全同态加密是否完美?

王付群

云计算一经提出，便成为人们关注的焦点。它拥有强大的计算能力，可以帮助人们执行复杂的计算。但是，在保护用户数据私密性的前提下，如何利用云计算的强大计算能力是云计算从理论走向实用必须解决的关键问题。在此迫切需求下，全同态加密如约而至。

全同态加密允许用户通过加密保护数据的私密性，同时允许服务器（比如“云”）对密文执行任意可计算的运算，得到的结果是对相应明文执行相应运算结果的某个有效密文。由此可见，全同态加密完美地解决了云计算环境下的安全计算问题。

早在 1978 年，研究人员就提出了全同态加密的思想，但是一直没有设计出一个具体方案。全同态加密成为密码界的一个公开问题，被誉为“密码学圣杯”。2009 年，Gentry 创造性地构造了第一个全同态加密方案，摘取了密码学圣杯。之后，全同态加密迅速吸引了一批资深专家、学者对之进行广泛、深入的研究，并取得了一系列的成果。为了设计全同态加密，人们提出了很多技术、方法，如压缩解密电路（Squashing Decryption Circuit）、自举（Bootstrapping）、再线性化（Relinearizition）、模数转换（Modulus-Switching）、维数转换（Dimension-Switching）、近似特征向量（Approximate Eigenvector）等。下面从构造技术的发展来分类介绍全同态加密的研究进展。

1. 第一代全同态加密

2009 年，Gentry [1] 取得突破性进展，构造出第一个全同态加密方案（Fully Homomorphic Encryption, FHE）摘取了“密码学圣杯”。Gentry 设计了一个构造全同态加密方案的“蓝图”：首先构造一个类同态加密（Somewhat Homomorphic Encryption, SHE）方案（可同态计算一定深度的电路[①]，但同态能力达不到解密电路的深度）；然后压缩解密电路，使得它能够同态计算它本身，即得到一个可自举[②]的同态加密 (Bootstrappable HE) 方案；最后有序执行自举操作，得到一个可以同态计算任意电路的方案，即全同态加密方案。同时，基于理想格上的困难问题，并结合循环安全与稀疏子集和（Sparse Subset

①算法可用电路来实现，电路深度指电路中任意路径所含有的最大的电路门数。

② 自举：利用密钥的密文（称为计算密钥）来计算解密电路，运行过程完全在密文状态下，得到的结果仍然是密文。

Sum）假设，Gentry 也开创性地构造了一个具体的方案。

随后，van Dijk 等人 [2] 提出了一个整数上的全同态加密方案，这个设计完全模仿了 Gentry 的蓝图。该方案的安全性基于近似最大公因子假设。它的优点在于概念简单，易于理解，其缺点在于公钥太大。

以上的方案被称为第一代全同态加密方案。

2. 第二代全同态加密

随着 Gentry 全同态加密方案的提出，人们开始尝试基于格上的困难问题（比如带错学习问题（Learning with Errors，LWE），环上的带错学习问题（Learning with Errors over Ring，RLWE））构造全同态加密方案，并结合理想格的代数结构、快速运算等优良性质来进行方案的优化和实现，最终取得了巨大的成功。

2011 年，Brakerski 和 Vaikuntanathan [3,4] 基于 LWE 与 RLWE 分别提出了全同态加密方案，其核心技术是再线性化和模数转换。这些新技术的出现使得我们无需压缩解密电路，从而也就不需要稀疏子集和假设，这样方案的安全性完全基于 (R)LWE 的困难性。Brakerski 和 Vaikuntanathan [4] 还提出了循环安全的类同态加密方案。但是，他们的方案不能够利用自举以达到全同态的目的，这是因为他们所得到的循环安全是相对于私钥作为环元素表示的，而不是自举算法所需要的比特表示。构造循环安全的可自举的同态加密至今依然是一个公开问题。

Brakerski 等人 [5] 指出：依次使用模数转换能够很好的控制噪音的增长。据此他们设计了一个层次型的全同态加密方案：BGV。BGV 可以同态计算任意多项式深度的电路③，从而在实际应用中无需利用计算量过大的自举来提升同态计算能力。

研究人员对 BGV 方案做了大量的优化、实现 [6]，对 BGV 方案的研究越来越深刻、完善，效率也越来越高。目前来看，（优化后的）BGV 方案是最高效的全同态加密方案之一。

以上方案与第一代方案相比，无需压缩解密电路，也就不需要稀疏子集和假设，方案的安全性得到极大的提升，故被称为第二代全同态加密方案。

3. 第三代全同态加密

上述所有方案无论是层次型的还是纯的全同态加密，都需要“计算密钥”（私钥信息的加密，可以看做公钥的一部分）的辅助才能达到全同态的目的。但是计算密钥的尺寸一般来说都很大，是制约全同态加密效率的一大瓶颈。

2013 年，Gentry 等人 [7] 利用“近似特征向量”技术，设计了一个无需计算密钥的全同态加密方案：GSW，标志着第三代全同态加密方案的诞生。他们进而还设计了基于身份

③ 电路所有路径所含有的电路门数至多是一个关于安全参数的多项式。

和基于属性的全同态加密方案，掀起了全同态加密研究的一个新高潮。此后，研究人员在高效的自举算法、多密钥全同态加密、CCA1 安全的全同态加密和电路私密的全同态加密等方面进行了大量的研究，得到了丰硕的成果。下面逐一介绍。

3.1 高效的自举算法

Alperin-Sheriff 和 Peikert [8] 基于 GSW，提出了一个简单的 GSW 方案，并用之设计了一个快速的自举算法：直接同态计算解密函数。该自举方法直接、简单、高效，向实际应用迈出了坚实的一步。

Ducas 和 Micciancio [9] 在 [8] 的基础上，利用一个变形的基于 RLWE 的 GSW 方案来直接同态计算解密函数，大大提升了效率，他们的测试表明：1 秒内就可以完成一次自举过程。Chillotti 等人 [10] 改进了 [9] 的方案，在自举时，他们巧妙地用矩阵与向量间的运算来代替矩阵与矩阵间的运算，有效地降低了自举所花费的时间：0.1 秒内就可以完成一次自举过程。从公开发表的文献来看，这是目前最高效的自举方案。

3.2 多密钥全同态加密

López-Alt 等人 [11] 最先开始研究多密钥全同态加密，他们基于 NTRU 构造了第一个多密钥全同态加密，并利用它设计了一个多方安全计算协议。但是，他们的方案用到了一个非标准的假设，并且近年来也遭受到比较严重的攻击。所以设计安全的多密钥全同态加密引起了人们的注意，并研究出多个基于 LWE 的多密钥全同态加密 [12,13]。其中，[MW16] 的多密钥全同态加密方案的密文会随着不同的密钥数的增长而膨胀，而且同态计算后的密文不能继续执行同态运算。Brakerski 等人 [13] 提出了完全动态的多密钥全同态加密，基本解决了上述两个问题。但是，他们利用了笨重的自举技术，并不实用。

3.3 CCA1 安全的全同态加密

CCA2 安全对于加密来说已经成为标准的安全性要求。遗憾的是 CCA2 安全与同态性质是矛盾的，不可能同时实现。但是，CCA1 安全与同态性质并无矛盾之处，它们可以共存。Canetti 等人 [14] 研究了 CCA1 安全的全同态加密方案，给出了 3 个构造：前两个都是由多身份全同态加密转化而来（我们也提出了这个转化方式，遗憾的是未能发表），并构造了两个多身份全同态加密方案，第三个使用了 SNARKs，得到了一个紧凑的方案。前两个构造的缺点是密文不紧凑，第三个构造建立在非标准的假设上。

3.4 电路私密的全同态加密

在全同态加密领域，有时不但要保护好数据的私密性，而且要保护好电路的私密性。电路的私密性是指同态计算出来的密文不泄露电路的任何信息，也就是说只有执行同态运算的人才知道电路，而其他人（包括解密者）都不能从同态计算出来的密文挖掘出电路的信息。

Gentry[1] 在提出全同态加密的时候就已经考虑了这个问题，他建议在输出同态计

算密文之前，给它增加一个大的噪音，完全掩盖该密文所隐含的同态计算所积累的噪音。这一方法的缺点也是明显的：这样的密文所含的噪音太大，故不能再对它执行同态运算了。

Ducas 等人 [15] 多次调用自举算法来控制同态计算后的噪音分布，使得同态计算后的密文的噪音分布与新鲜密文的噪音分布是统计不可区分的。他们的方法可以运用到所有（理想）格全同态加密方案 [1,5,7]。这个方法依赖于高效的自举算法，目前并不可行。

在短短的几年内，国际上在全同态加密技术方面已经取得了丰硕的成果，全同态加密也从第一代发展到了第三代，其效率与安全性都得到了极大地提升。但是离实际应用还有很大的差距，值得我们全面、深入的研究。

附：参考文献

[1] L. Ducas and D. Micciancio. FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In EUROCRYPT (I) 2015, LNCS vol. 9056, pp. 617–640.Springer, 2015.

[2] Z. Brakerski and V. Vaikuntanathan. Fully Homomorphic Encryption from Ring–LWE and Security for Key Dependent Messages.In CRYPTO 2011, LNCS vol. 6841, pp. 505–524.Springer, 2011.

[3] A. L ó pez–Alt, E. Tromer and V. Vaikuntanathan. On–the–fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. In STOC 2012, pp. 1219–1234.ACM, 2012.

[4] Z.Brakerski and R. Perlman. Lattice–Based Fully Dynamic Multi–Key FHE with Short Ciphertexts.In CRYPTO 2016.

[5] Z. Brakerski and V. Vaikuntanathan..Efficient Fully Homomorphic Encryption from (Standard) LWE.In FOCS 2011, pp. 97–106.

[6] C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices.In STOC 2009, pp. 169–178.ACM, 2009.

[7] P. Mukherjee and D. Wichs. Two Round Multiparty Computation via Multi–Key FHE. In EUROCRYPT 2016. Available at http://eprint.iacr.org/2015/345.

[8] L. Ducas and D. Stehl é . Sanitization of FHE Ciphertexts.In EUROCRYPT 2016, pp. 294–310.Springer 2016. Available at http://eprint.iacr.org/2016/164.

[9] M. van Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan.Fully Homomorphic Encryption over the Integers.In EUROCRYPT 2010, LNCS vol. 6110, pp. 24–43.Springer, 2010.

[10] J. Alperin–Sheriff and C. Peikert. Faster Bootstrapping with Polynomial Error.In CRYPTO (I) 2014, LNCS vol. 8616, pp. 297–314.Springer, 2014.

[11] R. Canetti, S.Raghuraman, S.Richelson and V. Vaikuntanathan. Chosen–Ciphertext Secure FullyHomomorphic Encryption.In PKC 2017, Part II, LNCS 10175, pp. 213–240, 2017.

[12] I. Chillotti, N. Gama, M. Georgieva and M.Izabach è ne. Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds. In ASIACRYPTO2016.

[13] C. Gentry, A. Sahai and B. Waters. Homomorphic Encryption from Learning with Errors: Conceptually–Simpler, Asymptotically–Faster, Attribute–Based. In CRYPTO 2013, LNCS vol. 8042, pp. 75–92.

[14] S. Halevi and V. Shoup. Bootstrapping for HElib.In EUROCRYPT (I) 2015, LNCS vol. 9056, pp. 641–670. Springer, 2015.

[15] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully Homomorphic Encryption without Bootstrapping.In ITCS 2012, pp. 309–325.