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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 November 2016 (03.11.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/173646 A1

(51) International Patent Classification:
H04L 9/30 (2006.01)

(21) International Application Number:
PCT/EP2015/059376

(22) International Filing Date:
29 April 2015 (29.04.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **NEC EUROPE LTD.** [DE/DE]; Kurfuersten-Anlage 36, 69115 Heidelberg (DE).

(72) Inventor: **GAJEK, Sebastian**; Schloss-Wolfsbrunnenweg 15/III, 69118 Heidelberg (DE).

(74) Agent: **ULLRICH & NAUMANN**; Schneidmuehlstrasse 21, 69115 Heidelberg (DE).

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

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

Published:

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

WO 2016/173646 A1

(54) Title: METHOD AND SYSTEM FOR PROVIDING HOMOMORPHICALLY ENCRYPTED DATA ON A CLIENT

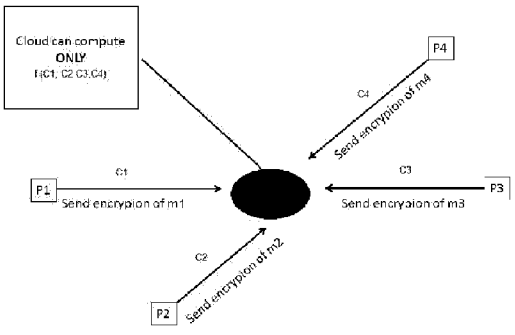

Fig. 1

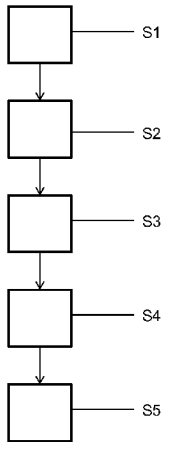

Fig. 2

(57) Abstract: The present invention relates to a method for providing encrypted data on a client, a cloud or the like, performed in a memory available to a computation device comprising the steps of a) Providing for each user a user specific encryption key for encrypting user- specific plaintext, b) Computing a common decryption key with a pre-determined function using the user specific encryption keys as input for said function, and wherein the common decryption key is computed based on at least two product groups of the same prime order, c) Encrypting each user-specific plaintext with the corresponding user-specific encryption key resulting in user-specific ciphertexts, d) Computing a common ciphertext with said function using the user-specific ciphertexts as input for said function, e) Providing the common ciphertext and the common decryption key for decryption, preferably to a user, wherein step c) is performed such that encryption is homomorphic in the user- specific plaintext as well in the user-specific encryption keys, wherein said function is a polysized function supporting poly-many additions and a single multiplication.

METHOD AND SYSTEM FOR PROVIDING HOMOMORPHICALLY ENCRYPTED DATA ON A CLIENT

The PCT-application PCT/EP2015/050432 is herein incorporated by reference.

The present invention relates to a method for providing encrypted data on a client, a cloud or the like performed in a memory available to a computation device.

The present invention further relates to a system for providing encrypted data.

Although applicable to any kind of encryption system, the present invention will be described with regard to an encryption system located in a cloud.

Cloud storage and services enable a user to offload storage of their data and associated computations on that data. Therefore companies and enterprises may avoid an expensive data center of their own and rely instead on cloud storage and corresponding computational services. However, data in the “cloud” raises serious privacy concerns, since the data in the cloud is for example accessible by the cloud provider itself. The enterprises have therefore to rely on the trustworthiness of the cloud provider and its security measures to avoid any misuse of the data.

Many cloud storage users employ in consequence encryption on user data to preserve the data privacy. One of the problems is however, that computation on encrypted data is very difficult to be efficiently performed without decrypting the encrypted data.

In US 2013/0097417 A1 a so-called somewhat homomorphic encryption scheme is provided. Collected data is encrypted using the somewhat homomorphic encryption based on a data provider’s private key. Afterwards the encrypted data is stored in a network-accessible storage. Then computations are performed on the encrypted data and the encrypted results from the computation are communicated to the data provider. The data provider decrypts the results using the data provider’s private key. These somewhat homomorphic encryption

schemes or l-leveled homomorphic encryption schemes are restricted in the number of ciphertext multiplication compositions or gates, i.e. the upper bound l for the number of multiplication gates is fixed in advance.

In contrast thereto fully-homomorphic encryption schemes – FHE – are encryption schemes supporting the homomorphic evaluation of any polysized function, which is described as an arithmetic circuit over poly-many additions and poly-many multiplication gates over encrypted data. However, fully-homomorphic evaluation is nowadays still a theoretic concept despite significant progress made recently.

Therefore, somewhat homomorphic encryption schemes are at present the schemes which can be practically used. Such partially homomorphic encryption systems are for example the El Gamal crypto system, the Paillier and the Goldwasser-Micali encryption system. These schemes are somewhat homomorphic in message plaintext.

In the non-patent literature of Craig Gentry and Shai Halevi "Implementing gentry's fully-homomorphic encryption scheme", Cryptology ePrint Archive, Report 2010/520, 2010, http://eprint.iacr.org/ shows an encryption scheme which is fully-homomophic in the message plaintext. Variants of these encryption schemes being fully-homomophic in the message plaintext are for example disclosed in US 2013/8565435 B2 or in US 2013/00170640 A1. Other variants which are based on the same mathematical structure, i.e. a lattice structure, are for example disclosed in the non-patent literature of Adriana Lopez-Alt, Eran Tromer and Vinod Vaikuntanathan "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption", Cryptology ePrint Archive, Report 2013/094, 2013, http://eprint.iacr.org/ which are additive homomorphic in the encryption key and fully-homomorphic in the message plaintext.

Another variant of the encryption scheme based on the mathematical structure lattice, is for example disclosed in the non-patent literature of Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan and Dhinakaran Vinayagamurthy "Fully key-homomorphic encryption, arithmetic circuit abe, and compact garbled circuits", Cryptology ePrint

Archive, Report 2014/356, 2014, http://eprint.iacr.org/, which have been shown to be fully-homomorphic in the encryption key but not in the message plaintext. These lattice-based constructions require much randomness sampled from the Gaussian distribution. Sampling randomness in particular from the Gaussian distribution cannot be effectively computed on low-level devices. Moreover the key and ciphertext sizes are very long, typically by a factor of 400 of the field size.

When these fully homomorphic encryption schemes are performed on the cloud, there are several problems arising: In some applications, for example one wishes to realize a supply chain where the cloud decrypts the outcome of the evaluation or a fraction thereof. One wants to make sure that the cloud learns no information about the input of the parties. The aforementioned fully homomorphic encryption schemes are useless here, because by definition there exists a single decryption key only. However, a cloud in possession of the decryption key will not only decrypt the evaluated ciphertext but may also decrypt the encrypted inputs of the parties and therefore a great security problem arises.

Another problem is that in many applications some verification mechanism has to be added to ensure a correct computation of the function, otherwise the parties do not have any guarantee of a correct evaluation. When using the aforementioned fully-homomorphic encryption schemes there is no guarantee that the cloud indeed computed the expected function. For this reason computationally expensive proofs of computation to verify the correct computation of the function are usually added.

It is therefore an objective to provide a method and a system for providing encrypted data which can be easily and efficiently implemented while being fast.

It is a further objective to provide a method and a system for providing encrypted data using short ciphertext sizes.

It is an even further objective to provide a method and system for providing encrypted data with which an easy verification for the proof of computation is enabled.

In at least one embodiment a method for providing encrypted data on a client, a cloud or the like, performed in a memory available to a computation device, is defined comprising the steps of

a) Providing for each user a user specific encryption key for encrypting user-specific plaintext,

b) Computing a common decryption key with a pre-determined function using the user specific encryption keys as input for said function, and wherein the common decryption key is computed based on at least two product groups of the same prime order,

c) Encrypting each user-specific plaintext with the corresponding user-specific encryption key resulting in user-specific ciphertexts,

d) Computing a common ciphertext with said function using the user-specific ciphertexts as input for said function,

e) Providing the common ciphertext and the common decryption key for decryption,

wherein step c) is performed such that encryption is homomorphic in the user-specific plaintext as well in the user-specific encryption keys and

wherein said function is a polysized function supporting poly-many additions and a single multiplication.

In at least one embodiment a system for providing encrypted data is defined comprising one or more computation devices comprising

User key provision means adapted to provide for each user a user specific encryption key for encryption of user-specific plaintext,

Decryption key generation means adapted to compute a common decryption key with a pre-determined function using the user specific encryption keys as input for said function, and wherein the common decryption key is computed based on at least two product groups of the same prime order,

Encryption means adapted to encrypt each user-specific plaintext with the corresponding user-specific encryption key resulting in user-specific ciphertexts,

Ciphertext means adapted to compute a common ciphertext with said function using the user-specific ciphertexts as input for said function,

wherein the encryption by the encryption means is performed such that encryption is homomorphic in the user-specific plaintext as well in the user-specific encryption keys and
wherein said function is a polysized function supporting poly-many additions and a single multiplication is defined.

At least one embodiment herein is homomorphic in the encryption key and in the message plaintext.

At least one embodiment herein enables an evaluation of level-1 circuits/functions supporting a single multiplication in poly-many additions. At least one embodiment enables the evaluation of Boolean circuits, i.e. formulas comprising “and” and “or” operations.

At least one embodiment herein enables sizes of ciphertext which are short compared to conventional methods and systems.

At least one embodiment provides the advantage of being very efficient and being easily to implement.

At least one embodiment enables the use of type-3 groups, i.e, groups pf prime order with an asymmetric pairing.

The term “homomorphic in the message plaintext” is to be understood in particular in the following way: If for any public key k and messages (m_1, m_2) it holds that $Enc(k, m_1) \cdot Enc(k, m_2) = Enc(k, m_1 \cdot m_2)$, where the operator “$\cdot$” is a operand allowing to “add” or “multiply” the message plaintext and Enc() is an encryption procedure or routine.

The term “homomorphic in the key” or “homomorphic in the encryption key” or similar expressions are to be understood in the following way: A public-key encryption scheme is deemed homomorphic in the encryption key, if in addition for any pair of encryption keys (k_1, k_2) and message plaintexts (m_1, m_2) it holds that $Enc(k_1, m_2) . Enc(k_2, m_2) = Enc(k_1 \cdot k_2, m_1 \cdot m_2)$.

The term “compact” in connection with encryption key or ciphertext means that the ciphertext (resp. encryption key) size after the homomorphic evaluation of ciphertexts (resp. encryption keys) does not increase as a function of the circuit.

User key provisions means include, but are not limited to, an application adapted to provide for each user a user specific encryption key for encryption of user-specific plaintext running in a memory available to a computer, a microprocessor, a single, dual, quad or octa-core processor or processors or the like or a processor, or the like with a memory. Said application or processor may have one or more interfaces, ports or the like for communication with other devices, entities, ports, interfaces or the like.

Decryption key generation means include, but are not limited to, an application adapted to compute a common decryption key with a pre-determined function using the user specific encryption keys as input for said function running in a memory available to a computer, a microprocessor, a single, dual, quad or octa-core processor or processors or the like or a processor, or the like with a memory. Said application or processor may have one or more interfaces, ports or the like for communication with other devices, entities, ports, interfaces or the like.

Encryption means include, but are not limited to, an application adapted to encrypt each user-specific plaintext with the corresponding user-specific encryption key running in a memory available to a computer, a microprocessor, a single, dual, quad or octa-core processor or processors or the like or a processor, or the like with a memory. Said application or processor may have one or more interfaces, ports or the like for communication with other devices, entities, ports, interfaces or the like.

Ciphertext means include, but are not limited to, an application adapted to compute a common ciphertext with said function using the user-specific ciphertexts as input for said function running in a memory available to a computer, a microprocessor, a single, dual, quad or octa-core processor or processors or the like or a processor, or the like with a memory. Said application or processor may

have one or more interfaces, ports or the like for communication with other devices, entities, ports, interfaces or the like.

Further features, advantages and further embodiments are described or may become apparent in the following.

A user may apply at the common decryption key to decrypt the common ciphertext. This may enable a user to perform decryption of the common ciphertext.

If a key other than the common decryption key for decrypting the common ciphertext is used a random value may be provided as decryption result. When applying the "right" decryption key for the common ciphertext evaluated under the function, then the correctness of the performed steps guarantees that the user will decrypt f (m_1, ... m_2) ← DEC (K, C). On the other hand security is safeguarded such that a curious or malicious party holding a key other than the common decryption key will decrypt a random value with overwhelming probability. For instance when applying the "right" encoding to a user-specific plaintext, e.g. by encoding the first half with "1"'s, each user can verify the correct computation of the function in question. An evaluation of the function other than said function decrypts to a message which is uniformly distributed. Even when a group of n-1 malicious or corrupted users combine their user-specific encryption keys these users are not able to decrypt the common ciphertext as long as a single user is honest or uncorrupted respectively.

A master secret key may be computed and may be used together with said function and the user-specific encryption keys to compute said common decryption key prior to computing the common decryption key. A master secret key may enable in an easy way to provide the further common decryption key, for example based on the function and the corresponding user-specific encryption keys.

The master secret key may comprise random seed information associated with identification information via a pseudo-random function defining elements in different user-specific ciphertexts. This may enable a reduction of the size of the

master secret key, for example by deriving tags through a pseudorandom function. Thus, storage space is saved.

When a user-specific encryption key is evaluated under said function the resulting evaluated user-specific encryption key may be compact in size. This may enable that any homomorphic key evaluation of the level-one circuit results in constant key sizes of two group elements.

When the user-specific ciphertext is evaluated under said function, the resulting evaluated user-specific ciphertext may be compact in size. This may enable that any homomorphic ciphertext evaluation of the level-1 circuit results in constant ciphertext sizes of two group elements.

Said function may be provided out of the set of polysized affine functions. This may enable an efficient computation of the function when being selected as a polysized affine function.

When computing the common decryption key an input vector comprising the user-specific encryption keys may be divided into two parts, each part having the same length and indicating different user-specific ciphertext. This may enable a fast and efficient computation of the common decryption key.

Said resulting user-specific encryption key may be constant. This may enable the evaluation of said keys independent of said function f.

Said resulting evaluated user-specific ciphertext may be constant. This may enable the evaluation of said keys independent of said function f.

Said master secret key may be computed based on said elements of said at least two product groups of said same prime order. This may enable to derive from the master secret key the decryption key based on said two product groups.

The user-specific ciphertext encrypted under a user-specific encryption key may be based on four parts, wherein the first and last part may be each computed

based on a product of different generators of one of said groups, wherein one said generator may be computed based on the user-specific plaintext. This may enable to compute the user-specific ciphertext efficiently while being secure.

The two middle parts of said four parts may be using products of generators, wherein said generators are computed based on the user-specific plaintext and different random information. This may enable to compute the user-specific ciphertext even more efficiently while being secure.

There are several ways how to design and further develop the teaching of the embodiments in an advantageous way. To this end it is to be referred to the patent claims on the one hand and to the above mentioned and following explanation of embodiments by way of example, illustrated by the figure on the other hand. In connection with the explanation of embodiments by the aid of the figure, generally embodiments and further developments of the teaching will be explained. In the drawings

Fig. 1 shows a part of a method according to an embodiment;

Fig. 2 shows steps of a method according to an embodiment;

Fig. 3 shows a system according to an embodiment; and

Fig. 4 shows a method according to an embodiment.

Fig. 1 shows a part of a method according to an embodiment.

In Fig. 1 an example scenario for a method according to an embodiment is shown. Different users P1, P2, P3, P4 each use their own user-specific encryption key k_1, k_2, k_3, k_4, to encrypt the user-specific plaintext m_1, m_2, m_3, m_4. Each user P1, P2, P3, P4 sends the encryption of its user-specific plaintext m_1, m_2, m_3, m_4, i.e. the corresponding ciphertext C1, C2, C3, C4 to the cloud, so that the cloud receives the encryptions of the messages m_1, m_2, m_3, m_4 encrypted under the

corresponding keys k_1, k_2, k_3, k_4. The cloud evaluates then not only the encrypted function f but also may decrypt the output f (C1, C2, C3, C4) provided if it is in possession of the specific key, i.e. the common decryption key $K = f(k_1, \ldots, k_n)$. This common decryption key K is useless for other operations than the decryption of the output f (C1, C2, C3, C4). Vice versa other keys other than $f(k_1, \ldots, k_n)$ are useless of the decryption of f (C1, C2, C3, C4). The number of users P1, P2, P3, P4 may be arbitrarily large and bounded by a polynomial in a security parameter, wherein the security parameter is e.g. used as shown in Fig. 4 in the Setup procedure.

Fig. 2 shows steps of a method according to an embodiment.

In Fig. 2 steps of a method according to an embodiment are shown.

In a first step S1 for each user a user-specific encryption key for encrypting user-specific plaintext is provided.

In a second step S2 a common decryption key is computed with a pre-determined function using the user-specific encryption keys as input for said function, and wherein the common decryption key is computed based on elements of at least two product groups of the same prime order.

In a third step S3 each user-specific plaintext is encrypted with a corresponding user-specific encryption key resulting in user-specific ciphertexts.

In a fourth step S4 a common ciphertext is computed with said function using the user-specific ciphertexts as input for said function and in a fifth step S5 the common ciphertext is provided and the common decryption key for decryption.

The third step S3 is performed such that the encryption is homomorphic in the user-specific plaintext as well as in the user-specific encryption keys and said function is a polysized function supporting poly-many additions and a single multiplication.

Fig. 3 shows a system according to an embodiment.

In Fig. 3 the system S comprises user key provision means 1 adapted to provide for each user a user-specific encryption key for encryption of user-specific plaintext.

The system S further comprises decryption key generation means 2 adapted to compute common encryption key with a pre-determined function using the user-specific encryption keys as input for said function, and wherein the common decryption key is computed based on elements of at least two product groups of the same prime order.

The system S further comprises encryption means 3 adapted to encrypt each user specific plaintext with the corresponding user-specific and encryption key resulting in user-specific ciphertexts.

The system S even further comprises ciphertext means 4 adapted to compute a common ciphertext with said function using the user-specific ciphertexts as input for said function and decryption means 5 may be provided and adapted to provide the common ciphertext and the common decryption key for decryption. The encryption by the encryption means 3 is performed such that the encryption is homomorphic in the user-specific plaintext as well as in the user-specific encryption key and wherein said function is a polysized function supporting poly-many additions and a single multiplication.

Fig. 4 shows a method according to an embodiment.

In Fig. 4 steps of a construction of a level-1 key in message homomorphic symmetric encryption system with short ciphertexts are shown.

In a first step a setup-procedure Setup(.) is performed:

Setup(1^λ): On input the security parameter 1^λ $(p, G_1, G_2, G_T, e) \leftarrow GG(1^\lambda)$ is computed and random generators $g_1, g_2 \leftarrow_R G_1$ and $h_1, h_2 \leftarrow_R G_2$ are chosen.

Further at random x, y $\leftarrow_R \mathbb{Z}_p$ is chosen. The message space is e.g. set to X = {0, 100} and the tag space to T = $\mathbb{Z}_p$. It is assumed that every index k = (i, j) $\in$ K is associated with a random tag $t_{i,j} \in T$, whose purpose is to identify the i^{th} sequence of n ciphertexts, where j denotes the j^{th} element. n is supposed to be even such that l = ceil(n/2), wherein ceil(.) is the ceiling function giving the smallest integer greater than or equal to the input value.

The procedure outputs the master key pair

$$MSK = x, y, T \text{ and } PP = X, K, p, g_1, g_2, h_1, h_2, g^{x}_1, g^{y}_2, h^{-y}_1, h^{-x}_2$$

The size of the master secret key can be reduced by deriving the tags through a pseudorandom function.

In a second step a key generation procedure KeyGen(.) is performed:

KeyGen (MSK, f, k = (k_1, ..., k_n): The first half of k is denoted as the tuple ((l, 1), ... , (l, l)) and the second half as ((r, 1), ... , (r, l)). The following secret key $SK_{f, k}$ is returned:

$$e(g_2, h_2)^{(y-x)\sum_{j=1}^{l}(t_{l,j} \cdot t_{r,j})}$$

If f is the unary identity function for a key $k_{i, j} \in K$ associated with tag $t \in T$, the tuple $SK_{k, j} = (g_2^{t}, h_2^{-tx}, h_2^{ty})$ is returned.

In a third step an encryption procedure Enc(.) for the message plaintext is performed:

Enc ($SK_{ki,j}$, m): To encrypt a message $m \in X$ under key $k_{i, j}$ associated with the one-time tag $t \in T$, the ciphertext $CT_{i, j}$ is set as

$$A = g_1^{m} \cdot g_2^{t}, B = g_1^{xm} \cdot g_2^{yt}, C = h_1^{-ym} \cdot h_2^{-xt}, D = h_1^{m} \cdot h_2^{t}$$

In a forth step an evaluation $Eval_{Ct}$ of the ciphertext is performed:

$Eval_{Ct}(f, CT_{i,1}, ..., CT_{i,n})$: To evaluate the function f on a tuple of n ciphertexts ($CT_{i,1}$, ..., $CT_{i,n}$), the left half is denoted as ($CT_{l,1}$, ..., $CT_{l,l}$) and the right half is denoted as ($CT_{r,1}$, ... , $CT_{r,l}$). The procedure parses from every ciphertext pair ($CT_{l,j}$, $CT_{r,j}$) the tuple ($A_{l,j}$, $B_{l,j}$, $C_{l,j}$, $D_{l,j}$ $A_{r,j}$,$B_{r,j}$, $C_{r,j}$, $D_{r,j}$) and computes the addition and multiplication operations as follows:

• Multiplication: The procedure computes the component-wise bilinear map

$$e(A_{l,j}, C_{r,j}) \cdot e(B_{l,j}, D_{r,j})$$

$$= e(g_1^{m_{l,j}} \cdot g_2^{t_{l,j}}, h_1^{-y \cdot m_{r,j}} \cdot h_2^{-x \cdot t_{r,j}}) \cdot e(g_1^{-x \cdot m_{l,j}} \cdot g_2^{y \cdot t_{l,j}}, h_1^{m_{r,j}} \cdot h_2^{t_{r,j}})$$
$$= e(g_1, h_1)^{(x-y) \cdot m_{l,j} \cdot m_{r,j}} \cdot e(g_2, h_2)^{(y-x) \cdot t_{l,j} \cdot t_{r,j}}$$
$$= A^*$$

and returns the ciphertext C T = A^*.

• Addition in G_T (after Multiplication): The procedure computes the component-wise product

$$A_1^* \cdot A_2^*$$
$$= e(g_1, h_1)^{(x-y) \cdot m_1} \cdot e(g_2, h_2)^{(y-x) \cdot t_1} \cdot e(g_1, h_1)^{(x-y) \cdot m_2} \cdot e(g_2, h_2)^{(y-x) \cdot t_2}$$
$$= e(g_1, h_1)^{(x-y) \cdot m_1 \cdot m_2} \cdot e(g_2, h_2)^{(y-x) \cdot t_1 \cdot t_2}$$
$$= A^+$$

and returns the ciphertext CT = A^+.

• Addition in $G_1^2 \times G_2^2$ (before multiplication): The procedure one more time computes then component-wise product

$$A_{l,j} \cdot A_{r,j}, B_{l,j} \cdot B_{r,j}, C_{l,j} \cdot C_{r,j}, D_{l,j} \cdot D_{r,j}$$

$$= g_1^{m_{l,j}} \cdot g_2^{t_{l,j}} \cdot g_1^{m_{r,j}} \cdot g_2^{t_{r,j}}, g_1^{x \cdot m_{l,j}} \cdot g_2^{y \cdot t_{l,j}} \cdot g_1^{x \cdot m_{r,j}} \cdot g_2^{y \cdot t_{r,j}},$$
$$h_1^{-y \cdot m_{l,j}} \cdot h_2^{-x \cdot t_{l,j}} \cdot h_1^{-y \cdot m_{r,j}} \cdot h_2^{-x \cdot t_{r,j}}, h_1^{m_{l,j}} \cdot h_2^{t_{l,j}} \cdot h_1^{m_{r,j}} \cdot h_2^{t_{r,j}}$$

$= \quad g_1^{m_{l,j}+m_{r,j}} \cdot g_2^{t_{l,j}+t_{r,j}}, g_1^{x\cdot(m_{l,j}+m_{r,j})} \cdot g_2^{y(\cdot t_{l,j}+t_{r,j})}, h_1^{-y\cdot(m_{l,j}+m_{r,j})} \cdot h_2^{-x\cdot(t_{l,j}+t_{r,j})},$
$h_1^{m_{l,j}+m_{r,j}} \cdot h_2^{t_{l,j}+t_{r,j}}$

$= \quad A^{++}, B^{++}, C^{++}, D^{++}$

and returns the ciphertext CT = (A^{++}, B^{++}, C^{++}, D^{++}).

In a fifth step an evaluation procedure EvalKey of the encryption keys is performed:

EvalKey(f, $SK_{ki,1}$, …, $SK_{ki,n}$: To evaluate the function f on a tuple of n keys ($SK_{ki,1}$, ..., $SK_{ki,n}$), the left half is e.g. denoted as ($SK_{kl,1}$, …, $SK_{kl,l}$) and the right half as ($SK_{kr,1}$, …, $SK_{kr,l}$). The procedure parses every key pair ($SK_{kl,j}$, $SK_{kr,j}$) as ($A_{l,j}$, $B_{l,j}$, $C_{l,j}$, $A_{r,j}$, $B_{r,j}$, $C_{r,j}$) $\in (G_1 \times G_2)^2$ and computes the addition and multiplication operation as follows:

• Multiplication Gate: The procedure computes

$e(A_{l,j}, B_{r,j}) \cdot e(A_{r,j}, C_{l,j})$

$= \quad e(g_2, h_2)^{-x\cdot t_{l,j}\cdot t_{r,j}} \cdot e(g_2, h_2)^{y\cdot t_{l,j}\cdot t_{r,j}}$

$= \quad A$

and returns the key SK = A.

• Addition (after multiplication): Multiplication of every element in G_{GT}.

• Addition (before multiplication): Multiplication of every element $G_1 \times G_2^2$.

In a sixth step a decryption procedure Dec may be performed:

Dec(SK, CT): To decrypt a ciphertext CT with secret key SK, two cases are distinguished:

• Before multiplication: If CT = (CT_1, CT_2, CT_3, CT_4) $\in G_1^2 \times G_2^2$, then

CT_1/SK_1

is computed, where $SK = (SK_1, SK_2, SK_3) \in G_1 \times G_2^2$.

• After multiplication: If $C\ T \in G_T$, then

CT/SK

is computed where $SK \in G_T$.

0 is returned if the output equals $e(g_1, h_1)^0 = 1$, and otherwise the discrete log to the base $e(g_1, h_1)^{x-y}$ is computed.

Inter alia for the above mentioned steps the background is the following:

Assuming that G_1, G_2, and G_T are three multiplicatively written groups of prime order p. Moreover, g_1 and g_2 are assumed to be generators of G_1, and G_2 resp., such that $(g_1) = G_1$ and $(g_2) = G_2$. Further assuming $e : G_1 \times G_2 \rightarrow G_T$ is a bilinear map with the following properties:

Bilinearity: For $u \in G_1$, $v \in G_2$, $a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$.

Non-Degerancy: $e(g_1, g_2) \neq 1$.

Symmetry: For $u, v \in G$, $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u^b, v^a)$.

G is a bilinear group if all group operations and the bilinear map are efficiently computable.

In summary at least one embodiment uses product groups and pairing composition to emulate an orthogonality: In detail product groups $G_1 = (g_1, g_1^x) \in G_1$, $G_2 = (g_2, g_2^y) \in G_1$, $H_1 = (h_1^{-y}, h_1) \in G_2$ and $H_2(h_2^{-x}, h_2) \in G_2$ are used.

It can be verified that $G = (G_1, G_2)$ and $H = (H_1, H_2)$ form an orthogonal bases.

At least one embodiment may enable at least one of the following advantages:

- It is both homomorphic in key and message plaintext.
- It supports the evaluation of level-1 circuits supporting a single multiplication and poly-many additions, also known as inner products. This circuit already allows the evaluation of Boolean circuits.
- It is efficient. The construction works over type-3 groups G of prime order p. Thus, the scheme can be efficiently implemented over elliptic curves.
- It has compact ciphertexts. Any multiplicative ciphertext evaluation of the level-1 circuit results a single ciphertext of one group element in G_T (where G_T is the target group of the pairing).
- It has compact keys. Any homomorphic key evaluation of the level-1 circuit results in a decryption key of a single group element in G_T.

Inter alia at least one embodiment may have one of the further advantages apart from being both homomorphic in key and message:

1. It can be efficiently implemented in type-3 groups (on elliptic curves) of prime order enabling a fast implementation.

2. Key and ciphertext sizes are constant. They are independent of the evaluation function f.

3. It requires less randomness. Moreover, it suffices to sample from the uniform distribution for which efficient implementations exists.

In the following examples for applications of embodiments of the present invention are shown. Of course the embodiments are not limited to this list:

1. Searchable Encryption: Supposing a file tagged with a vector y should be searched. Assuming a vector x satisfies the search query, if the inner product <xy> = 0. The database should do the orthogonality check while learning neither the (encrypted) search query x nor the (encrypted) database y.

2. Smart Metering: Supposing that every household i has a secret key SK_i, and the center receives encrypted metering data from each household. The center should compute a function, say the average consumption, without learning each household's value.

3. Machine Learning. Many machine learning algorithms and procedures require the computation of linear functions, (e.g. average, variance, distance.) Embodiments described herein can realize the computation in a private way.

4. Image processing. Many image processing algorithms and procedures require the compution of affine transformation over matrix groups (e.g. FFT, wavelets, etc.). Embodiments described herein can realize the computation in a private way.

Many modifications and other embodiments of the invention set forth herein will come to mind to the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

Claims

1. A method for providing encrypted data on a client, a cloud or the like, performed in a memory available to a computation device
comprising the steps of
a) Providing (S1) for each user (P1, P2, P3, P4) a user specific encryption key (k_1, k_2, k_3, k_4) for encrypting user-specific plaintext (m_1, m_2, m_3, m_4),
b) Computing (S2) a common decryption key (SK) with a pre-determined function (f) using the user specific encryption keys (k_1, k_2, k_3, k_4) as input for said function (f), and wherein the common decryption key is computed based on at least two product groups of the same prime order,
c) Encrypting (S3) each user-specific plaintext (m_1, m_2, m_3, m_4) with the corresponding user-specific encryption key (k_1, k_2, k_3, k_4) resulting in user-specific ciphertexts (c_1, c_2, c_3, c_4),
d) Computing (S4) a common ciphertext (CT) with said function (f) using the user-specific ciphertexts (c_1, c_2, c_3, c_4) as input for said function (f),
e) Providing (S5) the common ciphertext (CT) and the common decryption key (SK) for decryption, preferably to a user (P1, P2, P3, P4),
wherein step c) is performed such that encryption is homomorphic in the user-specific plaintext (m_1, m_2, m_3, m_4) as well in the user-specific encryption keys (k_1, k_2, k_3, k_4), and
wherein said function (f) is a polysized function supporting poly-many additions and a single multiplication.

2. The method according to claim 1, wherein a user applies the common decryption key (SK) to decrypt the common ciphertext (CT).

3. The method according to one of the claims 1-2, wherein, if a key other than the common decryption key (SK) for decrypting the common ciphertext (CT) is used, a random value is provided as decryption result.

4. The method according to one of the claims 1-3, wherein prior to computing the common decryption key (SK) a master secret key (MSK) is computed and

used together with said function (f) and the user-specific encryption keys (k_1, k_2, k_3, k_4) to compute said common decryption key (SK).

5. The method according to claim 4, wherein the master secret key (MSK) comprises random seed information associated with identification information via a pseudorandom function defining elements in different user-specific ciphertexts (c_1, c_2, c_3, c_4).

6. The method according to one of the claims 1-5, wherein when a user-specific encryption key (k_1, k_2, k_3, k_4) is evaluated under said function (f), the resulting evaluated user-specific encryption key (k_1, k_2, k_3, k_4) is compact in size.

7. The method according to one of the claims 1-6, wherein when the user-specific ciphertext (c_1, c_2, c_3, c_4) is evaluated under said function (f), the resulting evaluated user-specific ciphertext (c_1, c_2, c_3, c_4) is compact in size.

8. The method according to one of the claims 1-7, wherein said function (f) is provided out of the set of polysized affine functions.

9. The method according to one of the claims 1-8, wherein when computing the common decryption key (SK) an input vector comprising the user-specific encryption keys (k_1, k_2, k_3, k_4) is divided into two parts, each part having the same length and indicating different user-specific ciphertexts (c_1, c_2, c_3, c_4).

10. The method according to claim 6, wherein said resulting user-specific encryption key is constant.

11. The method according to claim 7, wherein said resulting evaluated user-specific cipertext is constant.

12. The method according to claim 4, wherein said master secret key is computed based on elements of said at least two product groups of said same prime order.

13. The method according to one of the claims 1-12, wherein the user-specific ciphertext encrypted under a user-specific encryption key is based on four parts, wherein the first and last part are each computed based on a product of different generators of one of said groups, wherein one of said generators is computed based on the user-specific plaintext.

14. The method according to claim 13, wherein the two middle parts of said four parts are using different products of generators, wherein said generators are computed based on the user-specific plaintexts and different random information.

15. A system (S) for providing encrypted data for performing with a method according to one of the claims 1-14,
comprising one or more computation devices comprising
User key provision means (1) adapted to provide for each user (P1, P2, P3, P4) a user specific encryption key (k_1, k_2, k_3, k_4) for encryption of user-specific plaintext (m_1, m_2, m_3, m_4),
Decryption key generation means (2) adapted to compute a common decryption key (SK) with a pre-determined function (f) using the user specific encryption keys (k_1, k_2, k_3, k_4) as input for said function (f), and wherein the common decryption key is computed based on at least two product groups of the same prime order,
Encryption means (3) adapted to encrypt each user-specific plaintext (m_1, m_2, m_3, m_4) with the corresponding user-specific encryption key (k_1, k_2, k_3, k_4) resulting in user-specific ciphertexts (c_1, c_2, c_3, c_4),
Ciphertext means (4) adapted to compute a common ciphertext (CT) with said function using the user-specific ciphertexts (c_1, c_2, c_3, c_4) as input for said function, and preferably
Decryption means (5) adapted to provide the common ciphertext (CT) and the common decryption key (SK) for decryption, preferably to a user (P1, P2, P3, P4),
wherein the encryption by the encryption means is performed such that encryption is homomorphic in the user-specific plaintext (m_1, m_2, m_3, m_4) as well in the user-specific encryption keys (k_1, k_2, k_3, k_4) and
wherein said function is a polysized function (f) supporting poly-many additions and a single multiplication.

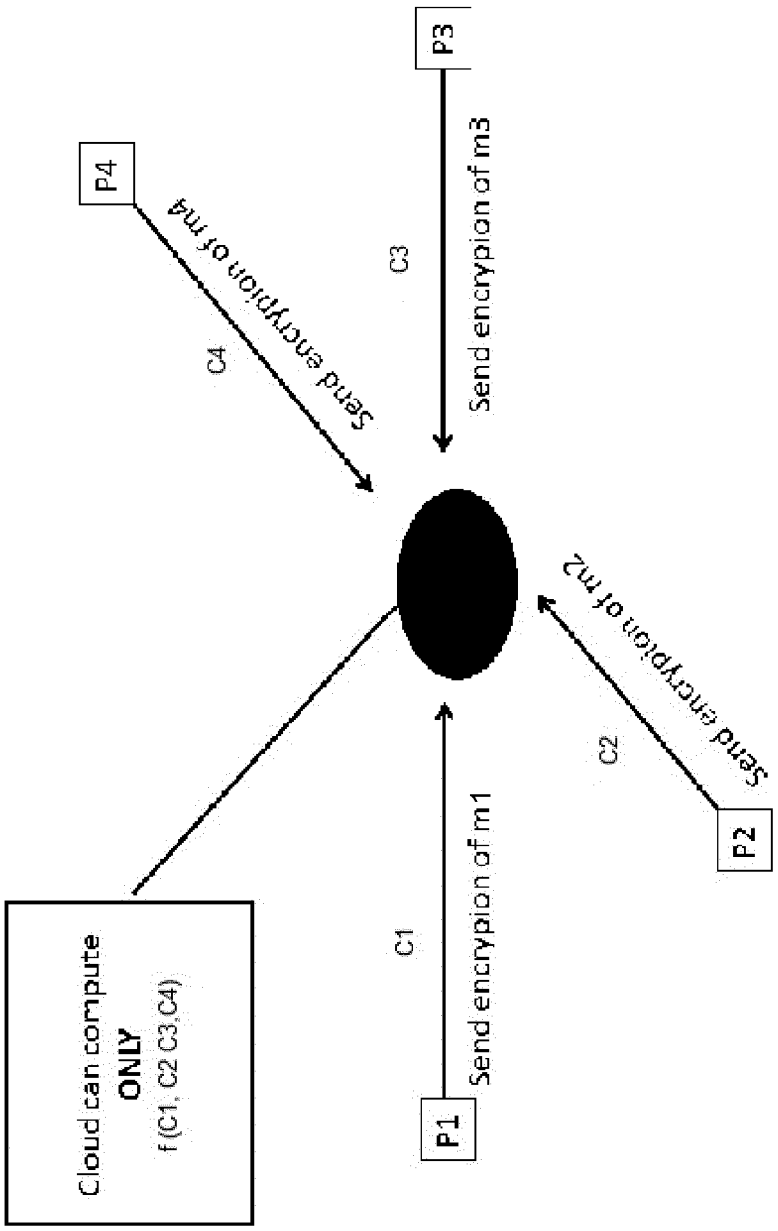
P3
Send encrypion of m3
C3
P4
Send encrypion of m4
C4
Send encrypion of m2
C2
P2
Cloud can compute
ONLY
f (C1, C2 C3,C4)
C1
Send encrypion of m1
P1

Fig. 1

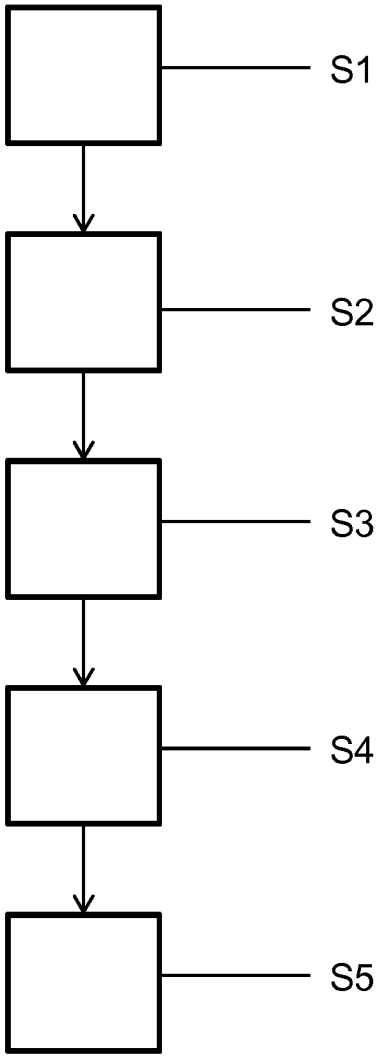

Fig. 2

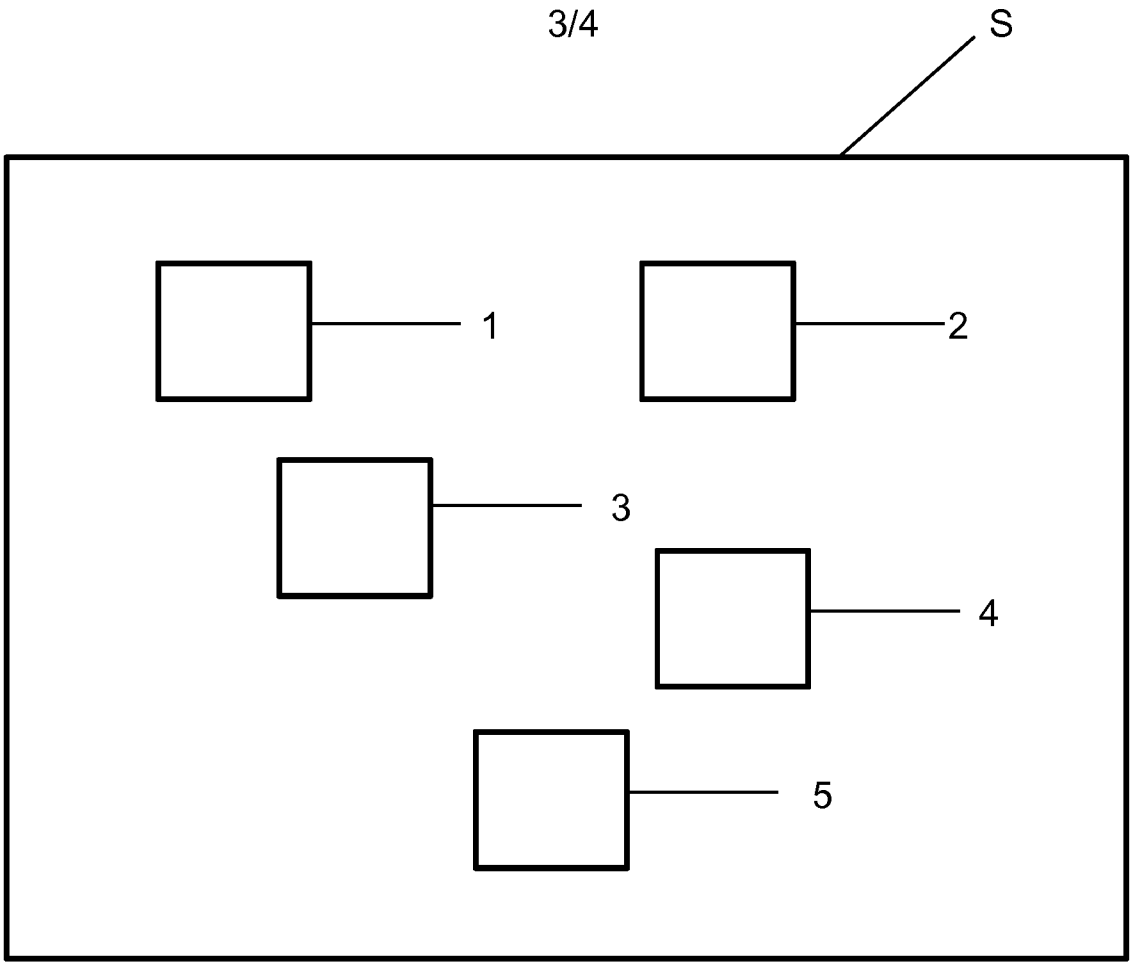

Fig. 3

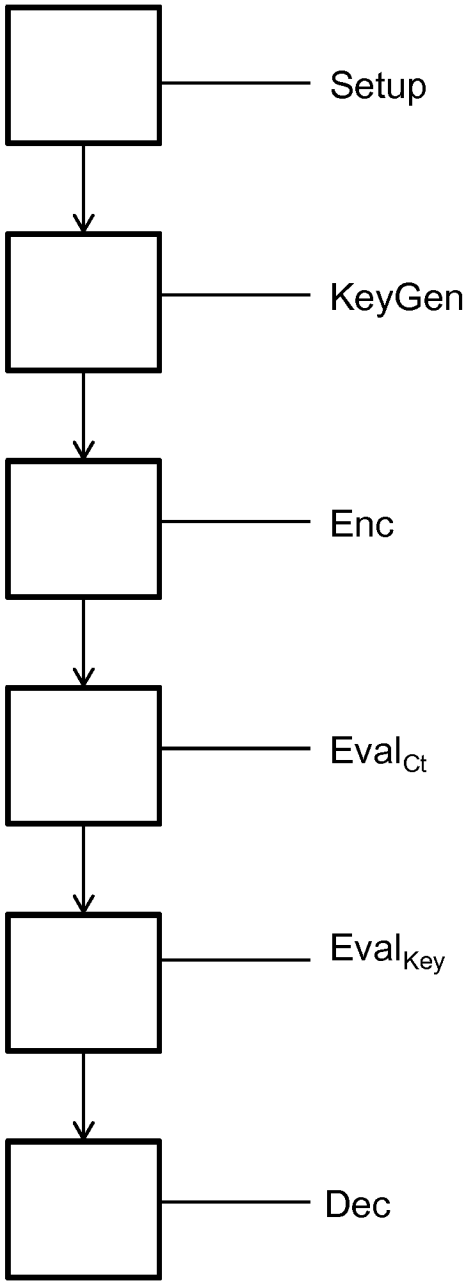

Fig. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/059376

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/30
ADD.

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

B. FIELDS SEARCHED

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

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WANG BOYANG ET AL: "A tale of two clouds: Computing on data encrypted under multiple keys", 2014 IEEE CONFERENCE ON COMMUNICATIONS AND NETWORK SECURITY, IEEE, 29 October 2014 (2014-10-29), pages 337-345, XP032714698, DOI: 10.1109/CNS.2014.6997502 [retrieved on 2014-12-23] section III.A; figure 3 ----- -/--	1-15

[X] 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
13 January 2016	20/01/2016

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Billet, Olivier

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DAN BONEH ET AL: "Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE, and Compact Garbled Circuits", INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH,, vol. 20140603:185537, 20 May 2014 (2014-05-20), pages 1-41, XP061016133, [retrieved on 2014-05-20] the whole document ----	1-15
A	ADRIANA LOPEZ-ALT ET AL: "On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption", INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH,, vol. 20130221:153834, 21 February 2013 (2013-02-21), pages 1-44, XP061007327, [retrieved on 2013-02-21] the whole document ----	1-15

1