# 全同态加密：云加密数据搜索

姓名：侯添久

学号：1120161912

班级：08111602

# 

目录

[全同态加密：云加密数据搜索 1](#_Toc1992798967)

[1 摘要 3](#_Toc817279757)

[2 介绍 3](#_Toc715469687)

[3符号标记 5](#_Toc1138089856)

[4 同态加密 5](#_Toc239365963)

[5 测试两个加密消息是否相等 6](#_Toc786869310)

[6 加密数据搜索 7](#_Toc708194944)

[7 PIR 8](#_Toc1278052134)

[8总结 9](#_Toc1090778844)

[参考文献 10](#_Toc1799620316)

# **1 摘要**

加密数据的搜索是一个非常重要的特性，它可以提供一些加密的解决方案。 同时，它也意味着可以在对加密数据进行查询时不用解密相应的数据，这可以在很大程度上保护敏感数据的机密性。个人信息检索是一个必不可少的约定，当被选中的信息从远程数据库中获取时，希望处理是安全的，全同态加密是密码学的革命性领域，它允许操作加密的数据而不需要对数据进行解密，因此，对于加密的结果适合用于纯文本中。在这篇文章中，我们将使用重要的可以提供强有力的完全同态加密的特性来展示如何实现一个PIR协议以及如何在云环境中搜索加密的数据。

# **2 介绍**

在云计算环境和外包数据到云存储的中，许多客户使用云服务来减少运行时数据备份和基础设施维护的花费，一个客户端将敏感的信息以加密的方式存储在公开的云环境中来避免那些不信任的服务和第三方使得他自己的敏感数据泄漏。

法律有时会要求强制性加密某些特殊的数据，比如：许多国家围绕法律这个词强调EHR必须被加密，正常使用加密后的数据变得极具挑战。加密数据的搜索是一种重要的特性，他可以提供一些加密的解决方案。这种也是具有强烈需求的，当用户想查询他加密的数据在云服务器上。

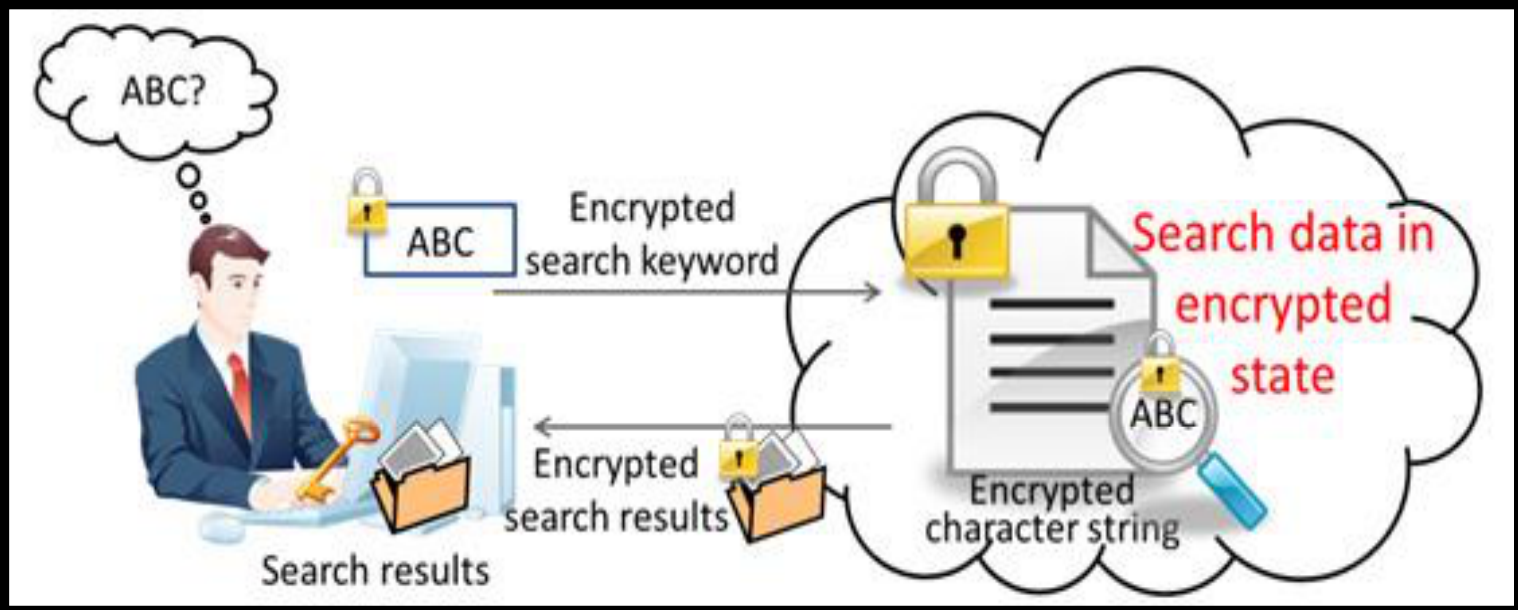
搜索加密数据首次是由Song，Wagner和Perrig提出的，方案中的是可搜索对称加密（ SSE ），这是以对称密钥密码学为基础的。他可以允许单个客户端读写数据，并只有密钥持有者可以创建可搜索的密文。第二类可搜索的加密解决方案在Boneh等人的开创性工作之后被提出，他发明了一种可基于关键词搜索（ PEKS ）的加密方案，这是基于公钥的加密方案，所有消息在使用公共的公钥加密，只有私钥可以对其进行解密，这种方案允许多个客户端可以进行数据的写（加密），但是只会有一个客户端进行数据的读取（解密）。

私人信息检索允许用户获得数据库的某个属性，该属性通过数据库的隐藏属性获取。虽然这种解决方案存在一些小问题，客户端检索整个数据库并进行完美的隐私查询，但是这种方案不适用于大型数据库，因为其会增加通信的复杂。

将加密数据搜索与私有数据检索相结合能够提升云安全性，并且可以为敏感数据提供更多保护。 这个解决方案为客户提供更大的灵活性来应对他们使用云来存储和管理数据，因为它允许云计算在加密的数据上执行更多操作。

全同态加密是一种强有力的工具，它可以允许搜索加密的数据并实现PIR协议，它使云服务器能够盲目搜索客户的加密的数据而不需要解密它，或者获得任何有关明文数据或搜索查询。同态加密第一次的想法是在1978年，由Rivest，Adleman和 Dertozous 以隐私同态被提出的，但是当时隐私同态仍然只是一个猜想而没有一个实际有效的解决方案知道2009年，在Craig Gentry的文章中，他提出了第一个语义安全的完全同态下隐私同态加密的技术方案，这种方案通过使用bootstrapping theorem来减小在对加密数据处理后产生的噪音，Gentry的明文空间是用一种二进制的字段表示，在这种同态加密的方案下，就类似于在一个空间中，它允许我们在加密的数据上评估任何的环。一些紧随着Gentry之后的研究取得了突破并且提出了新的完全同态加密的解决方案，这些新的方案尽可能的使用更大的明文空间来减小加密时的消耗。

下图展示了一个基于关键字的加密数据搜索流程，首先用户输入自己想加密的关键字，然后对其进行加密操作，将密文发送至云服务器上，云服务器接收到密文后，按照密文进行搜索，得到搜索的结果，最后将搜索的结果返回给用户，用户解密拿到结果。



在本文中，我们提出了一种有效的搜索解决方案和基于全同态加密的PIR协议。因次，我们主要将关注于基于全同态加密中的二进制环，即密码系统使用异或和与操作使得明文空间处于{0，1}。

# **3符号标记**

在下文中，符号ℰ（m）表示加密后的消息m，使用符号D（c）表示解密后的明文c。简单来说，如果我们有c = ℰ（m），然后m = D（c）。

# **4 同态加密**

从广义上将，同态加密是一个加密系统，它允许用户除了基本的加密和解密操作之外，还可以在加密的数据上进行计算。如果加密系统允许有限的计算操作，那么它被称为部分同态。例如：对于未进行填充的RSA加密系统，可以认为是可以相乘的同态，即，对于明文m1和明文m2，我们可以得到ℰ（m1）= m1e % N和ℰ（m2）= m2e % N，N是RSA的模数，e是RSA公钥的指数，因此，可以得到ℰ（m1）ℰ（m2）=（ m1e ·m2e ）% N = （m1 ·m2）e % N = ℰ（m1 ·m2）。RSA允许我们只是仅仅对密文进行乘法操作而不能进行加法操作，关于加法的同态加密方案是Paillier的加密系统，如果公钥是模N并且是基于g的，那么加密的信息m则是ℰ（m）= gmrN % N2，对于r是区间[1，N-1]之间任意的数，Paillier加密算法的同态特性则是ℰ（m1）ℰ（m2）=（gm1 r1N）·（gm2 r2N）% N2

= gm1+m2 （r1 r2）N % N2 = ℰ（m1 + m2）。

从另一个方面来讲，完全同态加密系统是一种允许客户端除了进行基本的加密与解密操作之外，还可以在加密数据上进行任何的计算操作而不需要解密数据。假设，我们有两个明文m1和m2，c1 = ℰ（m1）和c2 = ℰ（m2）分别是其加密后的密文，完全同态加密可以让我们得到c1 + c2 = ℰ（m1 + m2）和c1·c2 = ℰ（m1 ·m2），因此，用户可以将自己复杂的计算交给远程的云服务器去完成，云服务在它自己的运行环境中可以提供无限制的计算。

从数学上将，一个完全同态加密算法是一个四元组的多项式( Gen , Enc , Dec , Eval )，验证：

Gen(ƛ)：是一种密钥生成算法，输入一个安全的参数ƛ，输出公共的和私密饿的密钥（pk，sk）。

Enc(m，pk)：是一种加密算法，输入明文m和公钥pk，输出加密后的密文c。

Dec(c，sk)：输入密文c和解密密钥sk，输出明文m。

Eval(C，c1，c2，...，cn)：是一种评估算法，输入环C和密文c1，c2，...，cn，并验证Dec(Eval(C，c1，c2，...，cn)，sk）= C(m1，m2，...，mn)，任何人都可以评估Eval，因为它不需要密钥sk。

接下来，我们的方案将基于环的加密系统，它允许客户端逐位的加密明文。

假设我们有一个数据库DB，它里面有许多机密的数据c1，c2，...，cn，数据库是放在远程云服务器上的。

# **5 测试两个加密消息是否相等**

因为是完全同态加密方案，一个人可以在不解密消息的条件下测试两个加密的信息相等或者不相等。在本节中，我们将致力于equal(c1,c2)，它接收输入的两个密文c1= ℰ（m1）和c2 = ℰ（m2）并且可以验证m1和m2是否相等而不需要去解密所给定的密文。为了实现这个操作，我们需要在密文上做以下两步操作。

5.1反转

加密位的反转是一种可以允许我们改变一个位仅仅是基于密文的反转，这种完全是可以通过完全同态加密算法实现的，例如：

假设，θ ϵ {0,1}，并且c = ℰ（θ），我们可以得到c = ℰ（），使得 = θ ⨁ 1是θ的反转。

c可以通过c的计算c = c + ℰ（1）= ℰ（θ）+ ℰ（1）= ℰ（ θ + 1 ）= ℰ（ ）。

5.2 补码

二进制的补码被定义为通过反转二进制中所有的位获得的值的数字表示形式，这个是一种同时反转多个为得到的。为了计算出逐位输入加密的补码，我们需要对每个加密位进行反转。

令m位二进制的明文（并不需要是一位）加密成c =ℰ（ m ）并且c = ℰ（） ，使得是m的补码，我们可以得到c = ℰ（ m ）+ℰ（1 1 1 ... 1）= ℰ（m ⨁ 1 1 1 ... 1）=ℰ（ ） 。

5.3相等

因为它优于equal(c1, c2)以两个密文作为输入，equal将会返回加密的结果，他之前通过单个位被解密。如果位的值是1，那么证明m1和m2是相等的，否则说明m1和m2是不相等的。

为了在只使用c1和c2的条件下测试m1和m2是否相等，我们执行一下操作：

1.按位进行加法运算，c1 ⨁c2 = ℰ（ m1 ⨁ m2 ）。

如果m1 = m2，我们可以得到C = c1 ⨁c2 = ℰ（0 0 0 ... 0），否则，我们将在c1⨁c2运算的加密位中得到只有有一个等于1.

2.计算 c1⨁c2 的补码 c1⨁c2。

3.将所得到的结果的所有加密位相乘，我们将会得到一个被加密的位（0或1）。

4.如果获得的结果equal(c1 ,c2)=ℰ（1），之后我们会有m1 = m2，否则，equal(c1,c2)= ℰ（0），那么m1 ≠ m2。

# **6 加密数据搜索**

当企业正在测试用于存储的云应用程序和数据库时，安全仍然是云计算中最主要的难题。如今，将数据以一种加密的形式存储于云数据库中是非常必须的，加密可以使我们保护数据中敏感的内容，但是这通常意味着为了安全必须牺牲一定的功能性。搜索加密数据是一种很意义的解决方案，它允许云服务器可以随意的搜索用户的密文，基于一个trapdoor令牌，它包含了客户端上传的需要搜索的关键字。当然，客户端必须在上传的时候加密那些需要搜索的关键字，这样也会增加一定的消耗。因为是任意的，所以它意味着云服务器不需要获取任何不必要的只是关于搜索的关键字和加密的数据在整个查询的过程中。

完全同态加密允许我们查询加密的数据库而不需要解密的密钥，除了对密文的计算之外，还可以在密文上进行任意的搜索。

假设，我们有一个数据库，它里面有数据c1，c2，...，cn，为了判断里面是否有加密的密文信息c = ℰ（m），我们可以进行如下的操作，在将查询请求发送到服务器之后，过程如下：

从查询中获取加密消息c并且测试它是否于我们加密数据库中每一行记录相等。

因为这个阶段，我们可以获取到n个加密位，b1 = equal ( C, c1 )，b2 = equal ( C , c2 )，... ，bn = equal ( C , cn ）。

如果数据库中不存在c，那么n个加密位都会等于被加密的0，即：b1 b2 b3 ... bn = ℰ（0 0 0 ... 0），否则，我们将获得至少一个bi  = ℰ（1），对于给定的i ϵ[1,n]，即：b1 b2 b3 ... bn = ℰ（0 0 0 ... 1 ... 0）。

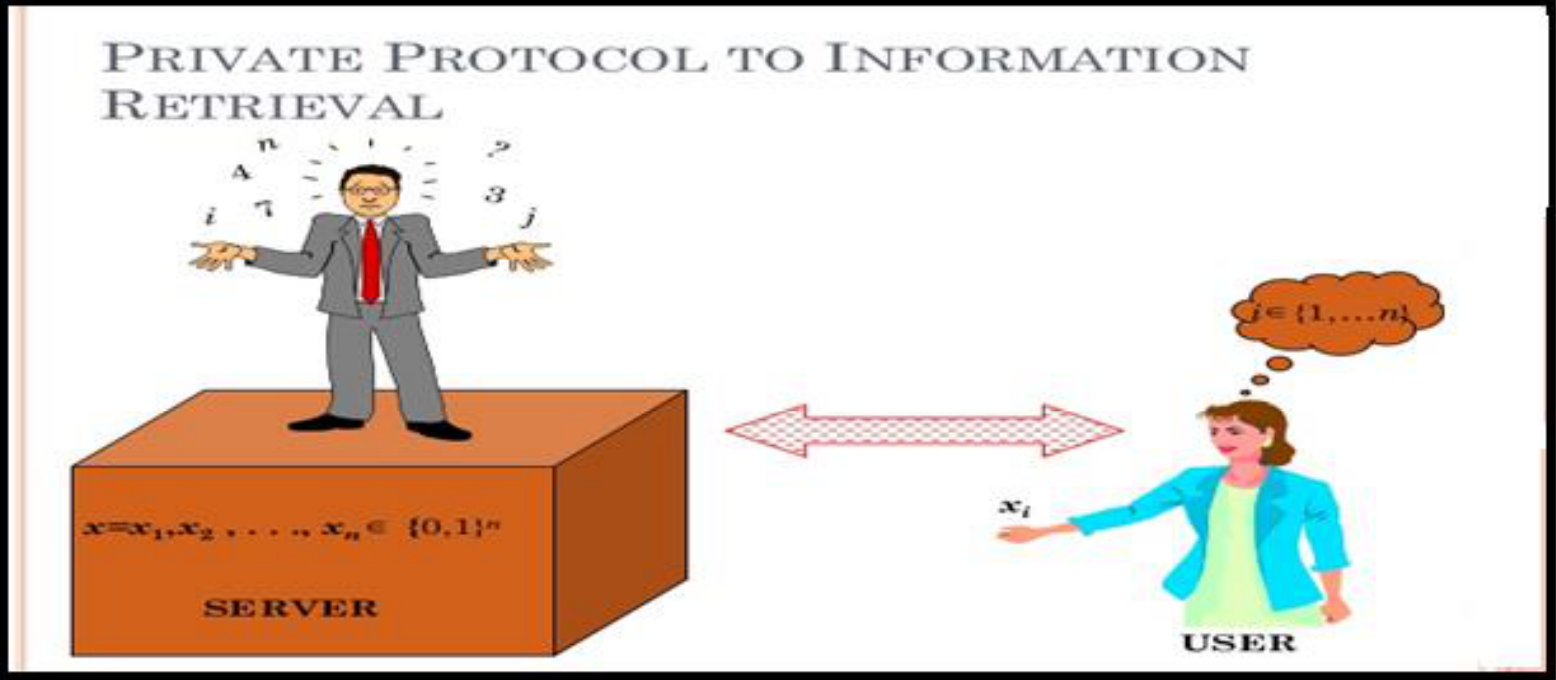
因此，要验证数据库中是否存在c，服务器应该测试（b1 b2 b3 ... bn）的结果与ℰ（0 0 0 ... 0）是否相等。即：它计算SEARCH = equal ( b1 b2 b3 ... bn，ℰ（0 0 0 ... 0））。

得到的结果将会被反转并且发送给请求方。即：SEARCH。

在收到加密后的SEARCH之后，请求方会将其解密。将会展现出两种结果：如果他发现是0，那么查询的数据不存在，否则，他将会发现1，也就意味这查询的结果是存在的。

# **7 PIR**

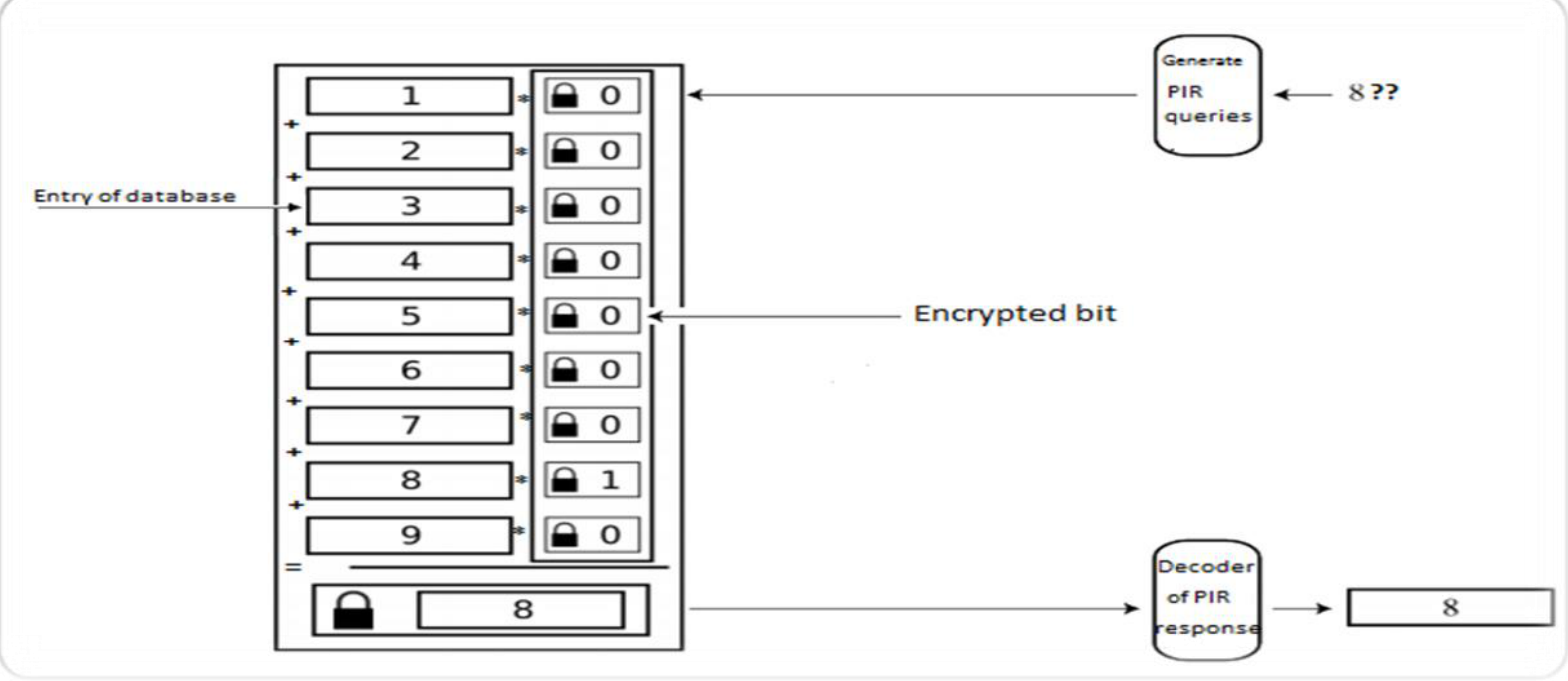
假设某个客户端已经查询了数据库并且得到肯定的结果搜索的数据是存在于数据库中的。PIR协议允许他获取他想要的信息以一种更加安全的方式并且不需要云服务器去决定哪些部分是被选中的。PIR协议可以与可被搜索的方案进行组合来得到一个更安全的协议，它可以允许客户端搜索加密在云上的数据并且进行私下的检索。当放在云上存储当数据是很敏感时，这种方案是非常好的。完全同态加密能够允许两者：搜索加密数据和私人检索所需的数据库的数据。



7.1完全同态加密中的PIR

PIR协议是可以通过完全同态加密实现的。假设，数据库中存在数据e1，e2，...en，他们是通过一个独一无二的整数进行组织的。这些数据库中的数据是不应该被加密的，他们只是简单的整数。我们想获取第k条记录而不需要知道数据库的所有者，假设我们有一个完全同态的加密方案。

客户端首先通过计算加密密文ci ，其中ci 是除i = k之外的0加密，在这种情况下ck是1的加密。因为我们的密码系统是一种概率的解决方案，0可以以不同的方式被加密，因此，所有的ci 都是不相同的。客户端将查询（c1 , c2 , c3 ... , cn ）发送到云服务器上，数据库的管理员计算c1e1 + c2e2 + c3e3 + ... cn en，并将结果返回给客户端。然后，他计算D（c1e1 + c2e2 + c3e3 + ... cn en）并且检索ek。实际上，我们有c1e1 + c2e2 + c3e3 + ... cn en = ℰ( 0 ) e1 + ℰ( 0 ) e2 + . . . + ℰ( 1 ) ek + . . . + ℰ( 0 ) en = ℰ (0 \* e1 + 0 \* e2 + . . . + 1 \* ek + . . . + 0 \* en ) = ℰ( ek ) 。



# **8总结**

在本文中，我们解决了搜索加密数据和私人信息检索（PIR）协议等问题，所有的操作都在存储数据的远程服务器当中，在数据库加密的情况下，对于在云环境中，将搜索加密数据与私人信息检索放在一个协议里是一种非常好的解决方案，我们协作这两种技术都是基于完全同态加密的。所有的操作都简单的执行在一个加密系统中，它允许我们在加密数据上进行搜索而不用事先对其进行解密。

# **参考文献**

[1] D. Song , D. Wagner et A. Perrig, «Practical Techniques for Searches on Encrypted Data,» IEEE Symposium on Security and Privacy, pp. 44-55, 2000.

[2] D. Boneh, G. Crescenzo, R. Ostrovsky et G. Persiano, «Public key encryption with keyword search,» chez EUROCRYPT (LNCS), 2004.

[3] B. Chor, O. Goldreich, . E. Kushilevitz et M. Suda, «Private information retrieval,» In Proc. of the 36th Annu. IEEE Symp. on Foundations of Computer Science Pages 41–51, 1995. Journal version: J. of the ACM, 45:965–981, 1998.

[4] R. Rivest, L. Adleman et M. Dertouzos, ««On Data Banks and Privacy Homorphisms,»,» In Foundataions of Secure Computataion, Academic Press, pp. 169-179, 1978.

[5] C.Gentry, «A fully homomorphic encryption scheme,

»https://crypto.stanford.edu/craig/craigthesis.pdf, September 2009.

[6] M. van Dijk, C. Gentry, S. Halevi et V. Vaikuntanan, «« Fully homomorphic encryption over the integers,»,» Cryptology ePrint Archive, Report 2009/616, 2009. http://eprint.iacr.org/..

[7] Z. Brakerski, C. Gentry et V. Vaikantanathan, «« Fully Homomorphique Encryption without Bootstrapping,»,» Available at <http://eprint.iacr.org/2011/277..>

[8] J. Fan et F. Vercauteren, ««Somewhat Practical Fully Homomorphic Encryption,»,» Available at http://eprint.iacr.org/2012/144. .

[9] J. Kim, M. Sung Lee, A. Yun et J. Hee Cheon, «“ CRT-based Fully Homomorphic Encryption over the Integers”,» Available at <http://eprint.iacr.org/2013/057.>

[10] A. Kipnis and E. Hibshoosh, « « Efficient Methods for Practical Fully Homomorphic Symmetric-key Encrypton, Randomization and Verification »,,» Cryptology ePrint Archive, Report 2012/637..

[11] R. Rivest, A. Shamir et L. Adleman, «A Method for Obtaning Digital Signatures and Public Key Cryptosystems,» Commununications of ACM, Vol. 21, pp. 120-126, April 1978..

[12] P. Paillier, «Public-key cryptosystems based on composite degree residuosity classes.,» chez 18th Annual Eurocrypt Conference (EUROCRYPT'99), Prague, Czech Republic, 1999.

[13] D. Boneh, E. Goh et K. Nissim, «Evaluating 2-DNF formulas on ciphertexts. In Kilian, J., ed.: Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings. Volume 3378 of Lecture,» chez Theory of Cryptography, Second Theory of Cryptography Conference, Combridge, February 10-12, 2005.