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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
04 January 2018 (04.01.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/000317 A1

(51) International Patent Classification:
H04L 9/14 (2006.01)

(21) International Application Number:
PCT/CN2016/087876

(22) International Filing Date:
30 June 2016 (30.06.2016)

(25) Filing Language: English

(26) Publication Language: English

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

(71) Applicant *(for LC only)*: **NOKIA TECHNOLOGIES (BEIJING) CO., LTD.** [CN/CN]; Unit 18, Room 1001, Level 10, East Tower, World Financial Center, No. 1, East Third Ring Middle Road, Chaoyang District, Beijing 100020 (CN).

(72) Inventors: **DING, Wenxiu**; POX 091, Xidian University, No. 2 South Taibai Road, Xi'an, Shaanxi 710071 (CN). **YAN, Zheng**; No. 403, Building 69, Xidian University, No. 2 South Taibai Road, Xi'an, Shaanxi 710071 (CN).

(74) Agent: **KING & WOOD MALLESONS**; 20th Floor, East Tower, World Financial Centre, No. 1 Dongsanhuan Zhonglu, Chaoyang District, Beijing 100020 (CN).

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

WO 2018/000317 A1

(54) Title: SECURE DATA PROCESSING

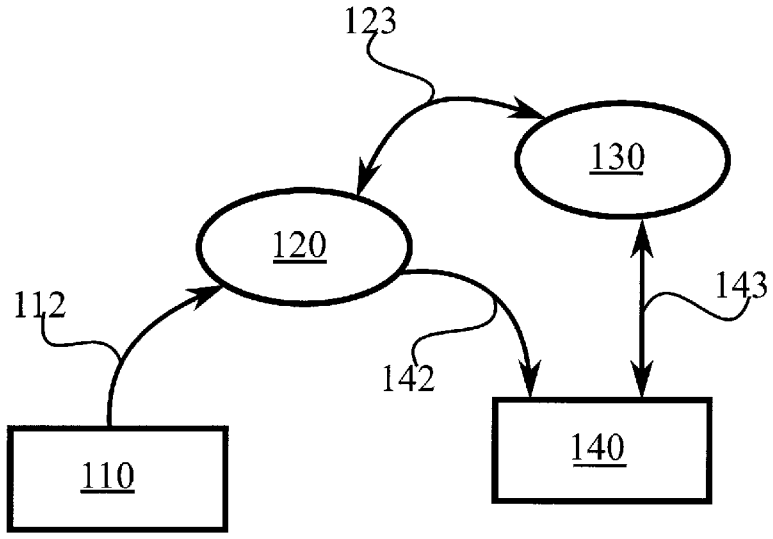

FIGURE 1

(57) Abstract: According to an example aspect of the present invention, there is provided an apparatus comprising at least one processing core, at least one memory including computer program code, the at least one memory and the computer program code being configured to, with the at least one processing core, cause the apparatus at least to receive, from a data provider, a first ciphertext, perform a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier, obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and provide the second ciphertext to a first computation party.

[Continued on next page]

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

SECURE DATA PROCESSING

FIELD

[0001] The present invention relates to processing information in encrypted form, for example in a cloud service provision environment.

BACKGROUND

[0002] Cloud computing services provide off-site opportunities for individuals and corporations. For example, cloud storage service enables off-site storage of data sets in a flexible manner in a data centre, reducing the need for users of the cloud service to obtain their own storage hardware, for example for archiving purposes.

[0003] A further example of a cloud service is a cloud processing service, wherein a user is given access to processor resources at a computer or computing grid. This may be useful, for example where a user needs access to high-capacity computing intermittently, and obtaining actual high-capacity computing hardware would be wasteful as the hardware would mostly be unused, since the need is only intermittent.

[0004] Consumers may use cloud services to back up their data, for example during operating system updates of their devices, such as computers, smartphones and laptops. Some smart devices are configured to automatically upload images captured by users to a cloud storage service.

[0005] While useful, cloud services present high risk to users. Personal information may accidentally, or purposefully, be stored on a cloud storage service. Such personal information may become vulnerable to theft, unauthorised modification or eavesdropping either during transit to or from the cloud storage service, or while in the cloud storage service. The cloud service provider may be untrusted or only partially trusted. Furthermore, the cloud service may be distributed between several data centres, and customer data may be communicated between such data centres to balance load between the centres. Such communication presents additional risk of eavesdropping.

[0006] As it relates to cloud processing services, risks are also present. Where a user seeks to analyse medical patient data, for example, not only the data itself but also its processing is highly confidential by its intrinsic nature. Similar considerations apply to other confidential data that may be processed, such as corporate, personal or military data. For example, analysing stealth-defeating radar may comprise complex processing of plural radar signals and their combinations. Clearly, such signal processing is secret due to its nature.

SUMMARY OF THE INVENTION

[0007] The invention is defined by the features of the independent claims. Some specific embodiments are defined in the dependent claims.

[0008] According to a first aspect of the present invention, there is provided an apparatus comprising at least one processing core, at least one memory including computer program code, the at least one memory and the computer program code being configured to, with the at least one processing core, cause the apparatus at least to receive, from a data provider, a first ciphertext, perform a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier, obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and provide the second ciphertext to a first computation party.

[0009] Various embodiments of the first aspect may comprise at least one feature from the following bulleted list:

- the apparatus is configured to obtain the computation identifier from the first computation party
- the apparatus is further configured to participate in negotiating a shared secret with the first computation party

- the cryptographic re-encryption operation is performed in dependence of the computation identifier
- the second ciphertext is not decryptable solely by a secret key of the first computation party
- the apparatus is further configured to obtain a key pair comprising a public key of the apparatus and a secret key of the apparatus
- the computation identifier identifies at least one of the following processes: addition, subtraction, multiplication, sign acquisition, comparison, equivalence test and variance
- the apparatus is further configured to obtain a third ciphertext from the first ciphertext, to provide the third ciphertext to a second computation party, and to obtain a fourth ciphertext from responses received in the apparatus from the first computation party and the second computation party, and to obtain an encrypted result of a computation process identified by the computation identifier.

[0010] According to a second aspect of the present invention, there is provided an apparatus comprising at least one processing core, at least one memory including computer program code, the at least one memory and the computer program code being configured to, with the at least one processing core, cause the apparatus at least to determine, based on a message from a data requester, a computation identifier, transmit a request to a data service provider, the request comprising the computation identifier, receive, from the data service provider, a first ciphertext, obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and provide the second ciphertext to the data requester as a response to the message.

[0011] Various embodiments of the second aspect may comprise at least one feature from the following bulleted list:

- the apparatus is further configured to check an access policy before providing the request to the data service provider
- the apparatus is further configured to participate in negotiating a shared secret with the data service provider
- the negotiating comprises a Diffie-Hellman negotiation
- the apparatus is configured to perform the cryptographic re-encryption operation in dependence of the computation identifier

- the computation identifier identifies one of the following computation processes: addition, subtraction, multiplication, sign acquisition, comparison, equivalence test and variance.

[0012] According to a third aspect of the present invention, there is provided a method comprising receiving, from a data provider, a first ciphertext, performing a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier, obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and providing the second ciphertext to a first computation party.

[0013] Various embodiments of the third aspect may comprise at least one feature corresponding to a feature from the preceding bulleted list laid out in connection with the first aspect.

[0014] According to a fourth aspect of the present invention, there is provided a method comprising determining, based on a message from a data requester, a computation identifier, transmitting a request to a data service provider, the request comprising the computation identifier, receiving, from the data service provider, a first ciphertext, obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and providing the second ciphertext to the data requester as a response to the message.

[0015] Various embodiments of the fourth aspect may comprise at least one feature corresponding to a feature from the preceding bulleted list laid out in connection with the second aspect.

[0016] According to a fifth aspect of the present invention, there is provided a system comprising an apparatus according to the first aspect, an apparatus according to the second aspect, a data requester and a data provider.

[0017] According to a sixth aspect of the present invention, there is provided an apparatus comprising means for receiving, from a data provider, a first ciphertext, means for performing a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier,

means for obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and means for providing the second ciphertext to a first computation party.

[0018] According to a seventh aspect of the present invention, there is provided an apparatus comprising means for obtaining a key pair comprising a public key of an apparatus and a secret key of the apparatus, means for determining, based on a message from a data requester, a computation identifier, means for transmitting a request to a data service provider, the request comprising the computation identifier and a public key of the data requester, means for receiving, from the data service provider, a first ciphertext, means for obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and means for providing the second ciphertext to the data requester as a response to the message.

[0019] According to an eighth aspect of the present invention, there is provided a non-transitory computer readable medium having stored thereon a set of computer readable instructions that, when executed by at least one processor, cause an apparatus to at least receive, from a data provider, a first ciphertext, perform a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier, obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and provide the second ciphertext to a first computation party.

[0020] According to a ninth aspect of the present invention, there is provided a non-transitory computer readable medium having stored thereon a set of computer readable instructions that, when executed by at least one processor, cause an apparatus to at least obtain a key pair comprising a public key of an apparatus and a secret key of the apparatus, determine, based on a message from a data requester, a computation identifier, transmit a request to a data service provider, the request comprising the computation identifier and a public key of the data requester, receive, from the data service provider, a first ciphertext, obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and provide the second ciphertext to the data requester as a response to the message.

[0021] According to a tenth aspect of the present invention, there is provided a computer program configured to cause a method in accordance with at least one of the third and fourth aspects to be performed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022] FIGURE 1 illustrates an example system in accordance with at least some embodiments of the present invention;

[0023] FIGURE 2 illustrates an example system in accordance with at least some embodiments of the present invention.

[0024] FIGURE 3 illustrates an example apparatus capable of supporting at least some embodiments of the present invention;

[0025] FIGURE 4 illustrates signalling in accordance with at least some embodiments of the present invention;

[0026] FIGURE 5 is a flow graph of a method in accordance with at least some embodiments of the present invention, and

[0027] FIGURE 6 is a flow graph of a method in accordance with at least some embodiments of the present invention.

EMBODIMENTS

[0028] Confidential processing of data in a cloud service may be obtained by dividing processing into parts, the processing being conducted on encrypted data, which is known as ciphertext. A two-level decryption process is used with two service provision entities, a data service provider and a computation party, which co-operate to jointly perform secure processing of data and deliver processed data in encrypted form to a data requesting party. The data requesting party may be a same party as the one that provided

the data, that it, a data provider. At least some embodiments of the invention operate using homomorphic re-encryption.

[0029] The data service provider may receive a computation identifier and perform a mathematical manipulation of a first ciphertext, to thereby modify a plaintext underlying the first ciphertext. In other words, the data service provider may perform a computation on plaintext of the first ciphertext, without decrypting the first ciphertext, by mathematically manipulating the first ciphertext. The manipulation, and/or corresponding modification of the plaintext, is identified by the computation identifier. Subsequently, the data service provider may re-encrypt the manipulated first ciphertext to obtain a second ciphertext. The re-encrypting may comprise use of a secret key of the data service provider and a public key of a data requester, for example. The re-encrypting may be performed in dependence of the computation identifier. Re-encrypting may comprise at least partial decryption followed by encryption.

[0030] FIGURE 1 illustrates an example system in accordance with at least some embodiments of the present invention. The system comprises data service provider 120, which may comprise a cloud data storage data centre or cloud data centre system, for example. Data service provider 120 may also be a cloud processing service provider. A cloud data centre system may comprise a plurality of data centres, with load balancing arranged in a suitable manner between individual data centres comprised in the plurality. In general, data service provider 120 may be configured to store data and provide some computation services.

[0031] The system of FIGURE 1 further comprises at least one computation party 130. Computation party 130 may comprise a processing-enabled computing entity, such as, for example, a data centre, data centre system, server, server farm or indeed an individual networked computer such as a desktop or a laptop. In general, computation party 130 may be configured to provide data computation services and/or data access control for its users. In the model described herein, there may exist several computation parties, CP, 130 that are operated by different entities, such as medical institutions, schools, and/or banks. Different CPs may deal with different kinds of data, for example health-related data, student records and financial information. Hence, a user may freely choose a CP he trusts for service consumption. This may enhance user security.

[0032] The system of FIGURE 1 further comprises at least one data provider 110. Data provider 110 may comprise a data owner, such as, for example, a consumer, corporation or government entity, for example. For example, data provider 110 may generate the data. Data may be provided by an X-ray device or body scanner where data provider 110 is a medical entity, such as a clinic or hospital. Data may be generated in an industrial process or a design tool where data provider 110 is a corporate entity, such as a manufacturer or engineering company. Data may be generated in a radar or flight control facility where data provider 110 is a government entity, such as a military or aviation authority. Data provider 110 may be configured to provide data in encrypted form to data service provider 120.

[0033] The system of FIGURE 1 further comprises data requester 140. Data requester 140 may comprise an entity authorised by data provider 110 to access, at least partly, data owned and/or generated by data provider 110. Data requester 140 may need the data of data provider 110 in a processed form. Data requester 140 may be the same entity as data provider 110.

[0034] Overall, data provider 110, data service provider 120, computation party 130 and data requester 140 may be seen as roles or functions that may be assumed and performed by different kinds of entities. As indicated above, data provider 110 and data requester 140 may be one and the same. On the other hand, data service provider 120 and computation party 130 are not physically the same entity. In detail, data service provider 120 need not be trusted by data provider 110, while computation party 130 may be trusted by data provider 110.

[0035] Networked connections interconnect the entities described above to each other. In detail, connection 112 enables data provider 110 to transmit ciphertext to data service provider 120. Connection 142 enables communication between data requester 140 and data service provider 120. Connection 123 enables communication between computation party 130 and data service provider 120. Finally, connection 143 enables communication between data requester 140 and computation party 130. The connections may be wired or, at least partly, wireless, connections, where applicable.

[0036] FIGURE 2 illustrates an example system in accordance with at least some embodiments of the present invention. Like numbering denotes like structure as in FIGURE 1. FIGURE 2 corresponds to the case, where data requester 140 is the same entity

as data provider 110. For example, this is the case where a data owner requests a cloud data processing result of his own data.

[0037] In the systems illustrated in FIGURE 1 and FIGURE 2, all entities are assumed to be curious but honest. That is, they are curious about others' data but act honestly by following the design of system protocols. In addition, the DSP 120 and the CPs 130 would be unlikely to collude with each other due to interest conflicts. Moreover, any collusion would decrease user trust in the CP 130, which leads to the loss of its users.

[0038] Three challenges are addressed by various embodiments of the present invention.

[0039] Firstly, existing ciphertext processing schemes cannot flexibly support multiparty access. Many schemes are designed based on a specified aggregator, wherein only this aggregator can access the aggregated result. Such schemes cannot support genuine multiparty access. The data provided by users cannot be used for data analysis by other authorized requesters. In some situations, data should be observed and collected all the time for potential use without knowing a concrete data access requester. For example, medical and clinical research can benefit greatly from statistics of patients. More than one party could be interested in requesting encrypted processing results after data collection and process. Hence, such an application scenario calls for a privacy-preserving data processing scheme for multiple unspecified requesters. However, existing work cannot support this demand in a flexibly and efficient way.

[0040] Secondly, it is desired that ciphertext processing results should be only accessed by authorized requesters. A data processing party, such as a cloud service provider, should not be able to access the results if it cannot be fully trusted by the data owner. The problem of privacy-preserving data aggregation with a distrusted aggregator has been studied, but such studies only describe systems that allow the aggregator to access the final aggregated results. Such schemes cannot satisfy the practical security requirement.

[0041] Thirdly, most existing research focuses on encrypted data aggregation and multiplication, which cannot support various computation operations over the same encrypted outsourced data. Aggregation and multiplication of encrypted data are not sufficient to satisfy emerging demands on data analytics in our daily life. Fully homomorphic encryption could support various computations of encrypted data, but it

cannot flexibly control multiparty access on the processed data in an encrypted form because the finally processed data can only be decrypted with a corresponding secret key.

[0042] To solve the above-described three problems, a re-encryption scheme is proposed, which can flexibly support access control on encrypted-data computation results with two-level decryption. Furthermore, the proposed re-encryption scheme is applied to realize a number of operations over ciphertexts including addition, subtraction, multiplication, sign acquisition, comparison, equivalent test, and variance, which may support various applications that request processing of encrypted data and/or analytics thereof.

[0043] We consider a scenario with two types of independent service providers that do not collude with each other. One is data service provider, DSP, 120 that is responsible for data storage and partial computations. The other is computation party, CP, 130 that is in charge of data access control and additional computations. DSP 120 collects and stores the data in an encrypted form from a number of data providers, DP, 110. When a data requester, DR, 140, requires a data processing result, CP 130 may first check an access right of DR 140. If DR 140 is eligible to access the data, CP 130 contacts DSP 120 to further process the ciphertext with a re-encryption process for the DR 140, and then CP 130 may provide the re-encryption result to the authorized DR 140 for decryption. In addition, the described scheme supports multiple CPs 130 served by different companies for distributed data processing and access control. In general, DSP 120 and CP 130 together produce the ciphertext, with contents of the ciphertext processed as requested by DR 140, such that DSP 120 does not obtain access to the contents of the ciphertext despite DSP 120 participating in performing the requested processing.

[0044] Specifically, the contributions of this invention can be summarized as the following three elements:

[0045] Firstly, a new cryptographic primitive is described, which uses two service providers, DSP 120 and CP 130, to manage encrypted data and realizes re-encryption over homomorphic encryption. The primitive enables processing and analysis of ciphertext. Only authorized users can access the encrypted data processing result in a secure way.

[0046] Secondly, seven basic operations on encrypted data based on the primitive are described: addition, subtraction, multiplication, sign acquisition, comparison, equivalent test, and variance, which can satisfy many data processing demands.

[0047] Thirdly, to enhance the flexibility and security of the proposed schemes, we apply multiple CPs to take in charge of the data from their own customers and design computing operations over ciphertexts belonging to multiple CPs.

[0048] Paillier's cryptosystem, as described in Paillier: "Public key cryptosystems based on composite degree residuosity classes" (Advances in cryptology, EUROCRYPT 1999, pp. 223-238) is one of the most important additive homomorphic encryption systems. Suppose we have N encrypted data under same key pk, which can be presented as $[m_i]_{pk}$ $(i = 1, 2, \dots, N)$. The additive homomorphic encryption satisfies the following equation:

[0049] $$D_{sk}\left(\prod_{i=1}^{N}[m_i]_{pk}\right) = \sum_{i=1}^{N} m_i$$

[0050] where $D_{sk}()$ is the corresponding homomorphic decryption algorithm with secret key sk. Reference is made to Paillier (see above), ElGamal "A public key cryptosystem and a signature scheme based on discrete algorithms" (Advances in cryptology, Springer, 1985, pp 10-18.) and Bresson et al. "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications", (Advances in cryptology, ASIACRYPT 2003, Springer, pp. 37-54).

[0051] For easy presentation, we use the acronym EDD to present the mechanism proposed by E. Bresson, D. Catalano and D. Pointcheval: "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications" (Advances in Cryptology, ASIACRYPT, 2003, pp. 37 – 54, Springer, 2003), which is a variant of Cramper-Shoup "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption" (Advances in cryptology, EUROCRYPT 2002, pp. 45-64).

[0052] Let g and h be two elements of maximal order in $\mathbb{G}$. Note that, if h is computed as g^x, where $x \in_R [1, \lambda(n^2)]$, then x is coprime with ord($\mathbb{G}$) with high probability, and thus h is of maximal order.

[0053] **Key Generation:** The public parameters are n, g and $h = g^x$ by randomly choosing a value $x \in [1, \mathrm{ord}(\mathbb{G})]$(where g is generated by first choosing a random element $\alpha \in \mathbb{Z}_{n^2}^*$, and then setting $g = \alpha^2 mod\ n^2$).

[0054] **Encrypt:** Given a message $m \in \mathbb{Z}_n$, random number r is chosen uniformly in $\mathbb{Z}_{n^2}^*$. The ciphertext is computed as $(T, T') = \{ h^r(1 + m * n)\ (mod\ n^2), g^r (mod\ n^2)\}$.

[0055] **Decryption:** Knowing x, one can obtain m as follows:

[0056] $m = L(T/(T')^x\ mod\ n^2)$, where $L(u) = (u - 1)/n$.

[0057] Note: if the factorization of $n = p * q$ ($\lambda(n) = (p - 1)(q - 1)/2$) is given, one can directly decrypt the ciphertext by computing $m = L\big((T)^{\lambda(n)}\big)[\lambda(n)]^{-1} mod\ n$.

[0058] In this section, we briefly introduce the original attempt of proxy re-encryption, PRE, in Ateniese et al. "Improved proxy re-encryption schemes with applications to secure distributed storage" (ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 1-30, 2006), which lays the foundation of our proposed schemes. Generally, the ciphertext that can be only decrypted by Entity A can be converted into the one that can be decrypted by Entity B through the re-encryption of a proxy.

[0059] The PRE is based on Cramper-Shoup and EDD. It has the same operation of key generation as EDD. Thus, we skip it and focus on re-encryption operation.

[0060] **Data Encryption**: to encrypt a message $m \in \mathbb{Z}_n$, select a random $r \in [1, n/4]$ and compute: $(T, T') = \{h^r(1 + m * n)(mod\ n^2), g^r\}$.

[0061] **Decryption with x**: $m = L(T/(T')^x mod\ n^2)$, where $L(u) = (u - 1)/n$, for all $u \in \{u < n^2 | u = 1\ mod\ n\}$.

[0062] **Proxy Re-encryption**: the secret key x is divided into two shares x_1 and x_2, such that $x = x_1 + x_2$. The share x_1 is given to the proxy, while x_2 is kept by Entity B.

[0063] **Re-encryption** (done by the proxy): $(\tilde{T}, \widetilde{T'}) = (T/(T')^{x_1}, T') = (\ h^r(1 + m * n)(mod\ n^2)/(g^r)^{x_1}, g^r)$;

[0064] **Decryption** (done by Entity B): $m = L(\tilde{T}/(\widetilde{T'})^{x_2} mod\ n^2)$.

[0065] This is a proxy-invisible scheme, which means that it is unable to distinguish the original ciphertext from the re-encrypted ciphertext. In our proposed scheme, two servers play as an invisible proxy to transfer the encrypted data to the ciphertext under the key of an authorized requester. Moreover, the non-colluding servers help ensure the correct transformation of ciphertext.

[0066] The following table summarizes notation used herein:

Symbols	**Description**
g	The system generator that is public;
n	The system parameter;
(SK_{DSP}, PK_{DSP})	The key pair of DSP;
(SK_{CP}, PK_{CP})	The key pair of CP
$PK = PK_{DSP}{}^{SK_{CP}} = PK_{CP}{}^{SK_{DSP}}$	The public parameter based on keys of DSP and CP;
(sk_j, pk_j)	The key pair of DR j;
(sk_i, pk_i)	The key pair of DP i;
m_i	The raw data provided by DP i;
$[m]$	The ciphertext of m under PK;
$[m]^+$	The re-encryption result of m by DSP;
$[m]_{pk_i}$	The ciphertext of m under pk_i;
r	The random number;
r_*	The random number chosen for encryption;
CID	The public computation identifier, which is related to the operation type;
N	The number of data providers;

$\mathcal{L}(*)$	The bit length of input data;
$H()$	The hash function;
$\mathcal{R}$	The set of data providers related to CID;
$\|\mathcal{R}\|$	The number of providers in $\mathcal{R}$.

[0067] In order to support privacy-preserving data processing, we adapt the EDD to encrypt personal data with the Diffie-Hellman key of two servers, i.e., $PK = {PK_{DSP}}^{SK_{CP}} = {PK_{CP}}^{SK_{DSP}}$. We design the primitive, HRES, which can realize proxy-invisible re-encryption and secure data processing. The HRES consists of the following algorithms:

[0068] **KeyGen:** Let k be a security parameter and p, q be two large primes, where $\mathcal{L}(p) = \mathcal{L}(q) = k$ ($\mathcal{L}(\cdot)$ returns the bit length of input data). Due to the property of safe primes, there exist two primes p' and q' which satisfy that $p = 2p' + 1$, $q = 2q' + 1$. We compute $n = p * q$ and choose a generator g with order $\lambda = 2p'q'$, which can be chosen by selecting a random number $z \in \mathbb{Z}_{n^2}^*$ and computing $g = -z^{2n}$. The value λ can be used to decrypt the encrypted data, but we choose to conceal it and protect it from all involved parties. In the HRES, we only use key pair (sk, g^{sk}) for data encryption and decryption. The DSP 120 and the CP 130 generate their key pairs: $(SK_{DSP} = a, PK_{DSP} = g^a)$ and $(SK_{CP} = b, PK_{CP} = g^b)$, and then negotiate their Diffie-Hellman key $PK = {PK_{DSP}}^{SK_{CP}} = {PK_{CP}}^{SK_{DSP}} = g^{a*b}$. To support encrypted data processing, PK is public to all involved parties. Cloud user i generates its key pair $(sk_i, pk_i) = (k_i, g^{k_i})$. The public system parameters include $\{g, n, PK\}$.

[0069] First, the **Original Encryption** scheme is directly obtained from EDD.

[0070] **Encryption (Enc):** For a personal purpose, a user can outsource private data with its own key pair, which can ensure the unavailability of data to other entities. It can also be used to send data to a specified target. User encrypts their data with the public key of user i and a random $r \in [1, n/4]$ as follows:

[0071] $[m]_{pk_i} = \left((1 + m * n){pk_i}^r,\ g^r\right) (mod\ n^2)$.

[0072] **Decryption (Dec):** Upon receiving the encrypted data under its own public key, user i can directly decrypt it to obtain the original data:

[0073] $$m = L\left(\frac{(1+m*n)pk_i{}^r}{(g^r)^{k_i}} mod\ n^2\right)$$

[0074] Second, the following encryption is a **Two-Level Decryption** scheme that can support outsourced data processing flexibly.

[0075] **Encryption with Two Keys (EncTK):** To support ciphertext process flexibly, we propose to encrypt original data under the keys of two servers. Given a message $m_i \in \mathbb{Z}_n$ provided by user i, we first select a random number $r \in [1, n/4]$ and then encrypt it with PK. The ciphertext is generated as $[m_i] = \{T_i, T_i'\}$, where $T_i' = g^r mod\ n^2$, $T_i = (1 + m_i * n) * PK^r mod\ n^2$.

[0076] Note: $[m_i]$ denotes the ciphertext of m_i encrypted with PK, which can only be decrypted under the cooperation of the DSP and the CP. $[m_i]_{pk_i}$ denotes the data that is encrypted with pk_i, which can be decrypted by user i.

[0077] **Partial Decryption with SK_{DSP} (PDec1):** Once $[m_i]$ is received by the DSP, algorithm $PDec1()$ will be run to transfer it into another ciphertext which can be decrypted by the CP as follows:

$$[m_i]_{PK_{CP}} = \{T_i^{(1)}, T_i'^{(1)}\} = \{T_i, (T_i')^{SK_{DSP}}\}$$

$$= \{(1 + m_i * n)PK^r\ mod\ n^2, g^{r*a}\ mod\ n^2\}$$

$$= \{(1 + m_i * n)PK_{CP}{}^{a*r} mod\ n^2, g^{r*a}\ mod\ n^2\}.$$

[0078] **Partial Decryption with SK_{CP} (PDec2):** Once the message $[m_i]_{PK_{CP}}$ is received, the CP can directly decrypt it with its own secret key as follows:

[0079] 1) Select the public computation identifier CID, which specifies the operation type.

[0080] 2) $h_1 = H\left((pk_j)^{SK_{DSP}} || CID\right)$;

[0081] 3) $[m_i]^+ = \{\hat{T}, \hat{T}'\} = \{T_i, (T_i')^{SK_{DSP}} * g^{h_1}\}$.

[0082] To achieve the proxy-invisible re-encryption, we further propose a **Somewhat Re-Encryption** scheme, wherein differently from the scheme described above, it aims to transfer the encrypted data to the ciphertext under the public key of an authorized

requester. Here, we assume DR j with key pair$(sk_j, pk_j) = (t, g^t)$ requires to obtain m_i through outsourced data $[m_i]$. In our scheme, the transformation needs the cooperation and recognition of both the DSP and the CP. They together play the role of a proxy.

[0083] **First Phase of Re-Encryption (FPRE):** In order to prevent the decryption $PDec2()$ by the CP, the DSP initiates algorithm $FPRE()$ as follows:

[0084] 1) Select the public computation identifier CID, which specifies the operation type.

[0085] 2) $h_1 = H\left((pk_j)^{SK_{DSP}} || CID\right)$;

[0086] 3) $[m_i]^+ = \{\hat{T}, \hat{T}'\} = \{T_i, (T_i')^{SK_{DSP}} * g^{h_1}\}$. Notes: $[\mathrm{m_i}]^+$ denotes the encrypted data that can only be decrypted by an authorized data requester with the support of the CP 130.

[0087] **Second Phase of Re-Encryption (SPRE):** Upon receiving the data packet $[m_i]^+$, the CP launches re-encryption algorithm $SPRE()$ as below:

[0088] 1) $h_2 = H\left((pk_j)^{SK_{CP}} || CID\right)$;

2) $[m_i]_{pk_j} = \{\bar{T}, \bar{T}'\} = \left\{\hat{T}, (\hat{T}')^{SK_{CP}} * g^{h_2}\right\}$. Note: If data outsourced by a user are extremely private and not allowed to be processed or analyzed, the cloud user can choose to use $Enc()$ to encrypt its personal data with its own public key and then store them at the DSP. If the data can be analyzed in a privacy-preserving way by authorized parties, the user need to store encrypted data by calling the encryption algorithm $EncTK()$.

[0089] FIGURE 3 illustrates an example apparatus capable of supporting at least some embodiments of the present invention. Illustrated is device 300, which may comprise, for example, a DP 110, DSP 120, CP 130 or DR 140 of FIGURE 1 or FIGURE 2. Comprised in device 300 is processor 310, which may comprise, for example, a single- or multi-core processor wherein a single-core processor comprises one processing core and a multi-core processor comprises more than one processing core. Processor 310 may comprise more than one processor. A processing core may comprise, for example, a Cortex-A8 processing core manufactured by ARM Holdings or a Steamroller processing core produced by Advanced Micro Devices Corporation. Processor 310 may comprise at

least one Qualcomm Snapdragon and/or Intel Atom processor. Processor 310 may comprise at least one application-specific integrated circuit, ASIC. Processor 310 may comprise at least one field-programmable gate array, FPGA. Processor 310 may be means for performing method steps in device 300. Processor 310 may be configured, at least in part by computer instructions, to perform actions.

[0090] Device 300 may comprise memory 320. Memory 320 may comprise random-access memory and/or permanent memory. Memory 320 may comprise at least one RAM chip. Memory 320 may comprise solid-state, magnetic, optical and/or holographic memory, for example. Memory 320 may be at least in part accessible to processor 310. Memory 320 may be at least in part comprised in processor 310. Memory 320 may be means for storing information. Memory 320 may comprise computer instructions that processor 310 is configured to execute. When computer instructions configured to cause processor 310 to perform certain actions are stored in memory 320, and device 300 overall is configured to run under the direction of processor 310 using computer instructions from memory 320, processor 310 and/or its at least one processing core may be considered to be configured to perform said certain actions. Memory 320 may be at least in part comprised in processor 310. Memory 320 may be at least in part external to device 300 but accessible to device 300.

[0091] Device 300 may comprise a transmitter 330. Device 300 may comprise a receiver 340. Transmitter 330 and receiver 340 may be configured to transmit and receive, respectively, information in accordance with at least one cellular or non-cellular standard. Transmitter 330 may comprise more than one transmitter. Receiver 340 may comprise more than one receiver. Transmitter 330 and/or receiver 340 may be configured to operate in accordance with global system for mobile communication, GSM, wideband code division multiple access, WCDMA, long term evolution, LTE, IS-95, wireless local area network, WLAN, Ethernet and/or worldwide interoperability for microwave access, WiMAX, standards, for example.

[0092] Device 300 may comprise a near-field communication, NFC, transceiver 350. NFC transceiver 350 may support at least one NFC technology, such as NFC, Bluetooth, Wibree or similar technologies.

[0093] Device 300 may comprise user interface, UI, 360. UI 360 may comprise at least one of a display, a keyboard, a touchscreen, a vibrator arranged to signal to a user by

causing device 300 to vibrate, a speaker and a microphone. A user may be able to operate device 300 via UI 360, for example to manage ciphertext-form data.

[0094] Device 300 may comprise or be arranged to accept a user identity module 370. User identity module 370 may comprise, for example, a subscriber identity module, SIM, card installable in device 300. A user identity module 370 may comprise information identifying a subscription of a user of device 300. A user identity module 370 may comprise cryptographic information usable to verify the identity of a user of device 300 and/or to facilitate encryption of communicated information and billing of the user of device 300 for communication effected via device 300.

[0095] Processor 310 may be furnished with a transmitter arranged to output information from processor 310, via electrical leads internal to device 300, to other devices comprised in device 300. Such a transmitter may comprise a serial bus transmitter arranged to, for example, output information via at least one electrical lead to memory 320 for storage therein. Alternatively to a serial bus, the transmitter may comprise a parallel bus transmitter. Likewise processor 310 may comprise a receiver arranged to receive information in processor 310, via electrical leads internal to device 300, from other devices comprised in device 300. Such a receiver may comprise a serial bus receiver arranged to, for example, receive information via at least one electrical lead from receiver 340 for processing in processor 310. Alternatively to a serial bus, the receiver may comprise a parallel bus receiver.

[0096] Device 300 may comprise further devices not illustrated in FIGURE 3. For example, where device 300 comprises a smartphone, it may comprise at least one digital camera. Some devices 300 may comprise a back-facing camera and a front-facing camera, wherein the back-facing camera may be intended for digital photography and the front-facing camera for video telephony. Device 300 may comprise a fingerprint sensor arranged to authenticate, at least in part, a user of device 300. In some embodiments, device 300 lacks at least one device described above. For example, some devices 300 may lack a NFC transceiver 350 and/or user identity module 370.

[0097] Processor 310, memory 320, transmitter 330, receiver 340, NFC transceiver 350, UI 360 and/or user identity module 370 may be interconnected by electrical leads internal to device 300 in a multitude of different ways. For example, each of the aforementioned devices may be separately connected to a master bus internal to device 300,

to allow for the devices to exchange information. However, as the skilled person will appreciate, this is only one example and depending on the embodiment various ways of interconnecting at least two of the aforementioned devices may be selected without departing from the scope of the present invention.

[0098] FIGURE 4 illustrates signalling in accordance with at least some embodiments of the present invention. On the vertical axes are disposed, from the left to the right, DP 110, DSP 120, CP 130 and DR 140. Time advances from the top toward the bottom.

[0099] Phase 410, which takes place in all the four entities, comprises a system setup phase. DSP 120 and CP 130 get their own key pairs respectively $(SK_{DSP}, PK_{DSP}) = (a, g^a)$ and $(SK_{CP}, PK_{CP}) = (b, g^b)$, and then negotiate $PK = {PK_{CP}}^{SK_{DSP}} = {PK_{DSP}}^{SK_{CP}} = g^{a*b}$. The corresponding secret key of PK is $a * b$. DP 110, that is, user i, generates its own key pair $(sk_i, pk_i) = (k_i, g^{k_i})$ by randomly choosing $k_i \in [1, n/4]$, and then registers at the CP. The public parameters are $(n, g, PK, \mathcal{L}(n))$ and the public keys of all involved entities. DR 140 j performs similarly to generate its key pair $(sk_j, pk_j) = (k_j, g^{k_j}) = (t, g^t)$. If multiple CPs 130 are employed in the system, each CP 130 may negotiate a Diffie-Hellman key with the DSP 120 and publish this key to its customers. For simplifying presentation, we only present the detailed operations in the case that there is only one CP 130 interacting with the DSP 120 as below.

[00100] Phase 420 comprises a data upload from DP 110 to DSP 120. DPs 110 encrypt their data before uploading it to the DSP 120. DP i 110 calls $EncTK()$ to encrypt data m_i:

[00101] $[m_i] = (T_i, T_i') = \{(1 + m_i * \mathrm{n}) * PK^{r_i} \, mod \, n^2 \, , \, g^{r_i} \, mod \, n^2\}$.

[00102] For correctly supporting various computations, the length of the data may be restricted $\mathcal{L}(m_i) < \mathcal{L}(n)/4$. Then DP i 110 uploads and stores $[m_i]$ at the DSP 120.

[00103] Phase 430 comprises DR 140 requesting the data uploaded in phase 420, by signalling to CP 130. The request of phase 430 may comprise a computation identifier, CID. The request may comprise a request to obtain the data in processed and encrypted form. The request may comprise a public key of DR 140.

[00104] Phase 440 may comprise CP 130 assessing, whether DR 140 is authorized to access the data. In case no, processing may stop here. In case DR 140 is authorized, CP 130 may forward the request to DSP 120, the forwarding being illustrated in FIGURE 4 as phase 450. The forwarded request may likewise comprise the CID, and/or a public key of DR 140.

[00105] Phase 460 comprises DSP 120 pre-processing the requested data, upon receiving the request from DR 140 authorized by CP 130, according to the computation identifier CID by calling the algorithm $FPRE()$, which is described above, to prepare data packet DPacket for CP 130. Phase 470 comprises DSP 120 providing the processed data, DPacket, to CP 130. Phase 460 may further comprise DSP 120 performing a mathematical manipulation of the data in encrypted form, in dependence of the CID.

[00106] Phase 480 comprises CP 130 further processing DPacket, received from DSP 120, by calling the algorithm $SPRE()$ to obtain DPacket'. Algorithm $SPRE()$ is described above. Alternatively, phase 480 may comprise CP 130 transferring DPacket under DR 140's public key using a different method, such as by first using $PDec2()$ and then $Enc()$. These algorithms are described above. Finally, in phase 490, DPacket' is provided to DR 140 as a response to the request of phase 430. Once DR 140 has DPacket', it may decrypt it, for example by calling algorithm $DPRE()$ to obtain the data. $DPRE()$ is described above.

[00107] In the following, processing of encrypted data will be described with respect to the computations listed above, namely The HRES primitive may support seven basic operations, indicated by different CID: 1) addition; 2) subtraction; 3) multiplication; 4) sign acquisition; 5) comparison; 6) equivalent Test; and 7) variance.

[00108] System setup and data collection have no difference from that in FIGURE 4. Hence, we ignore the process of data upload and mainly focus on phases 460, 480 and 490 in the following presentation. DP i 110 offers personal data as $[m_i]$ and DR j 140 holds the key pair $(sk_j, pk_j) = (k_j, g^{k_j}) = (t, g^t)$.

[00109] **Addition**: this scheme aims to obtain the sum of all raw data ($m = \sum_{i \in \mathcal{R}} m_i$) where $\mathcal{R}$ represents the set of data providers related to CID in the request.

[00110] **Phase 460 (Data Preparation at DSP)**: Due to the additive homomorphism, the DSP can directly multiply encrypted data one by one as following:

[00111] $[m] = (T, T') = (\prod_{i\in\mathcal{R}} T_i, \prod_{i\in\mathcal{R}} T_i')$.

[00112] To transfer it into the ciphertext under DR j's public key, the DSP further calls the algorithm $FPRE()$ to process the data with its own secret key and DR j's public key pk_j:

[00113] $[m]^+ = (\hat{T}, \hat{T'}) = \{T, (T')^a * g^{H((pk_j)^a||CID)}\}$.

[00114] The DSP finally prepares a data packet $([m]^+, CID)$ and sends it to the CP.

[00115] **Phase 480 (Data Process at CP)**: The CP calls the second re-encryption algorithm $SPRE()$ to finally transfer the encrypted data to the ciphertext under DR j's public key:

[00116] $[m]_{pk_j} = (\bar{T}, \bar{T'}) = \{\hat{T}, (\hat{T'})^b * g^{H((pk_j)^b||CID)}\}$.

[00117] Then the CP sends $\left([m]_{pk_j}, CID\right)$ to the DR.

[00118] **Phase 490 (Data Access at DR)**: The DR can obtain the aggregated result by calling the algorithm $DPRE()$:

[00119] $m = L\left(\bar{T} * PK_{CP}{}^{H((PK_{DSP})^{sk_j}||CID)} * g^{H((PK_{CP})^{sk_j}||CID)} / \bar{T'} \bmod n^2\right)$.

[00120] **Subtraction**: this function aims to obtain the subtraction of some data (for example, $m = m_1 - m_2$) with encrypted data $[m_1]$ and $[m_2]$. The HRES has an additional property as follows:

[00121] $[m_i]^{n-1} = \{\{(1 + m_i * n)PK^{r_i}\}^{(n-1)}, (g^{r_i})^{(n-1)}\} = \{(1 + m_i(n-1) * n)PK^{r_i(n-1)}, g^{r_i(n-1)}\} = \{(1 - m_i * n)PK^{r_i(n-1)}, g^{r_i(n-1)}\} \bmod n^2 = [-m_i]$

[00122] **Phase 460 (Data Preparation at DSP)**: The DSP first computes $[-m_2] = [m_2]^{n-1}$, and then multiplies it with $[m_1]$ to obtain $[m] = [m_1 - m_2]$.

[00123] Then the subsequent process is the same as that in Addition. For length and simplicity reasons, we skip its details.

[00124] **Multiplication**: This function aims to obtain the product of all raw data ($m = \prod_{i \in \mathcal{R}} m_i$). For ease of presentation, we describe the details with two pieces of data ($[m_1]$, $[m_2]$). The DR wants to get the multiplication result $m = m_1 * m_2$.

[00125] Note that the available number of the data in multiplication influences the length of raw data. If we need to get the product of f pieces of data, it must be guaranteed that the length of each raw data $\mathcal{L}(m_i) < \mathcal{L}(n)/(2f)$.

[00126] **Phase 460 (Data Preparation at DSP)**: First, the DSP chooses two random numbers c_1, c_2 (the number of random numbers is equal to that of provided data) and sets another one $c_3 = (c_1 * c_2)^{-1} \bmod n$.

[00127] To conceal each raw data from the CP, the DSP does one exponentiation and one decryption with its own secret key by calling $PDec1()$. Then the DSP encrypts c_3 with $Enc()$ using the public key pk_j of the requesting DR:

[00128] $[c_1 * m_1] = \{T_1, (T_1')^{c_1}\}$

[00129] $[c_1 * m_1]_{PK_{CP}} = \left(T_1^{(1)}, T_1'^{(1)}\right) = \{T_1^{c_1}, (T_1')^{c_1 * a}\} = \{(1 + c_1 * m_1 * n) * PK^{r_1 * c_1}, g^{r_1 * a * c_1}\}$

[00130] $[c_2 * m_2] = \{T_2, (T_2')^{c_2}\}$

[00131] $[c_2 * m_2]_{PK_{CP}} = \left(T_2^{(1)}, T_2'^{(1)}\right) = \{T_2^{c_2}, (T_2')^{c_2 * a}\} = \{(1 + c_2 * m_2 * n) * PK^{r_2 * c_2}, g^{r_2 * a * c_2}\}$

[00132] $[c_3]_{pk_j} = (T_j, T_j') = \{(1 + c_3 * n) * pk_j^{r_3}, g^{r_3}\}$

[00133] The data packet sent to the CP 130 is $\{[c_1 * m_1]_{PK_{CP}}, [c_2 * m_2]_{PK_{CP}}, [c_3]_{pk_j}\}$.

[00134] **Phase 480 (Data Process at CP)**: Upon receiving the data packet from the CSP, the CP uses the algorithm $PDec2()$ to decrypt the data:

[00135] $c_1 * m_1 = T_1^{(1)} / (T_1'^{(1)})^b$,

[00136] $c_2 * m_2 = T_2^{(1)} / (T_2'^{(1)})^b$.

[00137] It further multiplies the two values and then calls $Enc()$ to encrypt it as $(T, T') = [c_1 * c_2 * m_1 * m_2]_{pk_j}$. Finally, the CP 130 forwards (T, T') and $[c_3]_{pk_j}$ to the DR 140.

[00138] **Phase 490 (Data Access at DR)**: the DR 140 can obtain the product by calling $Dec()$ to decrypt the two ciphertexts with its secret key:

[00139] $$m = m_1 * m_2 = L\left(\frac{T}{(T')^{sk_j}} mod\ n^2\right) * L\left(\frac{T_j}{(T_j')^{sk_j}} mod\ n^2\right) mod\ n.$$

[00140] **Sign Acquisition**: As $\mathcal{L}(m) < \mathcal{L}(n)/4$, we assume that BIG is the largest raw data of m. Then the raw data is in the scope $[-BIG, BIG]$. DR j wants to know the sign of raw data m_1 from $[m_1]$.

[00141] **Phase 460 (Data Preparation at DSP)**: The DSP chooses a random number c_1 where $\mathcal{L}(c_1) < \mathcal{L}(n)/4$. It first computes

$$[2 * m_1 + 1] = (T, T') = \{{T_1}^2 * (1 + n) * PK^{r'}, {T_1}'^2 * g^{r'}\}$$
$$= \{(1 + (2 * m_1 + 1) * n) * PK^{r'+2*r_1},\ g^{r'+2*r_1}\}$$

[00142] Then it flips a coin s. If $s = 0$; it computes as follows: $\left({T_1}^{(1)}, {T_1}'^{(1)}\right) = \left\{T^{n-c_1}, (T')^{a*(n-c_1)}\right\} = [-c_1 * (2 * m_1 + 1)]$. Otherwise, it calls $PDec1()$ and computes: $\left({T_1}^{(1)}, {T_1}'^{(1)}\right) = \{T^{c_1}, T'^{a*c_1}\} = [c_1 * (2 * m_1 + 1)]$.

[00143] The DSP also encrypts s with pk_j through $Enc()$: $[s]_{pk_j} = (T_s, T_s') = \{(1 + s * n) * {pk_j}^{r_s},\ g^{r_s}\}$. The data packet sent to the CP is $\{\left({T_1}^{(1)}, {T_1}'^{(1)}\right), [s]_{pk_j}\}$.

[00144] **Phase 480 (Data Process at CP)**: Upon receiving the data packet from the DSP 120, the CP 130 decrypts $\left({T_1}^{(1)}, {T_1}'^{(1)}\right)$ with $PDec2()$ to obtain raw data $m' = (-1)^{s+1} * c_1 * (2 * m_1 + 1)\ mod\ n^2$. The CP compares $\mathcal{L}(m')$ with $\mathcal{L}(n)/2$. If $\mathcal{L}(m') < \mathcal{L}(n)/2$, it calls $Enc()$ to encrypt $u = 1$ with pk_j; otherwise, it encrypts $u = 0$ with pk_j;

[00145] $[u]_{pk_j} = (T_u, T_u') = \{(1 + u * n) * {pk_j}^{r_u},\ g^{r_u}\}$

[00146] It further multiplies the two ciphertexts.

[00147] $[s + u]_{pk_j} = (T, T') = \{T_s * T_u, T_u' * T_s'\}$. Finally, the CP forwards (T, T') to DR j.

[00148] **Phase 490 (Data Access at DR)**: DR j can call $Dec()$ to obtain the final result: $u + s = L(T/(T')^{sk_j}\ mod\ n^2)$. Then DR j needs to check it and determine the sign of raw data: if $u + s = 1$, the original data is negative (i.e., $m_1 < 0$); otherwise, it is positive or zero (i.e., $m_1 \geq 0$). Note: if $(s = 1, u = 0)$ or $(u = 1, s = 0)$, $m_1 < 0$; if $(s = 1, u = 1)$ or $(u = 0, s = 0)$, $m_1 \geq 0$.

[00149] **Comparison**: Similar to the schemes above, DR j wants to compare the raw data (m_1, m_2) based on their encrypted data. For ease of presentation, $m_1 - m_2$ is denoted as m_{1-2}.

[00150] $[m_1] = (T_1, T_1') = \{(1 + m_1 * \mathrm{n}) * PK^{r_1},\ g^{r_1}\}$

[00151] $[m_2] = (T_2, T_2') = \{(1 + m_2 * \mathrm{n}) * PK^{r_2}, g^{r_2}\}$

[00152] **Phase 460 (Data Preparation at DSP)**: DSP first computes to get the subtraction of encrypted data:

[00153] $(T, T') = \{T_1 * (T_2)^{n-1}, T_1' * (T_2')^{n-1}\} = [(m_1 - m_2)]$.

[00154] The following steps are the same to that in Sign Acquisition, which is skipped for the reason of length limitation. Through the cooperation of the DSP 120 and the CP 130, the DR 140 finally gets the sign of $m_{1-2} = m_1 - m_2$. In the end, the DR can obtain the comparison result. If $m_{1-2} \geq 0$, $m_1 \geq m_2$; otherwise, $m_1 < m_2$.

[00155] **Equivalent test**: If DR j 140 wants to know if m_1 is equal to m_2 with encrypted data ($[m_1]$, $[m_2]$). The DSP 120 and the CP 130 directly interact with each other in two parallel computations of Comparison.

[00156] They compare m_1 and m_2 in two forms: 1) $m_{1-2} = m_1 - m_2$; 2) $m_{2-1} = m_2 - m_1$. Through the operations in Comparison, DSP can get two computation results $[s_1 + u_1]_{pk_j}$ and $[s_2 + u_2]_{pk_j}$ respectively.

[00157] To conceal the comparing result of m_1 and m_2, $[s_1 + u_1]_{pk_j}$ and $[s_2 + u_2]_{pk_j}$ are sent to the DR in a random order. If both testing result are "≥", we can know $m_1 = m_2$.

[00158] **Variance**: In some scenarios, DR j 140 may want to get the variance of some data according to provided encrypted data. In this presentation, we set N be the number of provided data and $m = \sum_{i=1}^{N} m_i$. Variance function can be presented as $M = \frac{\sum_{i=1}^{N}(m_i - \bar{m})^2}{N} = \frac{\sum_{i=1}^{N}(N * m_i - m)^2}{N^3}$, where $\bar{m}$ is the average of m_i $(i = 1, \ldots, N)$. For ease of presentation, we assume there are three pieces of encrypted data (i.e., $N = 3$): $[m_1]$, $[m_2]$ and $[m_3]$.

[00159] **Phase 460 (Data Preparation at DSP)**: First, the DSP 120 obtains $[N * m_i - \sum_{i=1}^{N} m_i]$ through following steps:

[00160] $[m] = (T, T') = [m_1] * [m_2] * [m_3]$,

[00161] $[-m] = (T^{n-1}, (T')^{n-1})$;

[00162] $[N * m_i] = [m_i]^N$ for $i = 1,2,3$;

[00163] $[N * m_i - m] = [m_i]^N * [-m]$ for $i = 1,2,3$;

[00164] Then the DSP 120 partially decrypts the data with its secret key by calling $PDec1()$ to obtain: $[N * m_i - m]_{PK_{CP}}$ for $i = 1,2,3$. The DSP 120 chooses three random numbers c_1, c_2, c_3, and computes to obtain:

[00165] $[c_i(N * m_i - m)]_{PK_{CP}} = \left([N * m_i - m]_{PK_{CP}}\right)^{c_i}$ for $i = 1,2,3$

[00166] Then the DSP 120 send the three ciphertexts to the CP 130. In addition, DSP 120 needs to store ${c_1}^2, {c_2}^2, {c_3}^2$.

[00167] **Phase 480 (Data Process at CP)**: Upon receiving the data from the DSP, the CP directly decrypts to obtain raw data and then processes the data for DR j as follows:

[00168] Decrypt to obtain: $C_i = c_i(\mathrm{N} * m_i - m)$ for $i = 1,2,3$;

[00169] Encrypt processed data with the public key pk_j of DR j 140:

[00170] $\left[{C_i}^2\right]_{pk_j} = [{c_i}^2(N * m_i - m)^2]_{pk_j}$ for $i = 1,2,3$. Then CP 130 sends them back to DSP 120. **Additional Operation at DSP**: The DSP first computes the reverse of ${c_1}^2, {c_2}^2, {c_3}^2$ respectively: $c_i' = (c_i^2)^{-1} mod\ n^2$ for $i = 1,2,3$. Then DSP 120 can prepare the final result for DR j:

$$[M']_{pk_j} = \left(\left[{C_1}^2\right]_{pk_j}\right)^{c_1'} * \left(\left[{C_2}^2\right]_{pk_j}\right)^{c_2'} * \left(\left[{C_3}^2\right]_{pk_j}\right)^{c_3'}$$
$$= [(N * m_1 - m)^2 + (N * m_2 - m)^2 + (N * m_1 - m)^2]_{pk_j}$$

[00171] Finally, $[M']_{pk_j}$ can be sent to DR j.

[00172] **Phase 490 (Data Access at DR)**: DR j can obtain M' by calling $Dec()$ and then get the variance:

[00173] $M' = (N * m_1 - m)^2 + (N * m_2 - m)^2 + (N * m_1 - m)^2$;

[00174] $M = M'/N^3$.

[00175] In the following, processing involving multiple CPs 130 will be described. Due to length limitations, we will only present such basic operations as addition, subtraction, multiplication, and comparison across the CPs 130 in this section. We set an example of two encrypted data belonging to two CPs: CP B and CP V. Besides the settings above, we further set the key pair of B and V as $(SK_b, PK_b) = (b, g^b mod\ n^2)$ and $(SK_v, PK_v) = (v, h^v)$. Hence, we have $PK = {PK_b}^a = {PK_a}^b$ and $PK' = {PK_v}^a = {PK_a}^v$. Two messages are encrypted as:

[00176] $[m_1]_{PK} = \{T_1 = (1 + m_1 * n)PK^{r_1} mod\ n^2, T_1' = g^{r_1} mod\ n^2\}$,

[00177] $[m_2]_{PK'} = \{T_2 = (1 + m_1 * n)PK'^{r_2} mod\ n^2, T_2' = g^{r_2} mod\ n^2\}$.

[00178] That is to say, the data provider of m_1 trusts CP B; while the data provider of m_2 trusts CP V. Hence, they encrypt their data with the corresponding Diffie-Hellman key (PK or PK').

[00179] DR j with key pair $(sk_j, pk_j) = (k_j, g^{k_j} \bmod n^2)$ wants to obtain a data processing result across CPs. We assume DR j is a customer of CP B. The detailed procedure is introduced as follows.

[00180] **Addition across CPs 130**: This computation wants to obtain the sum of data over two servers.

[00181] **Data Preparation at DSP**: DSP selects a random number w and then operates as follows: 1) Encrypt w and $-w$: $[w]_{PK}$ and $[-w]_{PK'}$; and 2) Compute $[m_1 + w]_{PK}$ and $[m_2 - w]_{PK'}$; then 3) call $PDec1()$ to re-encrypt the two data to obtain $[m_1 + w]_{PK_B}$ and $[m_2 - w]_{PK_V}$.

[00182] **Data Process at CPs:** Upon receiving $[m_1 + w]_{PK_B}$ CP B first checkes its CID and determines if the requester is allowed to access the data; if positive, CP B calls $PDec2()$ to obtain the fused raw data $m_1 + w$ and then encrypt it with DR j's public key $[m_1 + w]_{pk_j}$. Similar to the operations of CP B, CP V also obtains $[m_2 - w]_{pk_j}$.

[00183] **Additional Operation at DSP:** DSP 120 multiplies the two ciphertexts to obtain $[m_1 + m_2]_{pk_j}$ and then forwards it to DR. Finally DR j can directly get the sum of data $(m_1 + m_2)$ by calling $Dec()$.

[00184] **Subtraction across CPs 130**: the operation is similar to addition, but it needs to do one more operation to obtain the negative of subtractor by doing exponentiation with the power of $(n - 1)$.

[00185] **Multiplication across CPs 130**: Different from Multiplication described earlier with reference to a single CP 130, multiple CPs are involved in the computation and leads to a slightly higher computation on the CPs.

[00186] **Data Preparation at DSP**: The DSP selects two random numbers (c_1, c_2) to conceal the raw data, and set $c_3 = (c_2 * c_2)^{-1} \bmod n$. Then the DSP does the same operations about Multiplication as those described above and obtains:

[00187] $[c_1 * m_1]_{PK_B} = (\widehat{T_1}, \widehat{T_1'}) = \{{T_1}^{c_1}, (T_1')^{a*c_1}\} = \{(1 + c_1 * m_1 * n) * PK^{r_1*c_1}, g^{r_1*a*c_1}\}$

$[c_2 * m_2]_{PK_V} = (\widehat{T_2}, \widehat{T_2'}) = \{{T_2}^{c_2}, (T_2')^{a*c_2}\} = \{(1 + c_2 * m_2 * n) * PK'^{(r_2*c_2)}, g^{r_2*a*c_2}\}$

[00188] $[c_3]_{pk_j} = (T_j, T_j') = \{(1 + c_3 * n) * {pk_j}^{r_3}, g^{r_3}\}$

[00189] The data packet sent to CP B is $\{[c_1 * m_1]_{PK_B}, [c_3]_{pk_j}\}$; while the data $[c_2 * m_2]_{PK_V}$ is sent to CP V.

[00190] **Data Process at CPs:** Upon receiving the data package, the CP first checks the legality and its access policy, and then calls $PDec2()$ if it is positive. Concretely, the CP V obtains the value of $c_2 * m_2$, encrypts it with PK_B and then sends $[c_2 * m_2]_{PK_B}$ to the CP B. The CP B obtains the two plaintext and multiplies them to get $c_1 * c_2 * m_1 * m_2$.

[00191] Finally, the CP B encrypts $c_1 * c_2 * m_1 * m_2$ with the DR j's public key and sends it together with $[c_3]_{pk_j}$ to DR j.

[00192] **Data Access at DR:** Upon obtaining the data form the CP B, the DR can directly calls $Dec()$ to get the data of $c_1 * c_2 * m_1 * m_2$ and c_3. Finally, it can get:

[00193] $m = m_1 * m_2 = c_1 * c_2 * m_1 * m_2 * c_3 \ mod \ n.$

[00194] **Comparison across CPs 130**: Different from the Comparison over one CP, the initial operation is executed by the CPs 130 rather than the DSP 120.

[00195] First, the DSP directly sends the data $[m_1]_{PK}$ and $[m_2]_{PK'}$ to the CP B and the CP V respectively.

[00196] **Data Preparation at CPs:** The CP V calls $PDec1()$ to obtain $[m_2]_{PK_a}$ and then sends it to the CP B through a secure way. The CP B first decrypts to obtain $[m_1]_{PK_a}$ and computes as follows:

[00197] $[m_{1-2}]_{PK_a} = [m_1]_{PK_a} * \left([m_2]_{PK_a}\right)^{n-1}$;

[00198] $\{T, T'\} = [2 * m_{1-2} + 1]_{PK_a} = \left\{[m_{1-2}]_{PK_a}\right\}^2 * [1]_{PK_a}$;

[00199] Then it flips a coin s. If $s = 0$; it computes as follows:

[00200] $\left(\widehat{T_1}, \widehat{T_1'}\right) = \left\{T^{n-c_1}, (T')^{a*(n-c_1)}\right\}$

[00201] Otherwise, it computes $\left(\widehat{T_1}, \widehat{T_1'}\right) = \{T^{c_1}, T'^{a*c_1}\}$

[00202] The CP B also encrypts s with the public key of DR: $[s]_{pk_j} = (T_s, T_s')$, and then sends $\left(\widehat{T_1}, \widehat{T_1'}\right)$ and $[s]_{pk_j}$ to the DSP.

[00203] **Data Process at DSP:** The DSP decrypts $\left(\widehat{T_1}, \widehat{T_1'}\right)$ to obtain raw data $m' = (-1)^{s+1} * c_1 * [2 * (m_1 - m_2) + 1] \ mod \ n^2$. The DSP compares its length with $\mathcal{L}(n)/2$. If $\mathcal{L}(m') < \mathcal{L}(n)/2$, it encrypts $u = 1$ with pk_j; otherwise, it encrypts $u = 0$ with pk_j;

[00204] $[u]_{pk_j} = (T_u, T_u') = \left\{(1 + u * n) * {pk_j}^{r_u}, \ g^{r_u}\right\}$

[00205] It further multiplies the two ciphertexts.

[00206] $[s+u]_{pk_j} = (\bar{T}, \bar{T}') = \{T_s * T_u, T_s' * T_u'\}$

[00207] **Data Access at DR**: DR j can call $Dec()$ to obtain the final result: $u+s = \bar{T}/(\bar{T}')^{k_j}$. Then DR j determines the sign of m_{1-2}. If $u+s=1$, the original data is negative (i.e., $m_1 < m_2$); otherwise, it is positive or zero (i.e., $m_1 \geq m_2$). Comparing with the schemes described herein above, we can observe that the cross-CP computation does not introduce too much overhead. For example, in Addition, the DSP needs to do some encryptions on random numbers, but it calls $PDec1()$ rather than $SPRE()$, which is more efficient. The CP also only needs to do one more $PDec2()$ and one more $Enc()$. The computation cost of DR is low due to the high efficiency of $Dec()$.

[00208] FIGURE 5 is a flow graph of a method in accordance with at least some embodiments of the present invention. The phases of the illustrated method may be performed in DSP 120, or in a control device configured to control the functioning thereof, when implanted therein.

[00209] Phase 510 comprises receiving, from a data provider, a first ciphertext. Phase 520 comprises performing a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier. Phase 530 comprises obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation. Finally, phase 540 comprises providing the second ciphertext to a first computation party

[00210] FIGURE 6 is a flow graph of a method in accordance with at least some embodiments of the present invention. The phases of the illustrated method may be performed in CP 130, or in a control device configured to control the functioning thereof, when implanted therein.

[00211] Phase 610 comprises determining, based on a message from a data requester, a computation identifier. Phase 620 comprises transmitting a request to a data service provider, the request comprising the computation identifier. Phase 630 comprises receiving, from the data service provider, a first ciphertext. Phase 640 comprises obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation. Finally, phase 650 comprises providing the second ciphertext to the data requester as a response to the message.

[00212] It is to be understood that the embodiments of the invention disclosed are not limited to the particular structures, process steps, or materials disclosed herein, but are extended to equivalents thereof as would be recognized by those ordinarily skilled in the relevant arts. It should also be understood that terminology employed herein is used for the purpose of describing particular embodiments only and is not intended to be limiting.

[00213] Reference throughout this specification to one embodiment or an embodiment means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment. Where reference is made to a numerical value using a term such as, for example, about or substantially, the exact numerical value is also disclosed.

[00214] As used herein, a plurality of items, structural elements, compositional elements, and/or materials may be presented in a common list for convenience. However, these lists should be construed as though each member of the list is individually identified as a separate and unique member. Thus, no individual member of such list should be construed as a de facto equivalent of any other member of the same list solely based on their presentation in a common group without indications to the contrary. In addition, various embodiments and example of the present invention may be referred to herein along with alternatives for the various components thereof. It is understood that such embodiments, examples, and alternatives are not to be construed as de facto equivalents of one another, but are to be considered as separate and autonomous representations of the present invention.

[00215] Furthermore, the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments. In the preceding description, numerous specific details are provided, such as examples of lengths, widths, shapes, etc., to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or with other methods, components, materials, etc. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the invention.

[00216] While the forgoing examples are illustrative of the principles of the present invention in one or more particular applications, it will be apparent to those of ordinary

skill in the art that numerous modifications in form, usage and details of implementation can be made without the exercise of inventive faculty, and without departing from the principles and concepts of the invention. Accordingly, it is not intended that the invention be limited, except as by the claims set forth below.

[00217] The verbs "to comprise" and "to include" are used in this document as open limitations that neither exclude nor require the existence of also un-recited features. The features recited in depending claims are mutually freely combinable unless otherwise explicitly stated. Furthermore, it is to be understood that the use of "a" or "an", that is, a singular form, throughout this document does not exclude a plurality.

INDUSTRIAL APPLICABILITY

[00218] At least some embodiments of the present invention find industrial application in secure data processing.

ACRONYMS LIST

ABC	Definition
CID	computation identifier
CP	computation party
EDD	Emmanuel, Dario and David mechanism
DP	data provider
DSP	data service provider
HRES	homomorphic re-encryption scheme
PRE	proxy re-encryption
DR	data requester

REFERENCE SIGNS LIST

110	data provider
120	data service provider

130	computation party
140	data requester
300 – 370	structure of the device of FIGURE 3
410 – 490	phases of the method of FIGURE 4
510 – 540	phases of the method of FIGURE 5
610 – 650	phases of the method of FIGURE 6

WHAT IS CLAIMED IS:

1. An apparatus comprising at least one processing core, at least one memory including computer program code, the at least one memory and the computer program code being configured to, with the at least one processing core, cause the apparatus at least to:

- receive, from a data provider, a first ciphertext;
- perform a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier;
- obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- provide the second ciphertext to a first computation party.

2. The apparatus according to claim 1, wherein the apparatus is configured to obtain the computation identifier from the first computation party.

3. The apparatus according to claim 1 or 2, wherein the apparatus is further configured to participate in negotiating a shared secret with the first computation party.

4. The apparatus according to claim 3, wherein the cryptographic re-encryption operation is performed in dependence of the computation identifier.

5. The apparatus according to any of claims 1 - 4, wherein the second ciphertext is not decryptable solely by a secret key of the first computation party.

6. The apparatus according to any of claims 1 – 5, wherein the apparatus is further configured to obtain a key pair comprising a public key of the apparatus and a secret key of the apparatus.

7. The apparatus according to any of claims 1 – 6, wherein the computation identifier identifies at least one of the following processes: addition, subtraction, multiplication, sign acquisition, comparison, equivalence test and variance.

8. The apparatus according to any of claims 1 – 7, wherein the apparatus is further configured to obtain a third ciphertext from the first ciphertext, to provide the third ciphertext to a second computation party, and to obtain a fourth ciphertext from responses received in the apparatus from the first computation party and the second computation party, and to obtain an encrypted result of a computation process identified by the computation identifier.

9. An apparatus comprising at least one processing core, at least one memory including computer program code, the at least one memory and the computer program code being configured to, with the at least one processing core, cause the apparatus at least to:

- determine, based on a message from a data requester, a computation identifier;
- transmit a request to a data service provider, the request comprising the computation identifier;
- receive, from the data service provider, a first ciphertext;
- obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- provide the second ciphertext to the data requester as a response to the message.

10. The apparatus according to claim 9, wherein the apparatus is further configured to check an access policy before providing the request to the data service provider.

11. The apparatus according to claim 9 or 10, wherein the apparatus is further configured to participate in negotiating a shared secret with the data service provider.

12. The apparatus according to claim 11, wherein the negotiating comprises a Diffie-Hellman negotiation.

13. The apparatus according to any of claims 9 - 12, wherein the apparatus is configured to perform the cryptographic re-encryption operation in dependence of the computation identifier.

14. The apparatus according to any of claims 9 – 13, wherein the computation identifier identifies one of the following computation processes: addition, subtraction, multiplication, sign acquisition, comparison, equivalence test and variance.

15. A method comprising:

- receiving, from a data provider, a first ciphertext;
- performing a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier;
- obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- providing the second ciphertext to a first computation party.

16. The method according to claim 15, further comprising obtaining the computation identifier from the first computation party.

17. The method according to claim 15 or 16, further comprising participating in negotiating a shared secret with the first computation party.

18. The method according to claim 17, wherein the cryptographic re-encryption operation is performed in dependence of the computation identifier.

19. The method according to any of claims 15 - 18, wherein the second ciphertext is not decryptable solely by a secret key of the first computation party.

20. The method according to any of claims 15 – 19, further comprising obtaining a key pair comprising a public key of the apparatus and a secret key.

21. The method according to any of claims 15 – 20, wherein the computation identifier identifies at least one of the following computation processes: addition, subtraction, multiplication, sign acquisition, comparison, equivalence test and variance.

22. The method according to any of claims 15 – 21, further comprising obtaining a third ciphertext from the first ciphertext, providing the third ciphertext to a second computation party, and obtaining a fourth ciphertext from responses received from the first computation party and the second computation party, and obtaining an encrypted result of a computation process identified by the computation identifier.

23. A method comprising:

- determining, based on a message from a data requester, a computation identifier;
- transmitting a request to a data service provider, the request comprising the computation identifier;
- receiving, from the data service provider, a first ciphertext;
- obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- providing the second ciphertext to the data requester as a response to the message.

24. The method according to claim 23, further comprising checking an access policy before providing the request to the data service provider.

25. The method according to claim 23 or 24, further comprising participating in negotiating a shared secret with the data service provider.

26. The method according to claim 25, wherein the negotiating comprises a Diffie-Hellman negotiation.

27. The method according to any of claims 23 - 26, comprising performing the cryptographic re-encryption operation in dependence of the computation identifier.

28. The method according to any of claims 23 – 27, wherein the computation identifier identifies one of the following computation processes: addition, subtraction, multiplication, sign acquisition, comparison, equivalence test and variance.

29. A system comprising an apparatus according to claim 1, an apparatus according to claim 8, a data requester and a data provider.

30. An apparatus comprising:

- means for receiving, from a data provider, a first ciphertext;
- means for performing a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier;
- means for obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- means for providing the second ciphertext to a first computation party.

31. An apparatus comprising:

- means for obtaining a key pair comprising a public key of an apparatus and a secret key of the apparatus;
- means for determining, based on a message from a data requester, a computation identifier;
- means for transmitting a request to a data service provider, the request comprising the computation identifier and a public key of the data requester;
- means for receiving, from the data service provider, a first ciphertext;
- means for obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- means for providing the second ciphertext to the data requester as a response to the message.

32. A non-transitory computer readable medium having stored thereon a set of computer readable instructions that, when executed by at least one processor, cause an apparatus to at least:

- receive, from a data provider, a first ciphertext;
- perform a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier;
- obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- provide the second ciphertext to a first computation party.

33. A non-transitory computer readable medium having stored thereon a set of computer readable instructions that, when executed by at least one processor, cause an apparatus to at least:

- obtain a key pair comprising a public key of an apparatus and a secret key of the apparatus;
- determine, based on a message from a data requester, a computation identifier;
- transmit a request to a data service provider, the request comprising the computation identifier and a public key of the data requester;
- receive, from the data service provider, a first ciphertext;
- obtain a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation, and
- provide the second ciphertext to the data requester as a response to the message.

34. A computer program configured to cause a method in accordance with at least one of claims 15 – 28 to be performed.

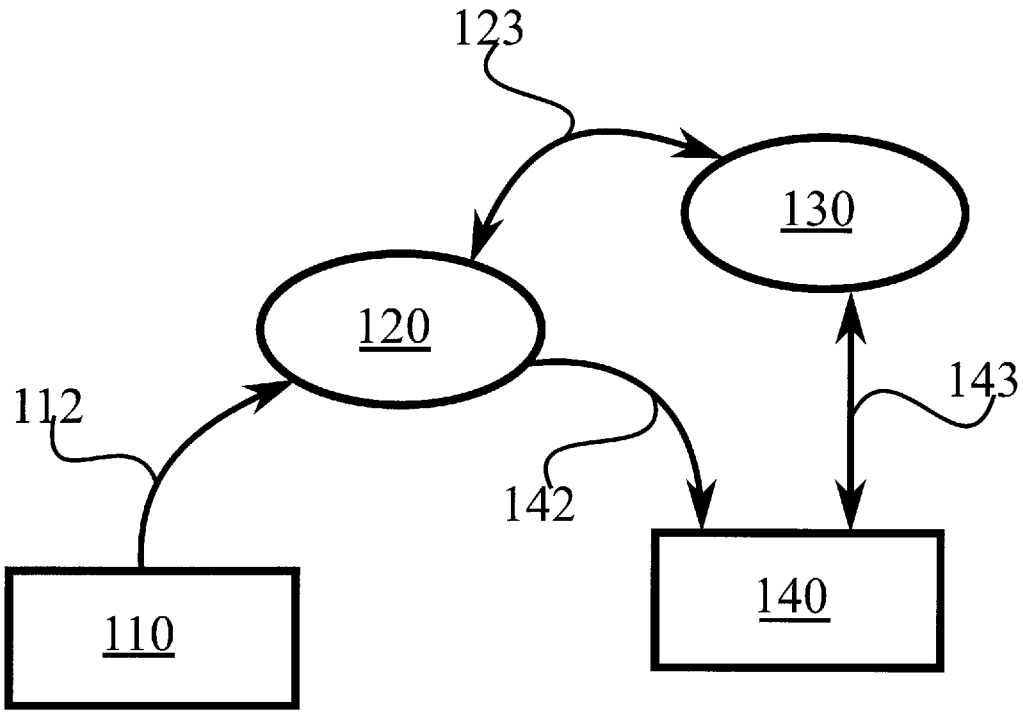

FIGURE 1

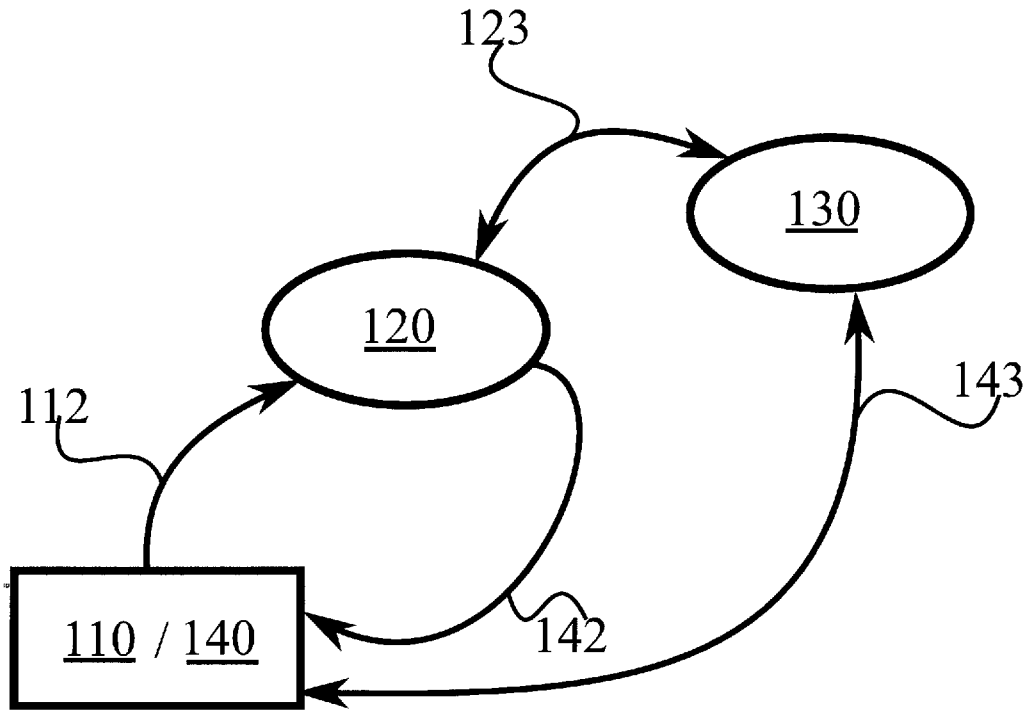

FIGURE 2

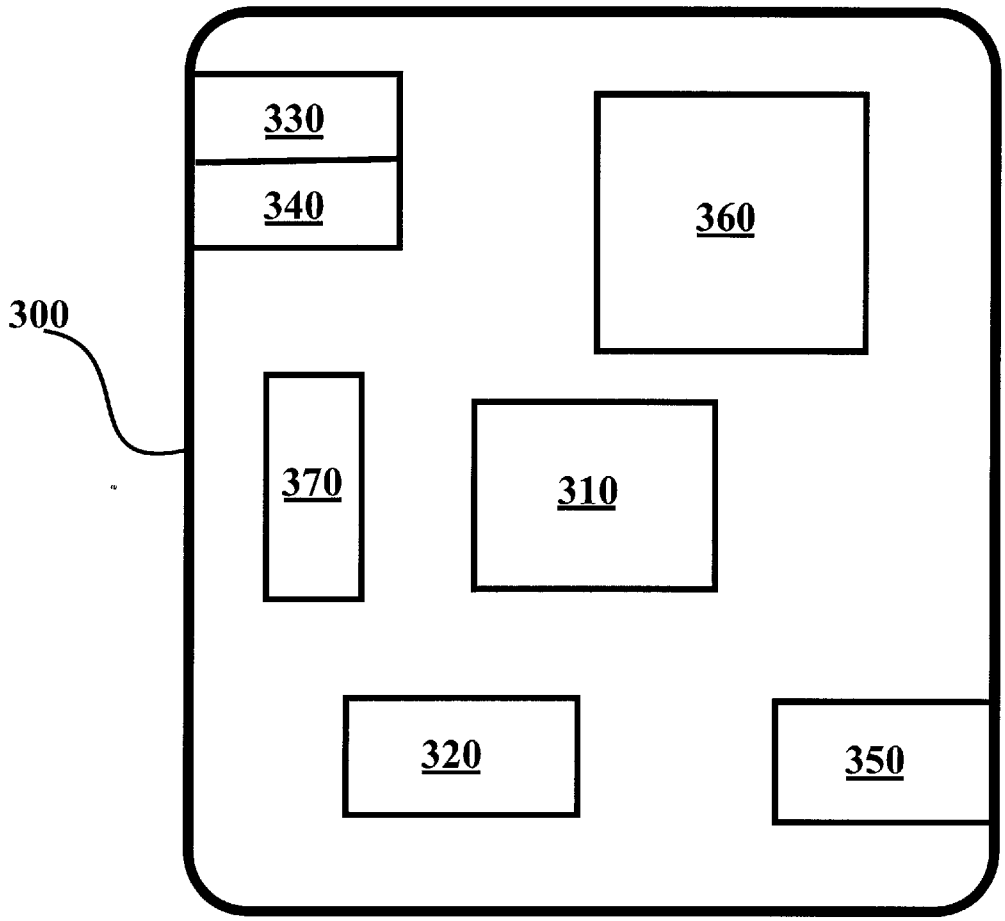

FIGURE 3

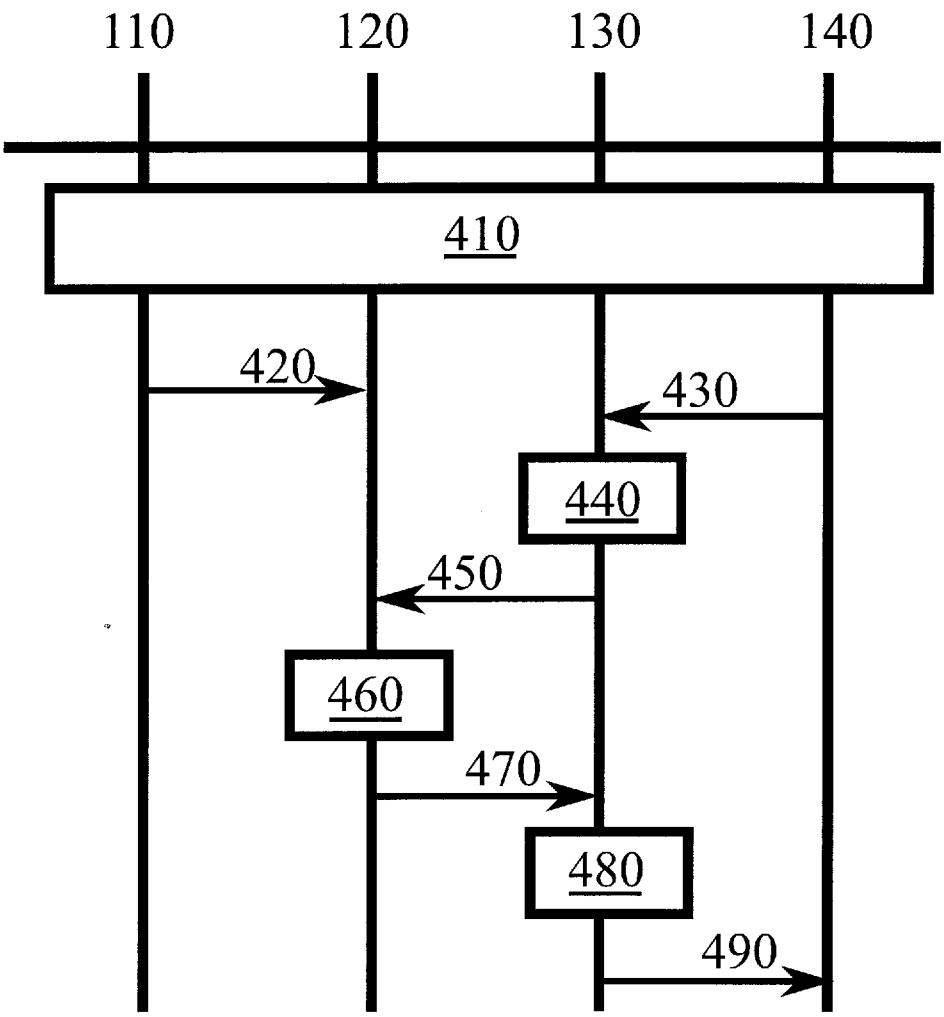

FIGURE 4

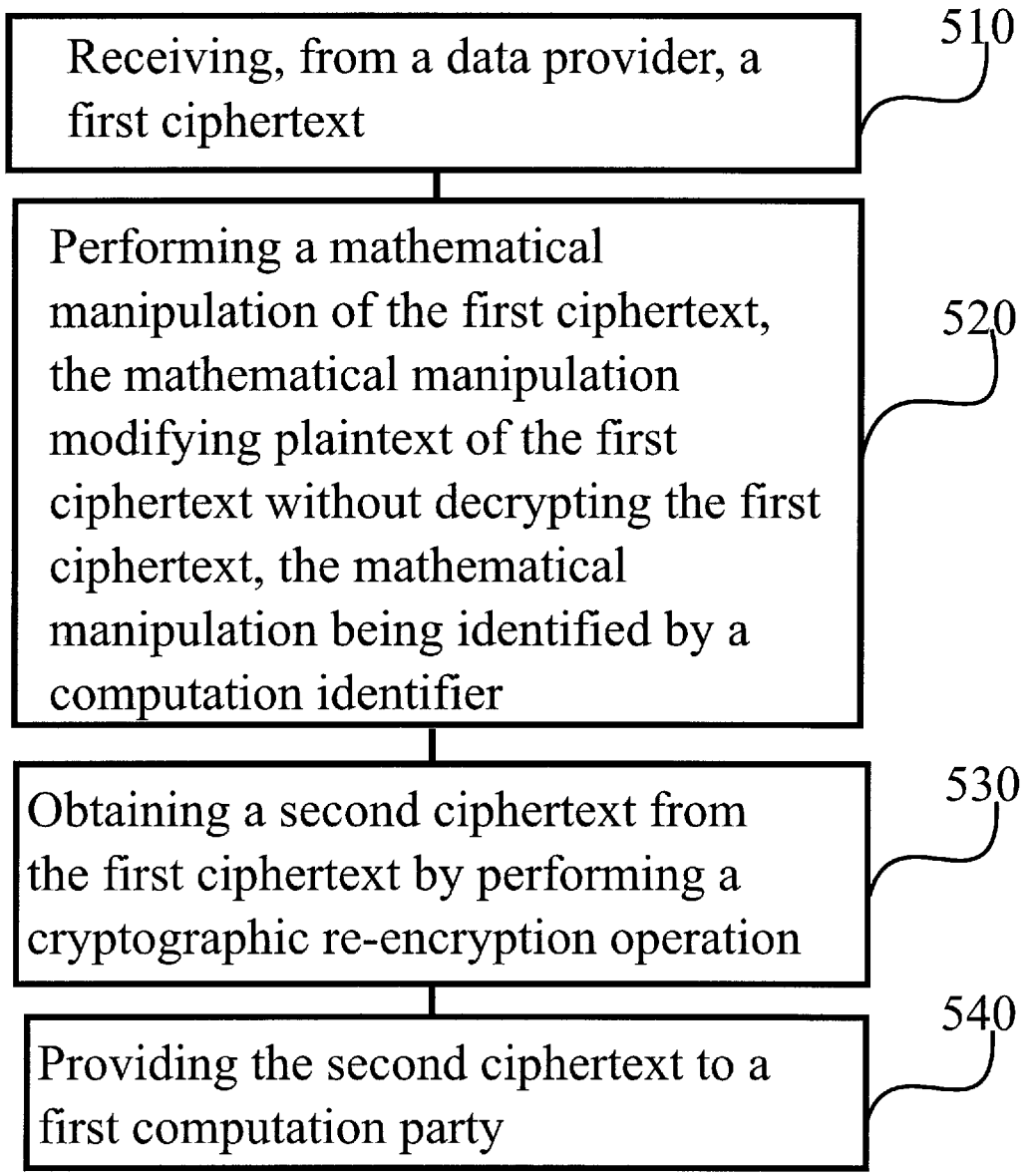
Receiving, from a data provider, a first ciphertext
510
Performing a mathematical manipulation of the first ciphertext, the mathematical manipulation modifying plaintext of the first ciphertext without decrypting the first ciphertext, the mathematical manipulation being identified by a computation identifier
520
Obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation
530
Providing the second ciphertext to a first computation party
540

FIGURE 5

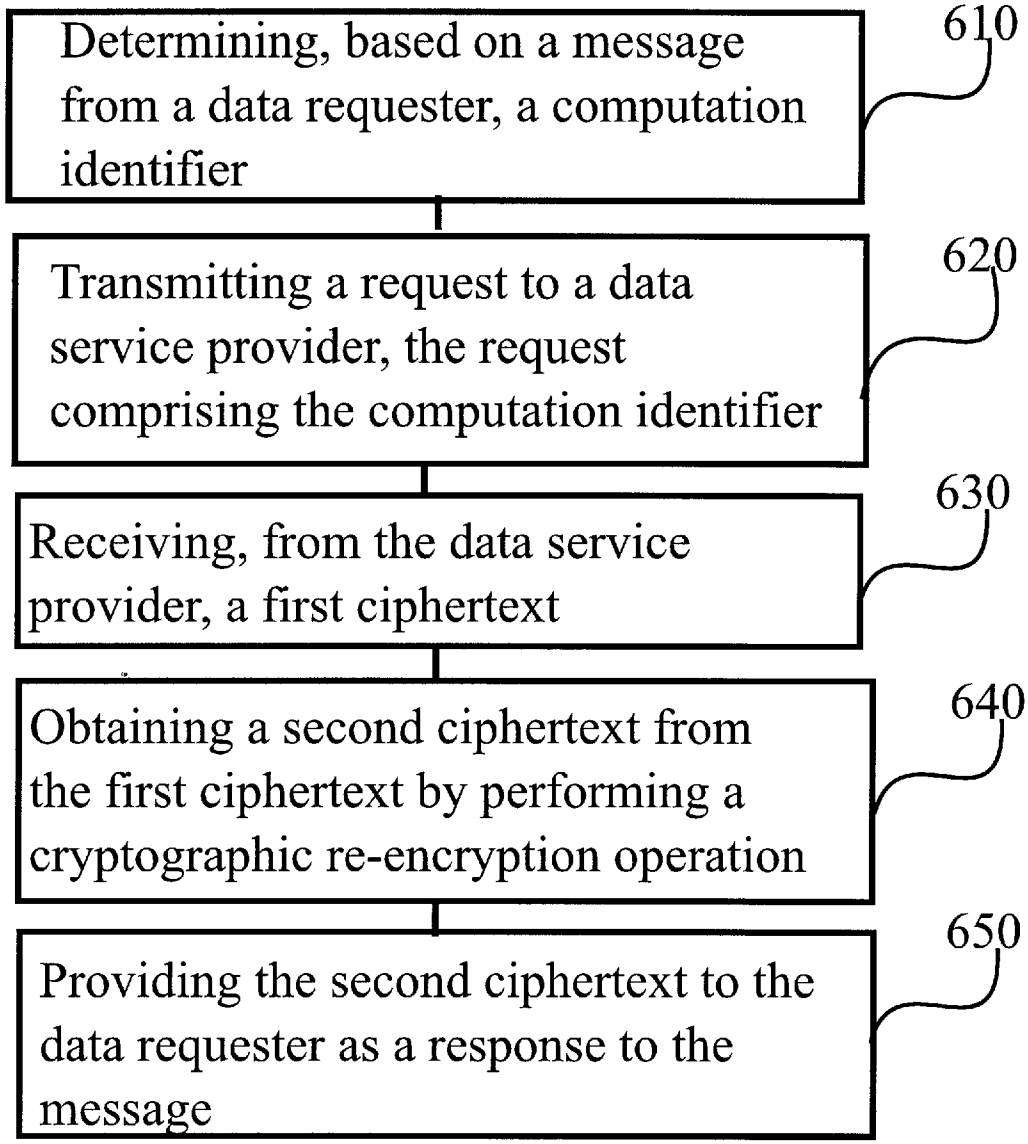
Determining, based on a message from a data requester, a computation identifier
610
Transmitting a request to a data service provider, the request comprising the computation identifier
620
Receiving, from the data service provider, a first ciphertext
630
Obtaining a second ciphertext from the first ciphertext by performing a cryptographic re-encryption operation
640
Providing the second ciphertext to the data requester as a response to the message
650

FIGURE 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2016/087876

A. CLASSIFICATION OF SUBJECT MATTER

H04L 9/14(2006.01)i

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; G06F

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

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

CNKI, CNPAT, EPODOC, WPI: ciphertext, cryptogram, cipher, cryptographic, key, plaintext, plain text, arithmetic, mathematical, computation identifier, algorithm identifier, re-encrypt, public key, secret key, private key

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2013246813 A1 (NEC CORPORATION) 19 September 2013 (2013-09-19) description, paragraphs [0093]-[0095], [0115], [0141]-[0151]	1-34
A	CN 1366751 A (SONY CORPORATION ET AL.) 28 August 2002 (2002-08-28) the whole document	1-34
A	CN 102271037 A (MICROSOFT CORPORATION) 07 December 2011 (2011-12-07) the whole document	1-34
A	US 2002186848 A1 (SHAIK, CHEMAN) 12 December 2002 (2002-12-12) the whole document	1-34

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
03 March 2017	**24 March 2017**
Name and mailing address of the ISA/CN **STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China**	Authorized officer **WEI,Ling**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)61648263**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2016/087876

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2013246813	A1	19 September 2013	KR	20130079607	A	10 July 2013
				JP	2014013582	A	23 January 2014
				JP	WO2013069776	A1	02 April 2015
				US	2015006908	A1	01 January 2015
				CN	103392178	A	13 November 2013
				EP	2778951	A1	17 September 2014
				WO	2013069776	A1	16 May 2013
CN	1366751	A	28 August 2002	BR	0104214	A	18 December 2001
				TW	514844	B	21 December 2002
				NZ	513863	A	28 September 2001
				EP	1164748	A1	19 December 2001
				CA	2361356	A1	02 August 2001
				AU	2883001	A	07 August 2001
				US	2003023847	A1	30 January 2003
				WO	0156224	A1	02 August 2001
				JP	2001211162	A	03 August 2001
				JP	2001209312	A	03 August 2001
				KR	20010108397	A	07 December 2001
				MX	PA01009230	A	01 December 2001
				RU	2001128666	A	20 July 2003
CN	102271037	A	07 December 2011	US	2011302398	A1	08 December 2011
US	2002186848	A1	12 December 2002	None			