Earthline Journal of Mathematical Sciences
http://www.earthlinepublishers.com
Volume 1, Number 1, 2019, Pages 97-103
https://doi.org/10.34198/ejms.1119.97103

Fully Homomorphic Cipher Based on Finite Algebraic Structures

Y. Ts. Alaverdyan[1] and E. G. Satimova[2]

[1] National Polytechnic University of Armenia, Yerevan, Armenia
e-mail: ealaverdjan@gmail.com

[2] Almaty University of Power Engineering and Telecommunications, Almaty, Kazakhstan
e-mail: lenasat12@gmail.com

Abstract

A way to increase the robustness of a cryptographic algorithm toward unauthorized inversion can be obtained through application of non-commutative or non-associative algebraic structures. In this regard, data security became a great issue in adaptation of cloud computing over Internet. While in the traditional encryption methods, security to data in storage state and transmission state is provided, in cloud data processing state, decryption of data is assumed, data being available to cloud provider. In this paper, we propose a special homomorphism between self-distributed and non-associative algebraic structures, which can stand as a premise to construct a homomorphic encryption algorithm aimed at the cloud data security in processing state. Homomorphic encryption so developed will allow users to operate encrypted data directly bypassing the decryption.

1. Introduction

Cryptography is a reliable means to guarantee the confidentiality of information. However, with the growth and development of information transmission media, telecommunications and computer networks, and especially the Internet, cryptography faces new challenges. Particularly, a need for provision of computations on encrypted data became actual.

Received: December 30, 2018; Accepted: January 15, 2019

2010 Mathematics Subject Classification: 94A60.

Keywords and phrases: cryptology, non-associative algebra, self-distributed algebra, homomorphic mapping.

There are several specific requirements for a secure cloud service. Firstly, the client data should be stored in such a way that, when reading them, it would be impossible to understand what the data is. That is, the data should be sent to the server already encrypted. This means that encryption should be done on the client side. Secondly, there must be the ability to process this data without deciphering it. Otherwise, the cloud server becomes just a secure repository. And for each operation on the data there will be a need to send them to the customer side.

Client data encrypted with a standard cipher and placed on a remote server may be processed several ways. In case the client trusts the server with its secret key, the server can decrypt the data and make the necessary changes, and then encrypt it again. However, in this case the server will be able to read all the client data, also, it may be possible to intercept a key when it is transmitted to the server through a channel. As a result of this, client data may become available to some third party.

Another way to process the encrypted data on the server is to download the data from the cloud to the end computer, decrypt, perform the necessary calculation on them and, if necessary, encrypt this result and upload it to the cloud. However, this will require a lot of time and computational resources that the client perhaps does not possess, since the client resorted to using a cloud service.

To perform secure computing on client data stored on a remote untrusted cloud server, homomorphic encryption is applied in order to perform calculations on the encrypted data without decrypting it. Firstly, the homomorphic encryption was put forth by the inventors of the RSA cryptosystem [1]. The RSA cryptosystem itself provided a multiplicative homomorphism, i.e., allowed to carry out the multiplication of ciphertexts, and after decrypting, extract the product of the plaintexts without compromising the data privacy.

The equation $(E(m1) \cdot E(m2)) = m1 \cdot m2$ stands for an example of a multiplicative homomorphic encryption, where $D(\cdot)$ stands for the decryption function; $E(\cdot)$ stands for encryption, $E(m1)$ and $E(m2)$ being ciphers of $m1$ and $m2$, respectively.

A particular interest is prescribed to the possibility of constructing fully homomorphic encryption, i.e., encryption, allowing to carry out any necessary calculations over ciphertexts. For example, such a cryptosystem could be obtained if it were homomorphic both for the operation of addition and for the operation of multiplication.

However, homomorphic cryptosystems have a fundamental disadvantage in that they introduce redundancy in data which can lead to the instability of cryptographic algorithms. Especially, for public-key cryptosystems, the desire to improve cryptographic strength leads to a decrease in efficiency.

In contrast, for symmetric ciphers with no increase in ciphertexts in the process of homomorphic computations, crypto-resistance to specific attacks is known to be quite sufficient for most applications.

However, development of newly homomorphic encryption algorithms capable to answer both symmetric and public key encryption requirements without the redundancy, is an actual problem.

The paper suggests one such solution to protect cloud data storage and processing through involvement of modern discrete mathematical structures.

2. Mathematical Preliminaries

An algebraic structure, written $\langle X, Y, ..., \circ, *, ..., R_1, R_2, ..., x, y\rangle$, is an n-tuple, where: elements $x \in X$, $y \in Y$ are distinct; domains and ranges of functions, also n-ary operations $\circ$ and $*$ are cartesian products of the sets; binary relations R_1 and R_2 are defined on the sets.

Non-associative algebra assumes a vector space over a field, which defines the operation of multiplication interacting with the addition operation by the ordinary distribution law. The operation of multiplication, meanwhile, is not necessarily commutative or associative. For example, a quasigroup, unlike finite groups, does not possess associativity, neither has an identity element. Obviously, handling such structures without possessing knowledge on their construction will require an exponential number of looking ups in order to identify underlying components. In the case of non-associative algebraic models, the number of parentheses can be huge with each placement of parentheses dictating a unique type of computing [2].

An algebraic structure, denoted by $(Q, *)$ is a quasigroup, if for any two its elements $a, b \in Q$, each of the following equations, $a * x = b$ and $y * a = b$, has a unique solution belonging to Q. In other words, a quasigroup is a set Q with a binary operation $*$, such that $Q \to Q$ satisfying to the following rule:

$$(\forall a, b \in Q)(\exists!\, x, y)\, a * x = b,\ y * a = b.$$

It is clear from the definition that if a quasigroup is a finite non-empty set, then each row and each column of its table provides a permutation of Q. The paper [3] presents a method for constructing quasigroup-based stream ciphers which can be applied for developing both symmetric and public key cryptographic algorithms. A recent paper [4] introduces a taxonomy for cryptographic schemes based on the problem of multivariate quadratic equations over quasigroups.

The concept of a quasigroup is equivalent to the definition of an algebra of three operations that satisfies the following four identities:

$$y = x * (x \setminus y),$$

$$y = x \setminus (x * y),$$

$$y = (y / x) * x,$$

$$y = (y * x) / x.$$

A quasigroup $(Q, *)$ and its parastrophs $(Q, /)$ and $(Q, \setminus)$ present a mutual inverse permutation, satisfying to the following conditions:

$$x \setminus (x * y) = y,$$

$$x * (x \setminus y) = y.$$

If a quasigroup is endowed with an identity element, then each element of the quasigroup has own inverse. Otherwise, a lookup table for inverting the quasigroup elements should be constructed.

Note also, that the multiplication table of any quasigroup is a Latin square with a balanced number of elements. Balanced cryptographic structures are known to be highly resistible against differential cryptanalysis.

By definition, a quasigroup operation is commutative, but not associative, and regrouping quasigroup operations results in distinct computation. This very circumstance and balanced structure makes it attractable to construct modern ciphers over quasigroups. Moreover, any quasigroup can be easily converted into a collection of balanced Boolean functions, also can be used for generating a required number of Boolean functions of a predefined order of non-linearity.

With this regard, a left self-distributive algebra $(Q, *)$ that satisfies the identity

$x * (y * z) = (x * y) * (x * z)$ for every $x, y, z \in Q$, and the operation $*$ distributes over itself, can stand for a non-associative version of the more well-known non-abelian finite groups [5]. The construction was defined by considering the conjugator searching problems (CSP) in noncommutative groups.

Suppose that Q is a non-empty set, and $F : Q \times Q \to Q$ is a well-defined function. If the following formula holds,

$$F_r(F_s(p)) = F_{F_r(s)}(F_r(p)), \quad (\forall p, r, s \in Q), \tag{1}$$

then the function $F_*(*)$ is called a *left self-distributive system*.

If $F_r(Q)$ is considered to be a binary operation $r * s$, then the formula (1) becomes

$$r * (s * p) = (r * s) * (r * p), \tag{2}$$

i.e., the operation $*$ is *left self-distributive* with respect to itself [5].

The hardness of a left self-distributed system implies the one-wayness or pre-image resistance of the same left self-distributed system, i.e., intractability of retrieving s from the pair $(p; F_s(p))$.

To summarize with mathematical preliminaries of the proposed homomorphic mapping, characteristics shown above point to the possibility to apply left self-distributed non-abelian systems (along with quasigroups) in the construction of both symmetric and public key cryptographic algorithms.

Given the fact that two different algebraic structures share similar characteristics, where one structure is defined on a set Q and a similar structure is defined on a set Q', one can establish a homomorphic mapping from Q into Q' that preserves some characteristics of the underlying domain structures.

3. Construction of the Homomorphic Cipher

Given $\langle Q, \circ \rangle$ and $\langle Q', * \rangle$ finite non-empty quasigroups with operations $\circ$ and $*$, respectively, the function $f : Q \to Q'$ introduces a homomorphism from $\langle Q, \circ \rangle$ to $\langle Q', * \rangle$, if for every $x, y \in Q$,

$$f(x \circ y) = f(x) * f(y).$$

A homomorphism between groups is called group homomorphism, and the product of two (and therefore finite number of) group homomorphisms is again a group homomorphism, if it exists.

Indeed, if $f : Q \to Q'$ and $f' : Q' \to Q''$ both are group homomorphisms, then the product $f^{\circ} f' : Q \to Q''$ is solved as is shown below:

$$(f^{\circ} f')(x^{\circ} y) = f'(f(x^{\circ} y)) = f'(fx^{\circ} fy) = f'(fx) * f'(fy) = (f^{\circ} f')x * (f^{\circ} f')y,$$

for every $x,\ y \in Q$.

As for any function, the group homomorphism can also be defined to be injective, surjective and bijective.

The group homomorphic function $f : Q \to Q'$ is injective, or monomorphic, if

$$f(x) = f(y) \to x = y, \quad x,\ y \in Q.$$

If it exists, the product of two (and therefore finite number of) group monomorphisms is again a group monomorphism.

The group homomorphic function $f : Q \to Q'$ is surjective, or epimorphic, if for every $y \in Q'$ there is an $x \in Q$ such that $f(x) = y$.

If it exists, the product of two (and therefore finite number of) group epimorphisms is again a group epimorphism.

The group homomorphic function $f : Q \to Q'$ is isomorphic, if it is monomorphic and epimorphic at the same time. If it exists, the product of two (and therefore finite number of) group isomorphisms is a bijective mapping.

Obviously, for data secure processing, application of monomorphic functions should be restricted, as the same cipher symbol will reveal the same plaintext symbol in every of its occurrence.

4. Conclusion

Non-trivial combinations of non-commutative and non-associative algebraic structures can serve as a premise to construct secure cloud ciphers and process the data at the server site without decryption. The proposed solution does not introduce redundancy

in data, and therefore, preserves a predefined level of security. Non-linearity of the algebraic groups is dictated by non-linearity of homomorphic mapping under consideration. Particularly, an epimorphic mapping between left self-distributed and non-associative quasigroups has been recommended.

References

[1] R. L. Rivest, L. Adleman and M. L. Dertouzos, On data banks and privacy homomorphisms, *Foundations of Secure Computation* 32(4) (1978), 169-178.

[2] Yu. Movsisyan, Superidentities in algebras and varieties, *Russian Math. Surveys* 53 (1998), 57-108.

[3] C. Koscielny, A method of constructing quasigroup-based stream ciphers, *Appl. Math. Comput. Sci.* 6 (1996), 109-121.

[4] C. Wolf and B. Preneel, Taxonomy of public key schemes based on the problem of multivariate quadratic equations, *Cryptology ePrint Archive*, *Report* 2005/077, 2005.

[5] Licheng Wang, Lihua Wang, Zhenfu Cao, Eiji Okamoto and Jun Shao, New cryptosystems from CSP-based self-distributive systems, *IACR Cryptology ePrint Archive*: *Report* 2009: 566, 2009.