FHE reading guide

August 26, 2020

Vinod Vaikuntanathan and Daniele Micciancio both keep a webpage with important (theoretical) FHE papers https://people.csail.mit.edu/vinodv/FHE/FHE-refs.html & http://cseweb.ucsd.edu/~daniele/LatticeLinks/FHE.html. The following ones are ones I am reading, I tried to substitute theoretical papers with ones that look at the schemes from an implementational point of view wherever possible.

1 Papers

1.1 Pre-FHE & Gen-I FHE

This is mostly historical. The assumptions and techniques used are different than those used today, and these schemes are very inefficient.

1.2 Gen-II FHE

These are still widely-used and implemented. They are based on (R)LWE assumptions.

- BGV [BGV12] (based on ideas from [BV11])
- (B)FV [FV12] (a port from LWE to RLWE of [Bra12])

These schemes allow *packing* (also *batching*, SIMD) of multiple plaintexts in ciphertext *slots* for higher throughput. There is an approach based on the CRT for RLWE schemes [SV14] and a different approach for LWE schemes [BGH12].

BGV was used to homomorphically evaluate the AES circuit [GHS12]. The paper describes some optimisations of BGV that are implemented in most libraries.

There are two separate approaches to implement (B)FV in full-RNS representation. This paper [BPA+18] talks about implementing both of them in PALISADE.

The LTV [LATV12] and YASHE [BLLN13] second generation schemes based on NTRU were quite popular as well, but have since been broken. This paper [LN14] compares (B)FV with YASHE from an implementational point of view.

1.3 Gen-III FHE

The third-generation starts with GSW, which uses a different approach than the second-generation schemes. GSW is quite inefficient, but remarkably simple and allows advanced constructions. GSW is often used in tutorials on FHE due to its simplicity. GSW ciphertexts are used by FHEW and TFHE during bootstrapping.

FHEW introduced the "gate bootstrapping" idea which allows for fast bootstrapping by bootstrapping after every single gate. In a way, TFHE can be seen as porting FHEW, which uses both normal LWE and ring-GSW, to the Torus. One of the merits of FHEW is that is quite easy to understand and implement, whereas TFHE appears to be much more complicated.

- GSW [GSW13]
- FHEW [DM15]
- TFHE ([CGGI18], journal version combining multiple works)

A recent paper compares FHEW with TFHE (if TFHE would use the ternary instead of binary secrets) and implements this in PALISADE [MP20]. The PALISADE team seems really opposed to the binary secret distribution used in TFHE.

1.4 Gen-IV FHE

The HEAAN/CKKS approach [CKKS17], which encrypts approximate numbers, is either seen as a new generation or a variant of Gen-II.

There have been several RNS and bootstrapping optimisations [CHK+18, CHK+19, CCS19].

2 Tutorials

There are a number of more tutorial-like resources for FHE.

The most extensive one is probably this one from Halevi [Hal17], it describes GSW in detail. There is also this overview by Brakerski [Bra], also describes GSW in detail. There is this general introduction to lattice-based crypto from Peikert [Pei16], also describes GSW near the end.

2.1 Videos

There are a lot of video lectures and seminars on FHE out there. Some of the best ones I found.

- The videos from the Simons lattice semester in 2020. They are quite down-to-earth.
  - A talk about BV11 (the precursor of BGV) (`https://simons.berkeley.edu/talks/advanced-lattice-based-cryptography-fhe-abe-etc`)
  - Tutorial on GSW (`https://simons.berkeley.edu/talks/tutorial-encrypted-computation-lattices`)
  - Ilaria on FHE and TFHE (`https://simons.berkeley.edu/talks/intro-fhe-and-tfhe`)
  - CKKS/HEAAN (`https://simons.berkeley.edu/talks/heaan-fhe`)
  - PALISADE (`https://simons.berkeley.edu/talks/palisade-lattice-library`)
  - ... much more on lattice-based crypto
- See (`http://cseweb.ucsd.edu/~daniele/LatticeLinks/FHE.html`, at the bottom) for a collection of more videos.

3 Standard

The FHE standard describes BGV, (B)FV and GSW `https://homomorphicencryption.org/standard/`.

4 Libraries

A full list of all the FHE libraries is maintained here [Sch19]. Looking at the manual of the libraries can sometimes quickly give you an understanding of the implementational aspects of a scheme. These are the most popular ones.

- Microsoft SEAL implements BFV and CKKS. There is only a real manual for an older version 2.3.1 of the library [Lai]
- HElib implements BGV and CKKS. Comes with two reports that describe the algorithms and bootstrapping used in the library [HS14a, HS14b]
- PALISADE implements BFV, BGV, CKKS, TFHE, FHEW. There is a manual at [noaa]
- FHEW. This is more like a proof-of-concept. It is a very small library, due to the simplicity of FHEW. [Duc20]
- TFHE [noab]

5 Hardware

The following is copied from my FWO proposal.

Naturally, the first FHE hardware architectures implement first-generation FHE schemes. Cao et al. [CMO+14] implement a variant of the van Dijk scheme using integer-NTT on FPGA, but assume unlimited bandwidth to off-chip memory. Wang et al. develop a 768K-bit multiplier[1] for a variant of Gentry's scheme and implement the result on both FPGA and ASIC [WH13, WHEW14]. Doröz et al. similarly proposed a million-bit integer multiplier [D13], and integrate it into an ASIC implementation of the same scheme [D15]. There are several implementations of the second-generation NTRU-based YASHE and LTV schemes [SRJV+15, PNPM15, DS15, CRS17], but the underlying FHE schemes have since been broken.

(R)LWE-based implementations remain secure to date. Migliore et al. [MSR+17] implemented a Karatsuba variant of the second-generation FV scheme. While computationally slower than NTT, this approach benefits from easier integration of second-generation packing techniques. The design is presented as a software library using AVX2 instructions, coupled with a fully pipelined hardware accelerator over PCIe. In a series of two works, Roy et al. [SRJV+18, SRTJ+19] accelerate the NTT variant of FV. The first work targets a large parameter

[1] The multiplier can take inputs up to 2^{768}. In comparison, modern CPUs can efficiently handle inputs up to 2^{64}.

set, incurring massive off-chip data transfers. In the second work, the authors target less complex cloud applications, such that sufficient on-chip memory is available and data transfers are minimized. This last design is implemented on an ARM+FPGA MPSoC. Finally, HEAX [RLPD19] is an NTT implementation of HEAAN on FPGA. The employed platform connects a host CPU and FPGA over PCIe, as well as requiring off-chip DRAM memory. To the best of our knowledge, no hardware implementations of third-generation schemes have appeared in the open literature.

References

[BGH12] Zvika Brakerski, Craig Gentry, and Shai Halevi. Packed Ciphertexts in LWE-based Homomorphic Encryption. Technical Report 565, 2012.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) Fully Homomorphic Encryption Without Bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM. event-place: Cambridge, Massachusetts.

[BLLN13] Joppe W. Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme. In Martijn Stam, editor, *Cryptography and Coding*, Lecture Notes in Computer Science, pages 45–64, Berlin, Heidelberg, 2013. Springer.

[BPA+18] Ahmad Al Badawi, Yuriy Polyakov, Khin Mi Mi Aung, Bharadwaj Veeravalli, and Kurt Rohloff. Implementation and performance evaluation of rns variants of the bfv homomorphic encryption scheme. Cryptology ePrint Archive, Report 2018/589, 2018. https://eprint.iacr.org/2018/589.

[Bra] Brakerski. Fundamentals of Fully Homomorphic Encryption. Technical report.

[Bra12] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, Lecture Notes in Computer Science, pages 868–886, Berlin, Heidelberg, 2012. Springer.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, October 2011. ISSN: 0272-5428.

[CCS19] Hao Chen, Ilaria Chillotti, and Yongsoo Song. Improved Bootstrapping for Approximate Homomorphic Encryption. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, Lecture Notes in Computer Science, pages 34–54, Cham, 2019. Springer International Publishing.

[CGGI18] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast Fully Homomorphic Encryption over the Torus. Technical Report 421, 2018.

[CHK+18] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for Approximate Homomorphic Encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, Lecture Notes in Computer Science, pages 360–384, Cham, 2018. Springer International Publishing.

[CHK+19] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. A Full RNS Variant of Approximate Homomorphic Encryption. In Carlos Cid and Michael J. Jacobson Jr., editors, *Selected Areas in Cryptography – SAC 2018*, Lecture Notes in Computer Science, pages 347–368, Cham, 2019. Springer International Publishing.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, Lecture Notes in Computer Science, pages 409–437, Cham, 2017. Springer International Publishing.

[CMO+14] Xiaolin Cao, Ciara Moore, Máire O'Neill, Neil Hanley, and Elizabeth O'Sullivan. High-Speed Fully Homomorphic Encryption Over the Integers. In Rainer Böhme, Michael Brenner, Tyler Moore, and Matthew Smith, editors, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 169–180, Berlin, Heidelberg, 2014. Springer.

[CRS17] David Bruce Cousins, Kurt Rohloff, and Daniel Sumorok. Designing an FPGA-Accelerated Homomorphic Encryption Co-Processor. *IEEE Transactions on Emerging Topics in Computing*, 5(2):193–206, April 2017.

[DM15] Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, Lecture Notes in Computer Science, pages 617–640, Berlin, Heidelberg, 2015. Springer.

[Duc20] Léo Ducas. lducas/FHEW, July 2020. original-date: 2014-12-03T22:07:13Z.

[D13] Yarkin Doröz, Erdinç Öztürk, and Berk Sunar. Evaluating the Hardware Performance of a Million-Bit Multiplier. In *2013 Euromicro Conference on Digital System Design*, pages 955–962, September 2013. ISSN: null.

[D15] Yarkın Doröz, Erdinç Öztürk, and Berk Sunar. Accelerating Fully Homomorphic Encryption in Hardware. *IEEE Transactions on Computers*, 64(6):1509–1521, June 2015.

[DS15] Yarkın Doröz, Erdinç Öztürk, Erkay Savaş, and Berk Sunar. Accelerating LTV Based Homomorphic Encryption in Reconfigurable Hardware. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems – CHES 2015*, Lecture Notes in Computer Science, pages 185–204, Berlin, Heidelberg, 2015. Springer.

[FV12] Junfeng Fan and Frederik Vercauteren. Somewhat Practical Fully Homomorphic Encryption. Technical Report 144, 2012.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic Evaluation of the AES Circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, Lecture Notes in Computer Science, pages 850–867, Berlin, Heidelberg, 2012. Springer.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, Lecture Notes in Computer Science, pages 75–92, Berlin, Heidelberg, 2013. Springer.

[Hal17] Shai Halevi. Homomorphic Encryption. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, Information Security and Cryptography, pages 219–276. Springer International Publishing, Cham, 2017.

[HS14a] Shai Halevi and Victor Shoup. Algorithms in HElib. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, Lecture Notes in Computer Science, pages 554–571, Berlin, Heidelberg, 2014. Springer.

[HS14b] Shai Halevi and Victor Shoup. Bootstrapping for HElib. Technical Report 873, 2014.

[Lai] Kim Laine. Simple Encrypted Arithmetic Library 2.3. page 34.

[LATV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1219–1234, New York, NY, USA, 2012. ACM. event-place: New York, New York, USA.

[LN14] Tancrède Lepoint and Michael Naehrig. A Comparison of the Homomorphic Encryption Schemes FV and YASHE. Technical Report 062, 2014.

[MP20] Daniele Micciancio and Yuriy Polyakov. Bootstrapping in FHEW-like Cryptosystems. Technical Report 086, 2020.

[MSR+17] Vincent Migliore, Cédric Seguin, Maria Méndez Real, Vianney Lapotre, Arnaud Tisserand, Caroline Fontaine, Guy Gogniat, and Russell Tessier. A High-Speed Accelerator for Homomorphic Encryption using the Karatsuba Algorithm. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):138:1–138:17, September 2017.

[noaa] doc/palisade_manual.pdf · release-v1.10.2 · PALISADE / PALISADE Development. Library Catalog: gitlab.com.

[noab] TFHE Fast Fully Homomorphic Encryption over the Torus.

[Pei16] Chris Peikert. A Decade of Lattice Cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[PNPM15] Thomas Pöppelmann, Michael Naehrig, Andrew Putnam, and Adrian Macias. Accelerating Homomorphic Evaluation on Reconfigurable Hardware. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems – CHES 2015*, Lecture Notes in Computer Science, pages 143–163, Berlin, Heidelberg, 2015. Springer.

[RLPD19] M. Sadegh Riazi, Kim Laine, Blake Pelton, and Wei Dai. HEAX: High-Performance Architecture for Computation on Homomorphically Encrypted Data in the Cloud. *arXiv:1909.09731 [cs]*, September 2019. arXiv: 1909.09731.

[Sch19] Jonathan Schneider. Awesome HE : Homomorphic Encryption Libraries, Software and Resources, November 2019. original-date: 2018-02-21T02:38:07Z.

[SRJV+15] Sujoy Sinha Roy, Kimmo Järvinen, Frederik Vercauteren, Vassil Dimitrov, and Ingrid Verbauwhede. Modular Hardware Architecture for Somewhat Homomorphic Function Evaluation. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware* *and Embedded Systems – CHES 2015*, Lecture Notes in Computer Science, pages 164–184, Berlin, Heidelberg, 2015. Springer.

[SRJV+18] Sujoy Sinha Roy, Kimmo Järvinen, Jo Vliegen, Frederik Vercauteren, and Ingrid Verbauwhede. HEPCloud: An FPGA-Based Multicore Processor for FV Somewhat Homomorphic Function Evaluation. *IEEE Transactions on Computers*, 67(11):1637–1650, November 2018.

[SRTJ+19] Sujoy Sinha Roy, Furkan Turan, Kimmo Jarvinen, Frederik Vercauteren, and Ingrid Verbauwhede. FPGA-Based High-Performance Parallel Architecture for Homomorphic Computing on Encrypted Data. In *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 387–398, February 2019. ISSN: 1530-0897.

[SV14] N. P. Smart and F. Vercauteren. Fully homomorphic SIMD operations. *Designs, Codes and Cryptography*, 71(1):57–81, April 2014.

[WH13] Wei Wang and Xinming Huang. FPGA implementation of a large-number multiplier for fully homomorphic encryption. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2589–2592, May 2013. ISSN: 2158-1525.

[WHEW14] Wei Wang, Xinming Huang, Niall Emmart, and Charles Weems. VLSI Design of a Large-Number Multiplier for Fully Homomorphic Encryption. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(9):1879–1887, September 2014.