·安全技术· 文章编号：1000-3428(2019)09-0143-04 文献标志码：A 中图分类号：TP391

基于 cuFHE 的同态比较运算器

刘文超[a]，潘 峰[a,b]，杨晓元[a,b]，周潭平[a,b]，涂广升[a]

（武警工程大学 a. 密码工程学院；b. 网络和信息安全武警部队重点实验室，西安 710086）

摘 要： 为在密态计算中实现高效的比较操作，设计一种支持并行加速的多比特同态比较运算器。基于 cuFHE 软件库构造单比特同态数值比较器，在并行运算模式下调用该同态数值比较器，通过 GPU 硬件实现可比较任意比特明文的多比特同态比较运算器。利用 cuFHE 同态算法库编写同态比较运算函数并进行测试，结果表明，该比较运算器效率较高，对100 bit的明文进行一次比较运算仅需 0.91 s。

关键词： 密态计算；全同态加密；并行加速；cuFHE 软件库；同态比较运算器

开放科学（资源服务）标志码（OSID）：

中文引用格式： 刘文超，潘峰，杨晓元，等. 基于 cuFHE 的同态比较运算器[J]. 计算机工程，2019，45(9)：143-146，152.

英文引用格式： LIU Wenchao，PAN Feng，YANG Xiaoyuan，et al. Homomorphic comparison operator based on cuFHE[J]. Computer Engineering，2019，45(9)：143-146，152.

Homomorphic Comparison Operator Based on cuFHE

LIU Wenchao[a]，PAN Feng[a,b]，YANG Xiaoyuan[a,b]，ZHOU Tanping[a,b]，TU Guangsheng[a]

（a. College of Cryptography Engineering；b. Key Laboratory of Network and Information Security of PAP，Engineering University of PAP，Xi'an 710086，China）

【Abstract】 A multi-bit homomorphic comparison operator supporting parallel acceleration is designed to achieve efficient comparison operation in dense state computing. A single-bit homomorphic digital comparator is constructed based on cuFHE software library，and it is called under parallel computing mode. A multi-bit homomorphic comparison operator that can compare plaintexts of any length is implemented by GPU hardware. The cuFHE homomorphic algorithm library is used to write homomorphic comparison operation function，and the function is tested. Results show that the comparison operator is more efficient，and only takes 0.91 s to perform a comparison operation on an 100-bit plaintext.

【Key words】 dense state computing；Fully Homomorphic Encryption（FHE）；parallel acceleration；cuFHE software library；homomorphic comparison operator

DOI：10.19678/j.issn.1000-3428.0053715

0 概述

全同态加密（Fully Homomorphic Encryption，FHE）能够对密文进行任意运算，且解密结果等同于对明文进行相同运算后的结果。FHE 既具有机密性又能进行密态计算，在云计算环境下可以提升数据的安全性[1]。文献[2]构造一种 CPA 安全的全同态加密方案，其标志着第一代全同态加密算法的出现，引起了密码学界的广泛关注。文献[3]基于环上误差学习问题（Ring Learning With Errors，RLWE）构造一种全同态加密算法 BV11a，其成为第二代全同态加密方案的典型代表。文献[4]提出近似特征向量方法并构造基于 LWE 问题的全同态加密方案 GSW，该方案标志着第三代全同态加密算法的出现。目前，全同态加密大多需要借助自举过程以对同态计算后的密文进行刷新，得到噪声较小的密文后重新启动同态运算过程。由于自举过程的计算复杂度较高，且其使用的计算密钥会占据大量的存储空间，从而降低了全同态加密方案的效率。文献[5]基于 TGSW 方案，构造自举过程小于 0.1 s 的高效双层全

基金项目：国家重点研发计划“新型数据保护密码算法研究”（2017YFB0802000）；国家自然科学基金“面向云计算的同态密码关键技术研究”（61772550）。

作者简介：刘文超（1994—），男，硕士研究生，主研方向为同态加密、并行计算；潘 峰、杨晓元，教授；周潭平，讲师、博士；涂广升，硕士研究生。

收稿日期：2019-01-17 **修回日期：**2019-03-04 **E-mail：**liuwch3@mail2.sysu.edu.cn

同态加密方案 CGGI16。文献[6]对 CGGI16 进行改进,得出一种效率更高的全同态加密方案 CGGI17。目前,全同态加密方案的效率不断提升[7-9],但其距广泛应用仍有一段距离。因此,设计高效的全同态加密算法,并对其编程实例进行优化,是该领域亟需解决的问题。

密态计算的多数场景均需使用比较操作,该操作也是安全多方计算、密文检索等协议的基本组成模块。文献[10]基于 somewhat 同态加密方案构造多比特的密文数值比较器。文献[11]在 HElib 同态运算库的基础上,构建一种并行比较多比特明文的同态运算模型。

相对于 CPU,GPU 硬件具有更强大的浮点计算和并行计算的能力,可以为全同态加密算法的快速实现提供良好的算力支撑[12]。目前,GPU 已在深度学习[13-14]、影像数据处理[15-16]、密码算法加速[17-18]等领域得到了广泛应用。全同态密码算法的自举技术涉及大量的矩阵乘法和 NTT 运算,因此,可以利用 GPU 来加速实现这些计算密集型运算。cuFHE 库是一个开源的 GPU 加速全同态密码库[19],其借鉴了 cuHE[20] 中 GPU 端的快速数论变换,实现一种基于 CUDA 平台的高效 CGGI 方案。然而,cuFHE 仅提供了基本的门电路接口,并未提供如比较运算等的函数接口。因此,利用 cuFHE 提供的接口开发同态运算器,可以使 CGGI 等同态加密方案得到更广泛的应用。

本文在 cuFHE 全同态加密库的基础上,编写单比特同态比较运算函数,利用 GPU 硬件和 CPU 多线程技术,设计并实现一种可比较任意长度明文的多比特同态比较运算器。

1 CGGI 全同态加密方案

cuFHE 软件库的底层方案为 CGGI。

定义 1(TLWE) 令 $n\geqslant 1$ 为整数,N 为 2 的幂,噪音参数 $\alpha\geqslant 0$,随机均匀选择私钥 $s\in\mathbb{B}_N[X]^k$。消息 $\mu\in\mathbb{T}_N[X]$ 被加密为 $\boldsymbol{c}=(\boldsymbol{a},b)\in\mathbb{T}_N[X]^k\times\mathbb{T}_N[X]$,$b\in\mathbb{T}_N[X]$ 服从高斯分布 $D_{\mathbb{T}_N[X],\alpha,s\cdot a+\mu}$。当向量 $\boldsymbol{a}$ 中的元素满足均匀分布时,实例 $\boldsymbol{c}=(\boldsymbol{a},b)$ 随机;当 $\boldsymbol{a}=0$ 时,实例称为平凡实例;当 $\alpha=0$ 时,实例称为无噪音实例;当 $\mu=0$ 时,实例称为齐次的实例。

搜索问题描述为:给定齐次的 TLWE 实例,找到 $s\in\mathbb{B}_N[X]^k$,使噪音分布集中。

判定问题描述为:区分 TLWE 实例和 $\mathbb{T}_N[X]^k\times\mathbb{T}_N[X]$ 中的均匀分布实例。

定义 2(Phase) 对于 $\boldsymbol{c}=(\boldsymbol{a},b)\in\mathbb{T}_N[X]^k\times\mathbb{T}_N[X]$ 和 $s\in\mathbb{B}_N[X]^k$,定义 TLWE 实例的相位 $\varphi_s(\boldsymbol{c})\triangleq b-s\cdot\boldsymbol{a}$。相位 $\varphi_s(\boldsymbol{c})$ 与 $\mathbb{T}_N[X]^{k+1}$ 上的输入 $(\boldsymbol{a},b)$ 具有线性关系,具体而言,对于 l_∞ 范数 φ_s 服从 $(kN+1)-lipschitzian$,即 $\forall x,y\in\mathbb{T}_N[X]^{k+1}$,$\|\varphi_s(x)-\varphi_s(y)\|_\infty\leqslant(kN+1)\|x-y\|_\infty$。

定义 3(TLWE 实例) $\boldsymbol{c}\in\mathbb{T}_N[X]^{k+1}$ 是一个随机变量。如果存在一个密钥 $s\in\mathbb{B}_N[X]^k$,使得相位的 $\varphi_s(\boldsymbol{c})$ 分布集中,则定义 $\boldsymbol{c}$ 是一个合法的 TLWE 实例。如果 $\boldsymbol{c}$ 是平凡实例,则对于所有的密钥 s,$\boldsymbol{c}$ 都是一个合法的 TLWE 实例;如果 $\boldsymbol{c}$ 服从随机均匀分布,则 s 唯一确定。

CGGI 方案中的自举过程[6]描述如下:

算法 1 Bootstrapping 自举过程

输入 $\mu_1\in\mathbb{T}[X]$,$\boldsymbol{c}=(\boldsymbol{a},b)\in\mathbb{T}_N[X]^k\times\mathbb{T}_N[X]$,$x\in\mathbb{B}$,$Bookstrapping\ key=(BK_i)_{i\in[1,n]}$

输出 $\boldsymbol{c}=(\boldsymbol{a},b)\in\mathbb{T}_N[X]^k\times\mathbb{T}_N[X]$

1. $\mu=\frac{1}{2}\mu_1\in\mathbb{T}[X]$

2. $b=[2Nb]$,$a_i=[2Na_i]\in\mathbb{Z}$,$i\in[1,n]$

3. $v=(1+X+\cdots+X^{N-1})\cdot X^{\frac{N}{2}}\cdot\mu\in\mathbb{T}_N[X]$

4. ACC←BlindRotate((0,v),(a_1,a_2,…,a_n,b),(BK_1,BK_2,…,BK_n))

5. Return (0,μ)+SampleExtract(ACC)

CGGI 方案中所提供的同态运算以 TLWE 密文 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$ 为输入,调用自举算法实现同态与非(HomNAND)、同态异或(HomXOR)、同态与(HomAND)、同态或(HomOR)、同态非(HomNOT)门电路,输出门电路操作对应的密文如下:

$$HomNAND(\boldsymbol{c}_1,\boldsymbol{c}_2)=Bootstrapping((\boldsymbol{0},5/8)-\boldsymbol{c}_1-\boldsymbol{c}_2)$$

$$HomXOR(\boldsymbol{c}_1,\boldsymbol{c}_2)=Bootstrapping(2(\boldsymbol{c}_1-\boldsymbol{c}_2))$$

$$HomAND(\boldsymbol{c}_1,\boldsymbol{c}_2)=Bootstrapping((\boldsymbol{0},-1/8)+\boldsymbol{c}_1+\boldsymbol{c}_2)$$

$$HomOR(\boldsymbol{c}_1,\boldsymbol{c}_2)=Bootstrapping((\boldsymbol{0},1/8)+\boldsymbol{c}_1+\boldsymbol{c}_2)$$

$$HomNOT(\boldsymbol{c})=(\boldsymbol{0},1/4)-\boldsymbol{c}$$

其中,$\boldsymbol{0}$ 表示维度为 n 的零向量,$Bootstrapping$ 为自举过程。

2 同态数值比较器

2.1 单比特同态数值比较器

单比特数值比较器通过比较 2 个输入端的单比特二进制数的大小,在输出端输出不同比较结果的逻辑电路,具体如下。

输入:2 个单比特二进制数 A、B。

计算:

1) $out_1=\overline{A}\cdot B$。

2) $out_2 = (\overline{A} \cdot B) \oplus (A \cdot \overline{B})$。

3) $out_3 = A \cdot \overline{B}$。

其中,·表示 *AND* 运算,⊕表示 *XNOR* 运算,¯表示 *NOT* 运算。单比特数值比较器真值表如表 1 所示。

表 1　单比特数值比较器真值表

输入		输出		
A	B	out_1	out_2	out_3
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

逻辑输出为:

1) $out_1 = 1$,表示 A 小于 B。

2) $out_2 = 1$,表示 A 等于 B。

3) $out_3 = 1$,表示 A 大于 B。

在单比特明文数值比较器的设计过程中,可以抽象出数值比较器的基本特征。对于密文状态下的单比特数值,只需通过如下同态操作即可实现比较运算:

明文 m_0、$m_1 \in Z_2$,定义密文上的运算 *AND*、*XNOR*、*NOT* 满足如下同态性质:

1) $dec(AND(enc(m_0), enc(m_1))) = AND(m_0, m_1)$。

2) $dec(XNOR(enc(m_0), enc(m_1))) = XNOR(m_0, m_1)$。

3) $dec(NOT(enc(m))) = NOT(m)$。

结合以上操作,参照已经实现的单比特明文数值比较器,设计单比特同态数值比较器,其运算过程如算法 2 所示。

算法 2　单比特同态比较算法

输入　m_0, m_1, pk, sk

输出　out_1, out_2, out_3

1. $M_0 = enc(m_0, pk)$, $M_1 = enc(m_1, pk)$
2. $out_1 = (NOT(M_0)) AND (M_1)$
3. $out_2 = (NOT(M_0) AND (M_1))$
   $XNOR((M_0) AND NOT(M_1))$
4. $out_3 = (NOT(M_0)) AND (M_1)$
5. return out_1 , out_2 , out_3

单比特同态数值比较器的真值表如表 2 所示,其中,方框表示在密文上进行该运算。

表 2　单比特同态数值比较器真值表

输入		输出		
A	B	out_1	out_2	out_3
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

对输出结果进行解密:

1) $out_1 = 1$,表示 A 小于 B。

2) $out_2 = 1$,表示 A 等于 B。

3) $out_3 = 1$,表示 A 大于 B。

2.2　并行多比特同态数值比较器

通过逐比特地调用单比特比较器并将结果发回客户端,可以实现多比特比较运算,但其涉及过多的解密运算,通信开销较大,且未利用硬件的并行优势,导致复杂度较高,效率较低。本文在单比特同态比较器的基础上,利用 CPU 多线程技术,通过调用基于 GPU 的单比特数值比较器来实现并行多比特数值的比较运算,从而构造一种任意比特长度的多比特数值比较器。因为该过程只需对单比特明文进行加密,所以其明文空间为 Z_2。定义密文上的运算 *XOR* 满足如下同态性质:

$$dec(XOR(enc(m_0), enc(m_1))) = XOR(m_0, m_1)$$

并行多比特数值的比较运算过程如下:

1) 选取 2 个二进制明文 $m = (x_0, x_1, \cdots, x_n)$, $n = (y_0, y_1, \cdots, y_n)$,分别对这 2 个明文逐位使用全同态方案 CGGI 的加密算法进行加密,得到对应密文 $M = enc(m, pk)$, $N = enc(n, pk)$。

2) 对上述 2 个密文使用多线程技术,并行调用单比特同态数值比较器进行同态比较运算,并保存结果。

3) 对保存的结果进行相应的 *AND* 和 *XOR* 门电路操作,得到最终结果。

对 2 个输入长度为 n 比特的二进制字符串,从高位到低位依次调用单比特同态数值比较器进行运算后可得到 *OUT*,将 *OUT* 解密为 *out* 后,程序结束并输出 $m \geqslant n$ 或 $m < n$。并行多比特明文同态比较算法描述如下:

算法 3　多比特同态比较算法

输入　$m = (x_0, x_1, \cdots, x_n), n = (y_0, y_1, \cdots, y_n), pk, sk$

输出　*OUT*

1. parallel for i = 0 to n
2. $M_i = enc(x_i, pk)$, $N_i = enc(y_i, pk)$
3. $FLevel_{i,1} = (NOT(M_i)) AND (N_i)$
4. $FLevel_{i,2} = (NOT(M_i) AND (N_i))$
   $XNOR((M_i) AND NOT(N_i))$
5. $SLevel_0 = FLevel_{1,2}$, $OUT = FLevel_{1,1}$
6. for i = 1 to n
7. $SLevel_i = (SLevel_{i-1}) AND (FLevel_{i,1})$
8. $OUT = (OUT) XOR (SLevel_i)$
9. return OUT

并行多比特同态比较器的真值表如表 3 所示。

表 3　多比特同态比较器真值表

输入 m、n	输出 *out*
$m \geqslant n$	0
$m < n$	1

3 实验结果与分析

本文在 cuFHE 全同态加密库的基础上,编写单比特同态比较函数,利用 CPU 多线程技术和 GPU 硬件,设计一种并行多比特同态比较器。本次实验分别对比较器的加解密时间、自举过程时间、单比特比较器运行时间和多组多比特比较运算时间进行测试。

实验环境:硬件采用 Intel(R) Core(TM) i7-8700 3.20 GHz CPU,8 GB 内存,NVIDIA GTX1080 8 GB 显卡,软件系统运行 Ubuntu 16.04 LTS 64 位操作系统(Linux kernel 4.13),测试程序基于 CUDA 9.2 平台,使用 gcc 7.2 编译器进行编译。

3.1 加解密与自举过程测试

在实验过程中,首先对单比特输入进行加解密操作并测试时间,对于每比特的输入输出,加密和解密所需平均时间均不足 0.01 ms。然后对单比特自举过程的时间进行测试,结果显示,自举过程所需时间为 3 ms。

3.2 单比特比较算法

在实验过程中,对多组 0、1 输入进行加密,对密文做同态比较运算,测量总时间后求平均值得到运算时间,结果显示,对单比特明文进行同态比较运算所需的时间为 10 ms。

3.3 多比特比较算法

在实验过程中,测量在 GPU(CUDA 平台)上同态比较器的运行时间。针对不同长度的二进制明文比特串,同态比较器运算所需的时间如表 4 所示。

表 4 多比特同态比较器运行时间结果

明文长度/bit	时间/ms	明文长度/bit	时间/ms
10	133.9	60	556.6
20	187.7	70	680.3
30	320.2	80	733.8
40	379.5	90	855.8
50	505.1	100	910.5

比较器运算时间与明文长度的关系如图 1 所示。

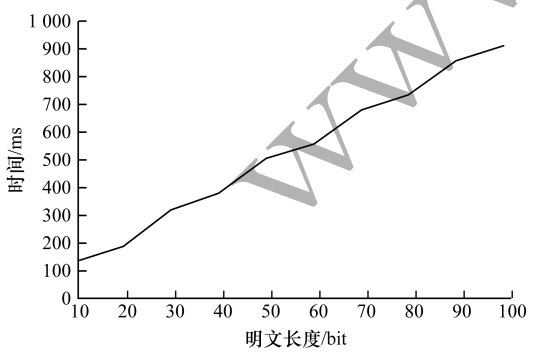

图 1 比较器运算时间与明文长度的关系

从图 1 可以看出,基于 CPU 多线程技术与 GPU 硬件,在 1 s 内可完成对 100 bit 明文的同态比较运算,且比较器支持对任意比特长度的明文进行同态比较运算。具体应用时对电路规模进行比较的需求通常不大,本文所设计的同态比较器基本能够满足实际需求。

4 结束语

本文在 cuFHE 全同态加密库的基础上,设计一种高效的多比特同态比较器。实验结果表明,该同态比较器可比较任意比特长度的明文,且运算速度较高。下一步将降低自举过程的计算量,以提升该比较器的运算效率。

参考文献

[1] 陈智罡. 基于格的全同态加密研究与设计[D]. 南京:南京航空航天大学,2015.

[2] GENTRY C. Fully homomorphic encryption using ideal lattices[C]//Proceedings of the 41st Annual ACM Symposium on Theory of Computing. New York, USA: ACM Press, 2009: 169-178.

[3] BRAKERSKI Z, VAIKUNTANATHAN V. Fully homomorphic encryption from Ring-LWE and security for key dependent messages[C]//Proceedings of the 31st Annual Cryptology Conference on Advances in Cryptology. Berlin, Germany: Springer, 2011: 15-23.

[4] GENTRY C, SAHAI A, WATERS B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based[C]//Proceedings of CRYPTO'13. Berlin, Germany: Springer, 2013: 75-92.

[5] CHILLOTTI I, GAMA N, GEORGIEVA M, et al. Faster fully homomorphic encryption: bootstrapping in less than 0.1 seconds[C]//Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Germany: Springer, 2016: 3-33.

[6] CHILLOTTI I, GAMA N, GEORGIEVA M, et al. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE[C]//Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Germany: Springer, 2017: 377-408.

[7] 李宗育,桂小林,顾迎捷,等. 同态加密技术及其在云计算隐私保护中的应用[J]. 软件学报,2018,29(7):8-29.

[8] 杨晓元,丁义涛,周潭平. 基于 CPU 多核的 FHEW 并行算法[J]. 密码学报,2017,4(6):620-626.

[9] 游林,梁家豪. 基于同态加密与生物特征的安全身份认证研究[J]. 信息网络安全,2018,18(4):1-8.

[10] 蒋林智,许春香,王晓芳,等. 全同态加密在基于密文计算模型中的应用[J]. 密码学报,2017,4(6):596-610.

[11] 孙爽,吴文渊,王会勇. 基于 HElib 的并行多比特明文同态比较模型[J]. 计算机应用,2015,35(S2):53-56,69.

(下转第 152 页)

（上接第 146 页）

[12] OWENS J D, HOUSTON M, LUEBKE D, et al. GPU computing[J]. Proceedings of the IEEE, 2008, 96(5): 879-899.

[13] 李传朋，秦品乐，张晋京. 基于深度卷积神经网络的图像去噪研究[J]. 计算机工程，2017，43(3)：253-260.

[14] 杨志刚，吴俊敏，徐恒，等. 基于虚拟化的多 GPU 深度神经网络训练框架[J]. 计算机工程，2018，44(2)：68-74，83.

[15] 杨鑫，许端清，赵磊. 基于多核架构的大图像实时浏览技术[J]. 中国图象图形学报，2018，16(2)：152-160.

[16] 卢兴敬，刘雷，贾海鹏，等. ParaC：面向 GPU 平台的图像处理领域的编程框架[J]. 软件学报，2017，28(7)：1655-1675.

[17] 温万里，游林. 基于混沌和比特级置乱的并行图像加密算法[J]. 信息网络安全，2014，14(4)：40-45.

[18] 成娟娟，郑昉昱，林璟锵，等. Curve25519 椭圆曲线算法 GPU 高速实现[J]. 信息网络安全，2017，17(9)：122-127.

[19] CUDA-accelerated fully homomorphic encryption library [EB/OL]. [2018-12-20]. https://github.com/vernamlab/cuFHE.

[20] DAI Wei, SUNAR B. cuHE：a homomorphic encryption accelerator library [C]//Proceedings of International Conference on Cryptography and Information Security in the Balkans. Berlin, Germany：Springer, 2015：169-186.

编辑　吴云芳