done, RNG server(s) (300) may encrypt RNG state information (313) and transformation information (312) by using, for example, symmetric encryption and/or a key shared between at least one RNG server(s) (300). In one embodiment, a same key is used for encryption of both transformation information (312) and RNG state information (313). Encrypted RNG state information (313) and/or encrypted transformation information (312) are sent to game server (100) with requested transformed random numbers (317). Encrypted RNG state information (113/313) and encrypted transformation information (112/312 and/or 302/102/202) may be preserved on game server (100) so that they can be sent at a later time to RNG server(s) (300).

[0029] In some embodiments, encrypted transformation information (312) is not sent from RNG server(s) (300) to game server (100). For example, encrypted transformation information (312) is not sent from RNG server(s) (300) to game server (100) if encrypted transformation information (312) is provided to RNG server(s) (300) in a same format as originally provided from remote client (200). Transformed random number response (117/317) may be sent to remote client (200). Remote client (200) transforms back transformed random numbers (218), and if

mapping is used, remote client (200) maps back random numbers. Further, if encryption is used, remote client (200) decrypts random numbers using a previously generated key. In one embodiment, transformed random numbers are preserved by game server (100) to allow for a verification that the transformed random numbers were provided to remote client (200). In one embodiment, transformed random numbers are secured by authentication data from game server (100). The authentication data is provided to remote client (200) together with transformed random numbers. This data may be subsequently provided by remote client (200) together with random numbers and transformed random numbers to game server (100) for verification. In embodiments, authentication data may be, for example, HMAC, with a secret key known primarily by game server (100).

[0030] Together, with a request for transformed random numbers (209/109/309), player identifier (108/308) may be provided. In one embodiment, player identifier (108/308) may be used to track distribution of random numbers to specific remote clients by RNG server(s) (300). In an environment where game server (100) does not provide player or remote client identifier (108/308), RNG server(s) (300) may create unique identifiers/tags for player or remote client (200). A design of the RNG state information in such a way that player identifier (108/308) is related to random numbers issued is used to allow for verification of random number to be tied to a specific player. However, if player or remote client identifier (108/308) is provided by game server (100) for random number generation, a verification method may be simplified by verifying that correct transformation information is provided by game server (100), for example, that transformation information belongs to a specific player (e.g., a specific remote client (200)). In a situation where player identification is not provided by game server (100), RNG server(s) (300) may verify that a particular random number belong to a specific player. For example, RNG server(s) (300) may generate unique player tags that are stored with RNG state information and transformation information. These tags can then be matched when random number, RNG state information, and/or transformation information are provided for verification.

[0031] In embodiments, processing of a request for subsequent random numbers, as shown in Figure 3, is similar to the processes shown/described with respect to Figure 2. For example, as shown in Figure 3, there is not a need for a generation of a transformation of random numbers because transformation information is already generated and stored on game server (100) or another system. Subsequently, secured transformation, as shown in Figure 2, is neither sent nor processed by remote client (200). Transformation information for remote client (200) is retrieved from either encrypted transformation sent with a first response (Figure 2, data 112 and 212), or originally sent secured transformation (Figure 2, data 102). Transformation information (102) and encrypted RNG state information (103) may be sent to one of the one or more RNG server(s) (300). Once RNG server(s) (300) receives transformation information (102/202) and/or encrypted RNG state information (103/303), RNG server(s) (300) decrypts RNG state information and retrieves remote client transformation (304). However, if transformation information was encrypted using asymmetric algorithm or symmetric algorithm, RNG server(s) (300) decrypts transformation information using a corresponding algorithm and/or a corresponding key. RNG server(s) (300) may generate requested random numbers (305) and transform generated random numbers according to transformation (306) of remote client (200), and RNG server(s) (300) may encrypt RNG state information (313). Encrypted RNG state information (313) may be sent together with transformed random numbers (317) to game server (100). In one embodiment, encrypted or secured transformation information and encrypted RNG state information (113/313) are preserved on game server (100) so that they can be sent to RNG server(s) (300) at a later time. Transformed random numbers response (117/317) is sent to remote client (200). Remote client (200) may retrieve random numbers from transformed random numbers using previously generated transformation (218). For example, if mapping is used, remote client (200) decodes mapped random numbers, and if encryption is used for transformation, remote client (200) decrypts transformed random numbers. In one embodiment, a transformed random number may be preserved on game server (100). The transformed random number could then be used by game server (100) for verifying that the transformed random number is provided to remote client (200).

[0032] In one embodiment, with a single RNG sever (300), RNG state information and transformation information can be kept on a single RNG server (300). In embodiments, when information is kept on a single RNG server (300), the information does not have to be encrypted.

[0033] Remote client (200) may generate a new transformation for each activity session or game, or remote client (200) may generate a new transformation each time remote client (200) sends a request for transformed random numbers. In one embodiment, remote client (200) sends transformation information identification or a hash. However, if a transformation identification or hash on game server (100) is not the same transformation identification or hash on remote client (200), transformation information is requested from remote client (200). One of ordinary skill in the art guided by the teachings here in will appreciate that steps for transforming and transforming back random numbers to protect random numbers content from game server (100), and combining transformation information for different players together may be executed in a number of different orders.

[0034] In embodiments, secured or encrypted transformation and encrypted RNG state information are preserved on game server (100). This allows a request for random numbers combined with secured or encrypted transformation and encrypted RNG state information to be sent to any one of the RNG servers (300). Random numbers passed from RNG server(s) (300) to remote client (200) may be transformed using a transformation of an individual remote client (200) transformation. For example, when mapping is used, random numbers may be transformed using mapping of remote client (200), and when encryption is used, generated random numbers (e.g., generated random numbers combined with accidental or "dummy" data) may be encrypted using an encryption key provided by remote client (200). In one embodiment, game server (100) does not have access to the transformation information because the transformation information is encrypted, nor does game server (100) retrieve information about random numbers provided to different remote clients (200).

[0035] In embodiments, game server (100) verifies random numbers as revealed, discarded, or played by remote client (200). This can be executed in a few different ways by, for example, using RNG state information and random numbers to verify random numbers, or using transformation information, random numbers, transformed random numbers, and optionally RNG state information to verify random numbers.

[0036] In embodiments, as shown in Figures 2 and 3 at (108/308), respectively, game server (100) provides player identifiers to RNG server(s) (300). RNG server(s) (300) incorporates a relation between generated random numbers and player identifier, or an RNG server (300) generated player tag.

[0037] For a verification shown in Figure 4, one or more random numbers (251/151/351) may be provided together with player identification (158/358) and encrypted RNG state information (153/353) to RNG server(s) (300). In one embodiment, RNG server(s) 300 decrypts RNG state info (354) and verifies (357) that random numbers were previously generated for remote client (200). A verification result (361/161) is then sent back to game server (100). Optionally, RNG state information may be updated to include that a random number was already verified and new encrypted RNG state information (163) may be sent back to game server (100). However, in a case of a verification failure, both RNG server(s) (300) and game server (100) may log the failure and, for example, set an alarm.

[0038] In embodiments, as shown Figure 5, remote client (200) may send random numbers (251) together with transformed random numbers (252) to game server (100) to be stored. In one embodiment, game server (100) verifies whether transformed random numbers were provided to remote client (200). Game server (100) may then forward random numbers (151/251) and transformed random numbers (152/252) together with secured or encrypted transformation information and encrypted RNG state information (153) to RNG server(s) (300). RNG server(s) (300) decode transformation information and encrypted RNG state information (353/153), transform back (354) transformed random numbers (351/151/251), and verify (355) if these transformed back transformed random numbers match transformed random

numbers received (352/152/252). If mapping is used, random numbers may be mapped and checked against mapped random numbers. If encryption is used, encrypted random numbers may be decrypted and checked against random numbers. Verification result (361) is then sent back to game server (100). Any verification failure may then be logged (162/362) on both RNG server(s) (300) and game server (100).

[0039] In one embodiment, game server (100) stores transformed random numbers during random number generation process. This allows for a verification of transformed random numbers received from remote client (200). In one embodiment, game server (100) may add some security information to the transformed data before sending the transformed data to remote client (200). This security information could be later sent back to game server (100) from remote client (200) during a verification process. Security information may be, for example, an identifier of transformation information stored by a game server, an HMAC, or an encryption of the transformation information with a secret key known only to game server (100). Sending such additional information might be of particular importance if game server (100) is in a distributed environment.

[0040] While the verification process above illustrates particular embodiments, it is within the scope of the present disclosure to provide a different verification process. Thus, in embodiments, verification may be done in many ways by, for example, sending a random number or transformed random number, encrypted transformation, and encrypted RNG state information only for transformation on RNG server(s) (300). However, some steps may be optional or done in a different order/location. In one embodiment, game server (100) may keep information about transformed random numbers and verify if a transformed random number belongs to a specific player, and RNG server(s) (300) may verify only transformation of random numbers, or game server (100) may request transformation information for a specific number and verify correctness.

[0041] In one embodiment, during verification, RNG server(s) (300) log every transaction, and an identifier may be assigned to each transaction while

remote client (200) receives and stores a transaction identifier. Remote client (200) and game server (100) may use a transaction identifier to correlate a transaction to a random number. RNG server(s) (300) may then use a transaction identifier to locate a random number and verify that the random number was issued for a particular transaction.

[0042] In embodiments, a state of available random numbers may be logged for future audit. This state could be logged on game server (100), RNG server(s) (300), or another external system. In one embodiment, provided herein is an audit system that analyzes a sequence of random numbers and states of available random numbers to verify an integrity of generated random numbers. Furthermore, random numbers may be compared against game outcomes, for example, to validate how random numbers were used.

[0043] The methods described above allow for generations of any type of distributions. Random numbers may be generated using any random generation method, such as true RNG, PRNG (pseudorandom number generator), random numbers distributed from a previously generated file or set of random numbers, cryptographic random from external or internal source, auditable RNGs, such as described in U.S. Patent No. 6,934,846, which is incorporated herein by reference in its entirety, or any other method.

[0044] Verification may be executed in real time. For example, verification may be executed when a player's random number is presented, or in the alternative, verification can also be done at a later time.

[0045] Random numbers may be used for any kind of game, including but not limited to, dice games, card games, multiplayer games on the Internet or in a casino, multiplayer action games requiring random input, simulated racing games, slot machines, table games, and the like.

[0046] In one embodiment, RNG server logs, locally or on an external system, necessary received and necessary processed information for future audit. In one embodiment, for future audit, game server (100) stores information

provided by RNG server(s) (300) in both encrypted and transformed format. An audit system may retrieve, decrypt, and verify this information.

[0047] In a further embodiment, game server (100) and RNG server(s) (300) are integrated into a same system. In this embodiment, RNG server(s) (300) control access to RNG state information and transformation information, as well as control a general flow of information that follows the methods described above.

[0048] As illustrated herein, game server (100) is a single server. In a gaming environment, game server (100) might be replicated, work in distributed, n-plexed, a clustered environment, or be any other configuration.

[0049] It is within the scope of the present disclosure to generate individual mapping of random numbers for remote client (200) in RNG server(s) (300) and pass this mapping to remote client (200) using encryption in a way that game server (100) cannot decode it.

[0050] Referring next to Figure 6, an exemplary flow chart illustrates a process 600 for protecting random numbers. At 602, encrypted transformation information is received. At 604, the encrypted transformation information is decrypted. At 606, random numbers are generated. At 608, random numbers are transformed according to the decrypted transformation information. At 610, the transformed random numbers are sent to the game server (100). At 612, the transformed random numbers are secured, via the game server (100). At 614, the transformed random numbers are sent from the game server (100) to the remote client (200). In embodiments, the remote client (200) transforms the transformed random numbers to recreate the generated random numbers.

EXEMPLARY OPERATING ENVIRONMENT

[0051] A computing device or a computer as described in the present disclosure may have one or more processors or processing units and a system memory. The one or more processors execute computer-executable instructions for implementing aspects of the disclosure. In some embodiments, the one or more

processors are transformed into a special purpose microprocessor by executing computer-executable instructions or by otherwise being programmed. For example, the one or more processors may be programmed with instructions such as described herein with reference to Figure 6. The computer typically has at least some form of computer readable media. Computer readable media, which include both volatile and nonvolatile media, removable and non-removable media, may be any available medium that may be accessed by computer. By way of example and not limitation, computer readable media comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. For example, computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that may be used to store the desired information and that may be accessed by computer. Communication media typically embody computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and include any information delivery media. Those skilled in the art are familiar with the modulated data signal, which has one or more of its characteristics set or changed in such a manner as to encode information in the signal. Wired media, such as a wired network or direct-wired connection, and wireless media, such as acoustic, RF, infrared, and other wireless media, are examples of communication media. Combinations of any of the above are also included within the scope of computer readable media.

[0052] The system memory includes computer storage media in the form of removable and/or non-removable, volatile and/or nonvolatile memory. The computer may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer.

[0053] Embodiments of the invention are operational with numerous other general purpose or special purpose computing system environments or

configurations. The computing system environment is not intended to suggest any limitation as to the scope of use or functionality of any aspect of the invention. Moreover, the computing system environment should not be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment. Well known computing systems, environments, and/or configurations may be suitable for use with aspects of the invention.

[0054] Embodiments of the invention may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. The computer-executable instructions may be organized into one or more computer-executable components or modules. Generally, program modules include, but are not limited to, routines, programs, objects, components, and data structures that perform particular tasks or implement particular abstract data types. Aspects of the invention may be implemented with any number and organization of such components or modules. For example, aspects of the invention are not limited to the specific computer-executable instructions or the specific components or modules illustrated in the figures and described herein. Other embodiments of the invention may include different computer-executable instructions or components having more or less functionality than illustrated and described herein. Aspects of the invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0055] In operation, a computer executes computer-executable instructions embodied in one or more computer-executable components stored on one or more computer-readable media to implement aspects of the invention described and/or illustrated herein.

[0056] Having described aspects of the invention in detail, it will be apparent that modifications and variations are possible without departing from the

scope of aspects of the invention as defined in the appended claims. As various changes could be made in the above constructions, products, and methods without departing from the scope of aspects of the invention, it is intended that all matter contained in the above description and shown in the accompanying drawings shall be interpreted as illustrative and not in a limiting sense.

WHAT IS CLAIMED IS:

1. A method for protecting random numbers, the method comprising:

receiving encrypted transformation information;

decrypting the encrypted transformation information;

generating random numbers;

transforming random numbers according to the transformation information;

sending the transformed random numbers to a game server;

securing, via the game server, the transformed random numbers;

sending the transformed random numbers from the game server to a remote client, wherein the remote client transforms the transformed random numbers to recreate the generated random numbers.

2. A method in accordance with claim 1, further comprising:

generating transformation information;

encrypting the transformation information using an asymmetric algorithm; and

sending the encrypted transformation information.

3. A method in accordance with claim 1, wherein transforming random numbers according to the decrypted transformation information comprises mapping, standard encryption, or using a private algorithm with a secret key.

4. A method in accordance with claim 1, further comprising retrieving encrypted transformation information by at least one or more of the following: using a separate system providing decryption service, and decrypting information with a private key used by one or more random number generation servers.

5. A method in accordance with claim 1, further comprising:

encrypting transformation information; and

safeguarding transformation information on a same system or another system for one or more remote clients.

6. A method in accordance with claim 5, further comprising adding a player identifier or player tag with transformation information.

7. A method in accordance with claim 5, further comprising using a same encryption key by one or more random number generation servers for encryption.

8. A method in accordance with claim 1, further comprising:

preserving, in an encrypted format, random number generator state information on a random number generation server, the game server, or another system; and

passing random number generator state between the game server and one or more random number generation servers

9. A method in accordance with claim 8, further comprising preserving random number generator state information about specific random numbers issued to specific remote clients.

10. A method in accordance with claim 8, further comprising preserving, in an unencrypted format, random number generator state information and transformation information on a random number generation server or another trusted system.

11. A method in accordance with claim 1, further comprising providing, by the remote client, one or more random numbers and one or more transformed random numbers for verifying random numbers.

12. A method in accordance with claim 1, further comprising verifying, via a game server, random numbers generated by sending random numbers and/or

transformed random numbers and transformation information to a random number generation server or a third party system for verification.

13. A method in accordance with claim 1, further comprising storing transformed random numbers for future verification.

14. A method in accordance with claim 1, further comprising one of storing encrypted random number generator state information for future generation and/or verification of random numbers, or sending transformed random numbers with authentication information to the remote client.

15. A method in accordance with claim 1, further comprising using a separate system for verifying random numbers, wherein the separate system is a random number generation server or another third party system, and wherein the separate system is configured to do one or more of the following:

use encrypted transformation information;

use encrypted random number generator state information and at least one random number; and

verify that a specific random number belongs to a particular remote client.

16. A method in accordance with claim 1, further comprising verifying random numbers, via an audit system, by one or more of the following: analyzing consecutively issued random numbers for a game, verifying if random numbers are properly issued to particular remote clients, and verifying if random numbers are consecutively issued for a game.

17. A method in accordance with claim 1, further comprising verifying issued random numbers by providing random numbers and a transaction identifier to a random number generation server.

18. A method in accordance with claim 1, further comprising using random numbers for a card game or a dice game.

19. A method in accordance with claim 1, connecting the gaming or gambling devices to the game server, and wherein the remote client is part of the gaming or gambling devices.

20. A method in accordance to claim 1, further comprising, sending, from the remote client to the game server, random numbers and one or more of the following: security information, and transformation information.

21. A method in accordance with claim 1, further comprising:

generating transformation information in a random number generation server;

encrypting transformation information using either an asymmetric or a symmetric algorithm; and

sending encrypted transformation information by the random number generation server to the remote client via a gaming system.

22. A method for protecting random numbers, the method comprising:

receiving a public key of a remote client;

generating random numbers;

encrypting the random numbers with the public key; and

sending the encrypted random numbers to a game server, wherein the game server secures the encrypted random number and sends the secured encrypted random numbers to a remote client for decryption.

23. A gaming system for playing card games, the system comprising:

a game server; and

one or more processors programmed to:

receive transformed random numbers, the transformed random numbers transformed by a random number generation server according to transformation information received from a remote client;

secure the transformation information; and

verify random numbers.

24. A gaming system in accordance with claim 23, wherein the one or more processors are programmed to verify random numbers by sending the random numbers to the random number generation server for verification, or by sending the random numbers to a separate third party for verification.

FIG. 1

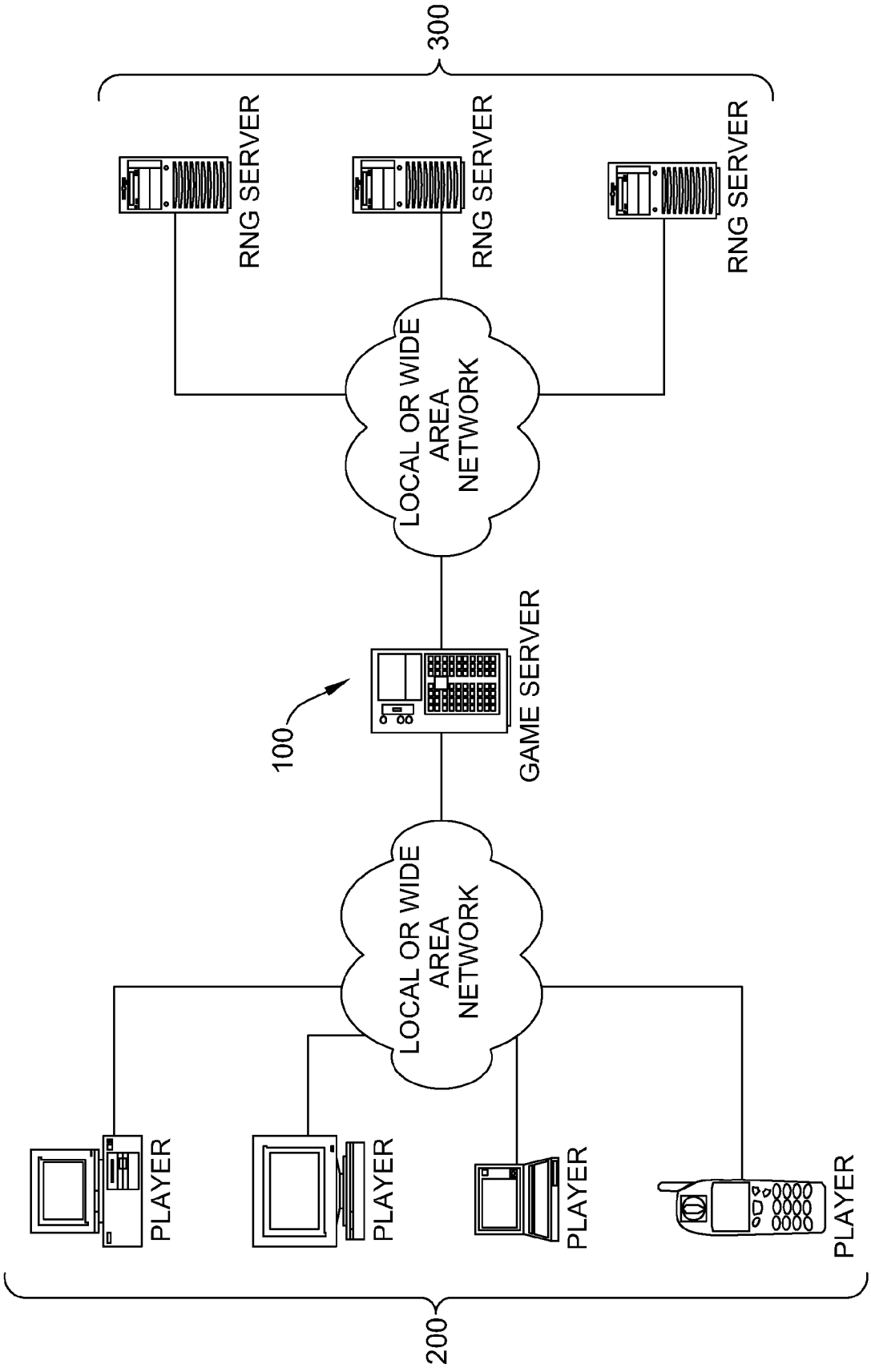

300
RNG SERVER
RNG SERVER
RNG SERVER
LOCAL OR WIDE AREA NETWORK
100
GAME SERVER
LOCAL OR WIDE AREA NETWORK
PLAYER
PLAYER
PLAYER
PLAYER
200

FIG. 2

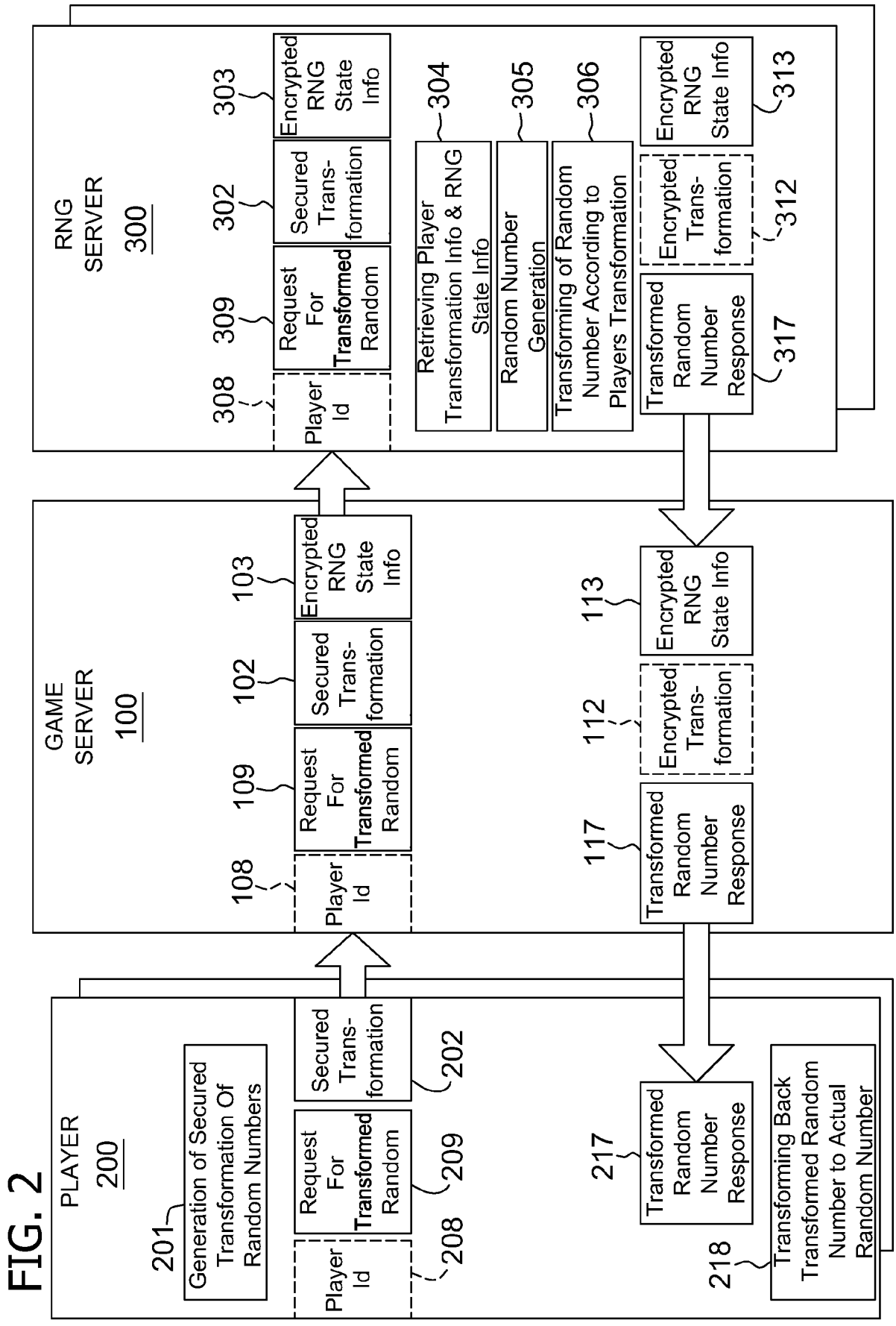
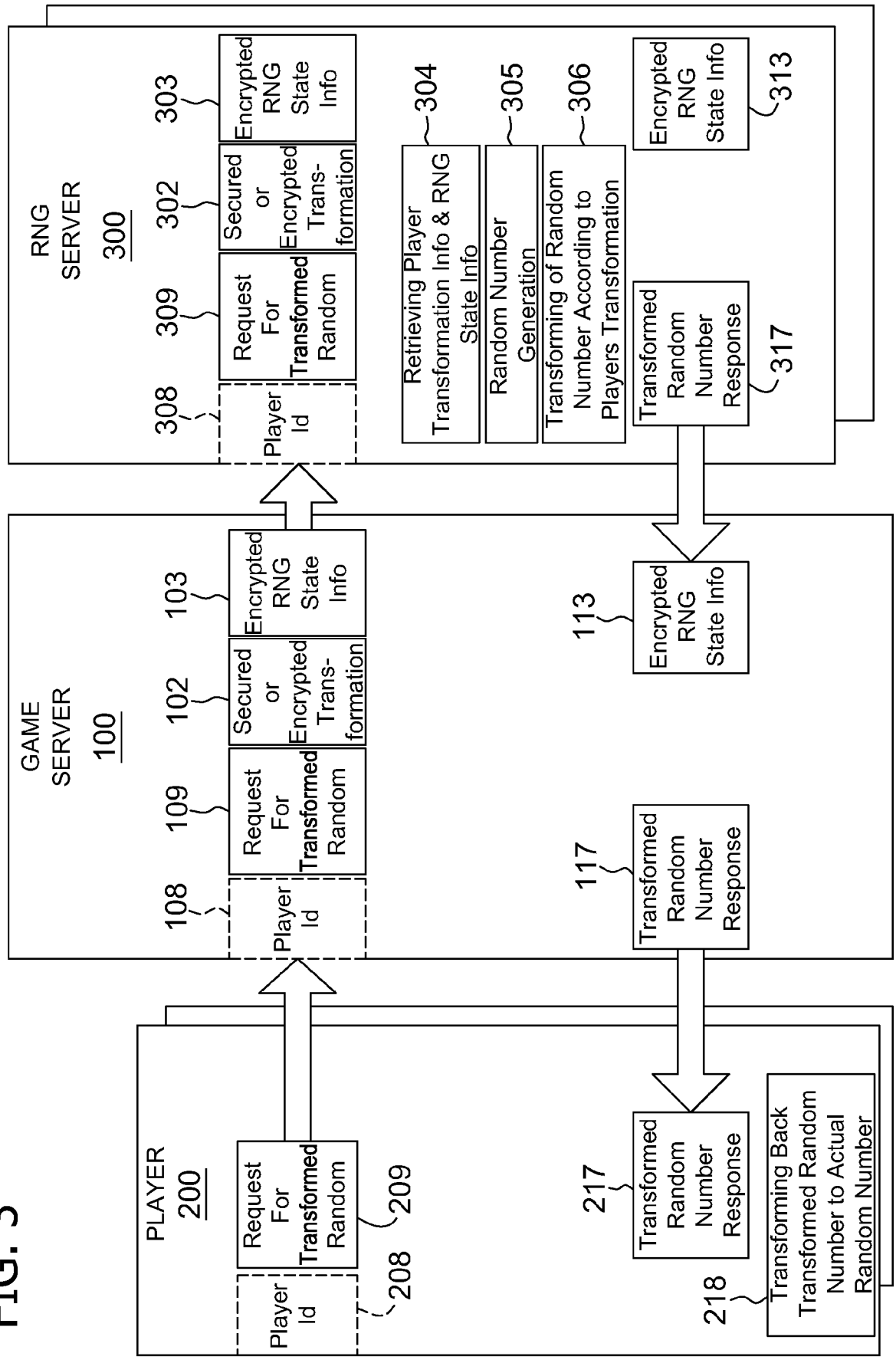

PLAYER
200
201
Generation of Secured Transformation Of Random Numbers
Player Id
208
Request For Transformed Random
209
Secured Trans-formation
202
GAME SERVER
100
Player Id
108
Request For Transformed Random
109
Secured Trans-formation
102
Encrypted RNG State Info
103
RNG SERVER
300
Player Id
308
Request For Transformed Random
309
Secured Trans-formation
302
Encrypted RNG State Info
303
Retrieving Player Transformation Info & RNG State Info
304
Random Number Generation
305
Transforming of Random Number According to Players Transformation
306
Transformed Random Number Response
317
Encrypted Trans-formation
312
Encrypted RNG State Info
313
Transformed Random Number Response
217
Encrypted Trans-formation
112
Encrypted RNG State Info
113
Transformed Random Number Response
117
218
Transforming Back Transformed Random Number to Actual Random Number

FIG. 3


PLAYER
200
Player Id
208
Request For Transformed Random
209
Transformed Random Number Response
217
Transforming Back Transformed Random Number to Actual Random Number
218
GAME SERVER
100
Player Id
108
Request For Transformed Random
109
Secured or Encrypted Trans-formation
102
Encrypted RNG State Info
103
Encrypted RNG State Info
113
Transformed Random Number Response
117
RNG SERVER
300
Player Id
308
Request For Transformed Random
309
Secured or Encrypted Trans-formation
302
Encrypted RNG State Info
303
Retrieving Player Transformation Info & RNG State Info
304
Random Number Generation
305
Transforming of Random Number According to Players Transformation
306
Transformed Random Number Response
317
Encrypted RNG State Info
313

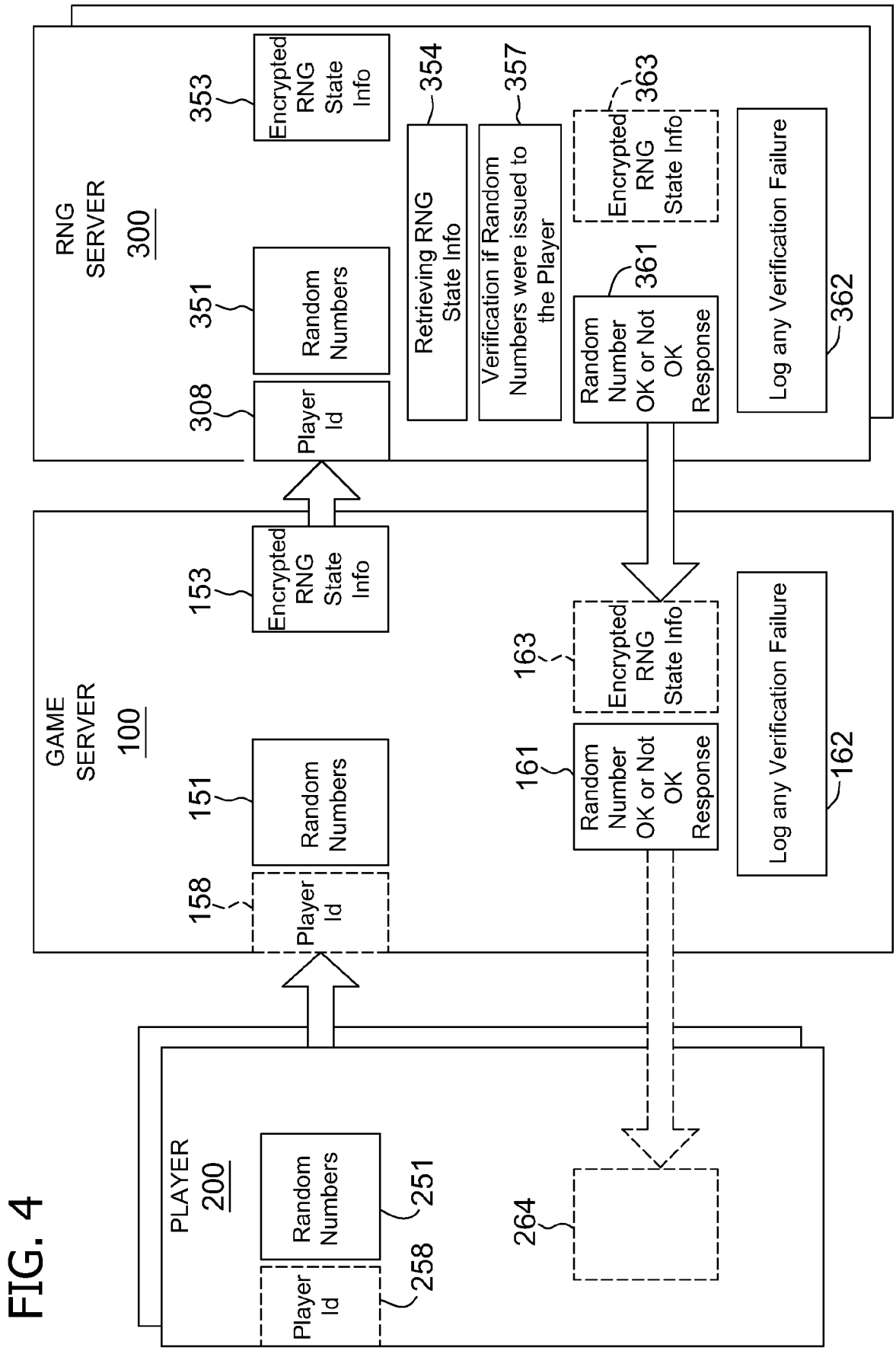

RNG SERVER 300
353
Encrypted RNG State Info
354
Retrieving RNG State Info
357
Verification if Random Numbers were issued to the Player
363
Encrypted RNG State Info
351
Random Numbers
361
Random Number OK or Not OK Response
362
Log any Verification Failure
308
Player Id
GAME SERVER 100
153
Encrypted RNG State Info
163
Encrypted RNG State Info
151
Random Numbers
161
Random Number OK or Not OK Response
162
Log any Verification Failure
158
Player Id
PLAYER 200
Random Numbers
251
Player Id
258
264

FIG. 4

FIG. 5

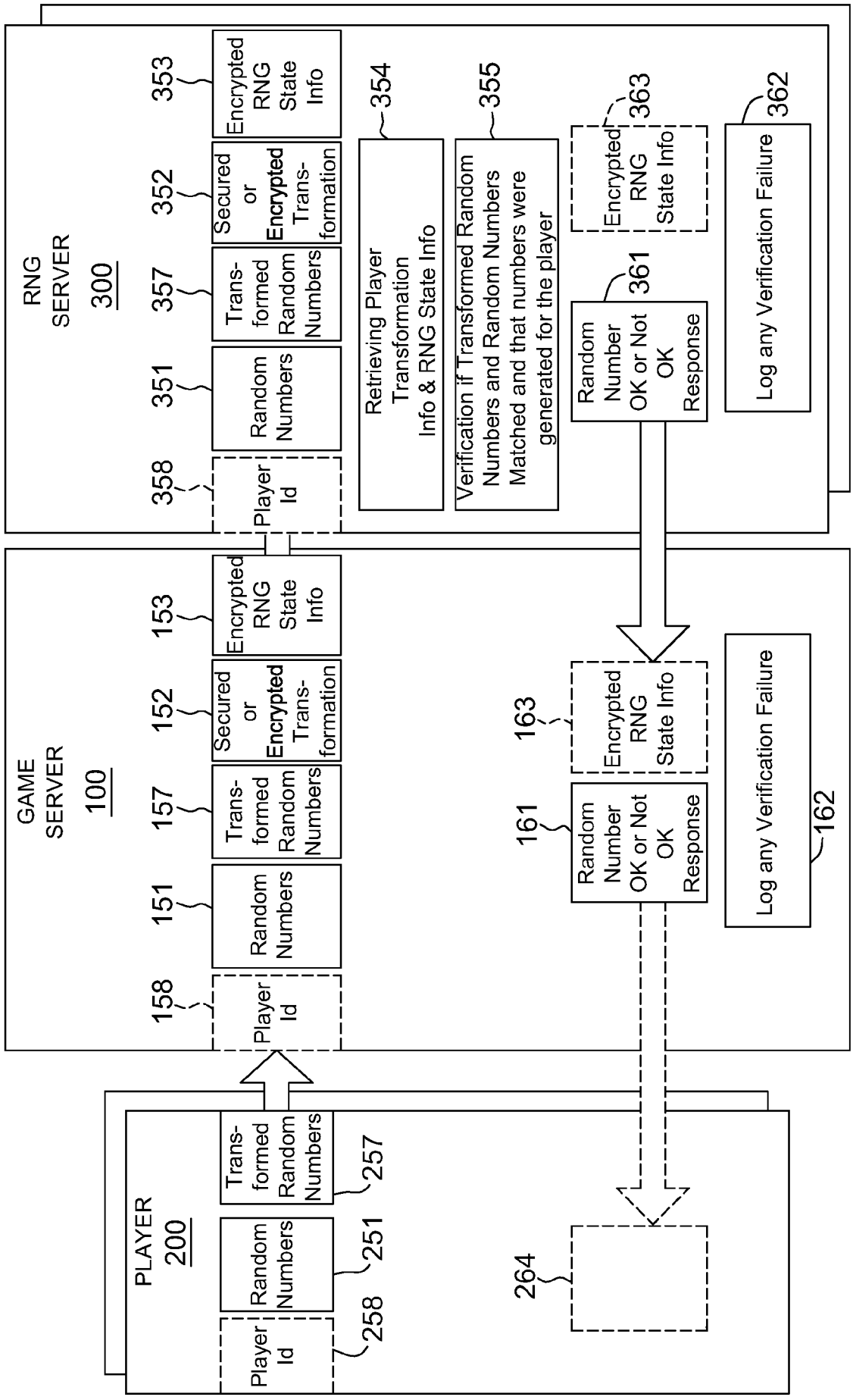
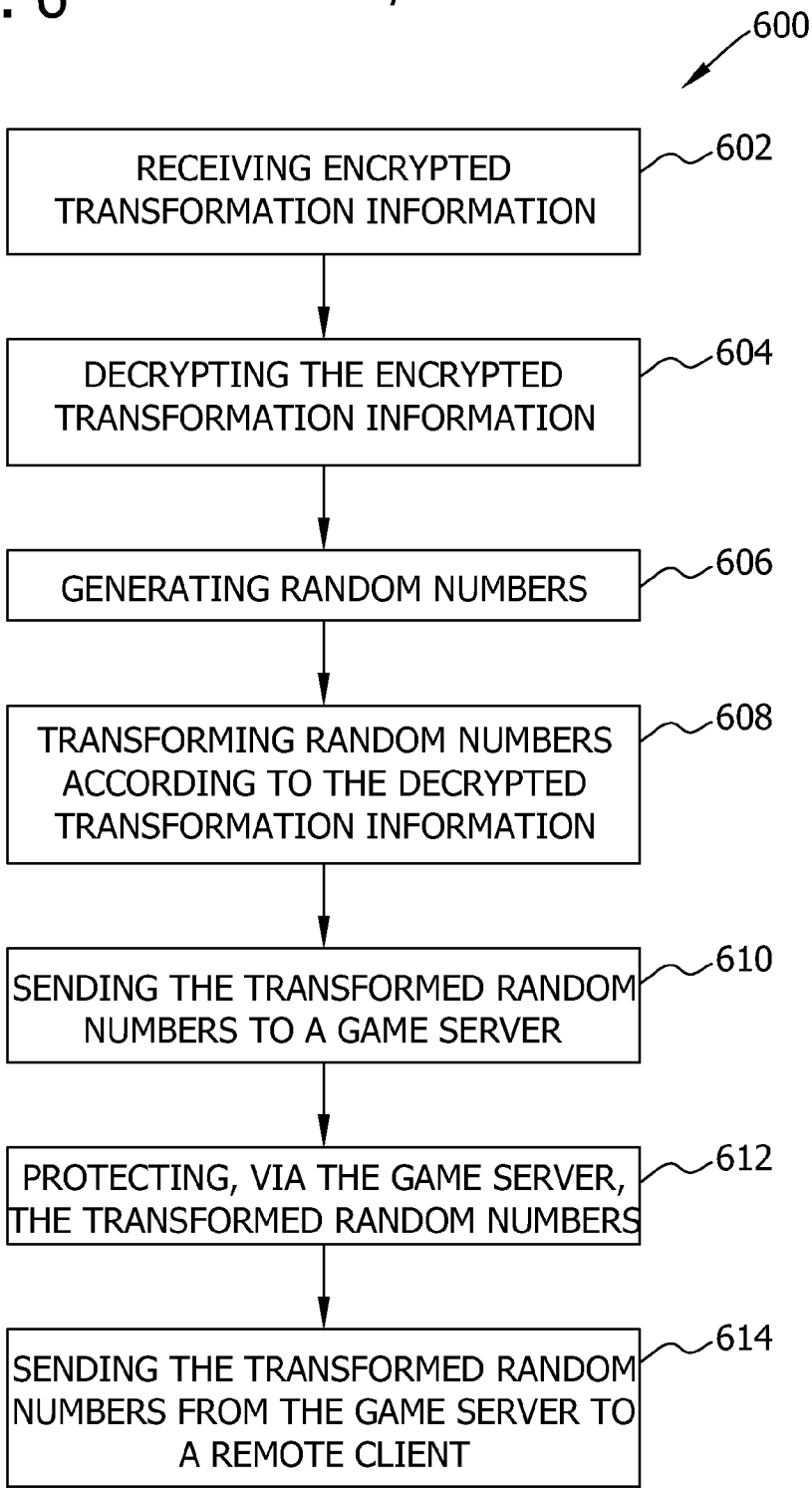

PLAYER
200
Player Id
Random Numbers
Trans-formed Random Numbers
251
257
258
264
GAME SERVER
100
158
Player Id
151
Random Numbers
157
Trans-formed Random Numbers
152
Secured or Encrypted Trans-formation
153
Encrypted RNG State Info
161
Random Number OK or Not OK Response
163
Encrypted RNG State Info
162
Log any Verification Failure
RNG SERVER
300
358
Player Id
351
Random Numbers
357
Trans-formed Random Numbers
352
Secured or Encrypted Trans-formation
353
Encrypted RNG State Info
354
Retrieving Player Transformation Info & RNG State Info
355
Verification if Transformed Random Numbers and Random Numbers Matched and that numbers were generated for the player
361
Random Number OK or Not OK Response
363
Encrypted RNG State Info
362
Log any Verification Failure

FIG. 6


600
RECEIVING ENCRYPTED TRANSFORMATION INFORMATION
602
DECRYPTING THE ENCRYPTED TRANSFORMATION INFORMATION
604
GENERATING RANDOM NUMBERS
606
TRANSFORMING RANDOM NUMBERS ACCORDING TO THE DECRYPTED TRANSFORMATION INFORMATION
608
SENDING THE TRANSFORMED RANDOM NUMBERS TO A GAME SERVER
610
PROTECTING, VIA THE GAME SERVER, THE TRANSFORMED RANDOM NUMBERS
612
SENDING THE TRANSFORMED RANDOM NUMBERS FROM THE GAME SERVER TO A REMOTE CLIENT
614