PRACTICAL LATTICE-BASED CRYPTOGRAPHY IN PALISADE

YURIY POLYAKOV, NJIT AND DUALITY
YPOLYAKOV@DUALITYTECH.COM

INCLUDES NEW RESULTS OF JOINT WORK WITH
ANDREY KIM (NJIT) AND
ANTONIS PAPADIMITRIOU (DUALITY)

SIMONS INSTITUTE WORKSHOP
LATTICES: FROM THEORY TO PRACTICE
MAY 1, 2020

HIGH-LEVEL OVERVIEW

- Lattices and lattice schemes implemented in PALISADE
- FHE Algorithms (most of this talk)
  - A quick introduction to double-CRT/RNS
  - **BFV RNS variants (includes new algorithms)**
  - **CKKS RNS variants (includes new algorithms)**
  - **Comparison of BGV vs BFV**
  - **Comparison of FHEW vs TFHE**
  - Extension of FHE to multi-party scenarios: PRE and Threshold FHE
- Lattice gadget toolkit (a quick review)
  - Efficient trapdoor sampling in RNS
  - Subgaussian sampling in RNS
  - Protocols based on lattice trapdoor sampling

Yuriy Polyakov ypolyakov@dualitytech.com

BASIC FACTS ON LATTICES IMPLEMENTED IN PALISADE

- Both q-ary integer lattices and cyclotomic rings are supported.
- All ring implementations are based on power-of-two cyclotomic rings:

$$R_q = \mathbb{Z}_q / \langle x^n + 1 \rangle.$$

- General cyclotomic rings are also supported but are not currently used for any special functionality.
- All protocols are based on LWE and/or Ring LWE as the hardness assumption.
- For ring instantiations, modulus q is either a prime or a product of primes, with all primes congruent to $1 \bmod m$, where m is the cyclotomic order. This constraint is set to support efficient NTT.

Yuriy Polyakov ypolyakov@dualitytech.com

LATTICE SCHEMES IMPLEMENTED IN PALISADE

- Homomorphic Encryption
  - Brakerski/Fan-Vercauteren (BFV) scheme [Bra12, FV12]: 2 RNS variants
  - Brakerski-Gentry-Vaikuntanathan (BGV) scheme [BGV14]
  - Cheon-Kim-Kim-Song (CKKS) scheme [CKKS17]: 2 RNS variants
  - Ducas-Micciancio (FHEW) [DM15] and Chillotti-Gama-Georgieva-Izabachene (TFHE) [CGGI16] schemes
  - Stehle-Steinfeld scheme (based on NTRU) [SS11]
- Multi-Party HE Extensions
  - Proxy Re-Encryption [PRSV17]
  - Threshold FHE [AJLTVW12, LTV11]

Yuriy Polyakov ypolyakov@dualitytech.com

LATTICE SCHEMES IMPLEMENTED IN PALISADE (CONT'D)

- Schemes based on lattice trapdoors
  - Gentry-Peikert-Vaikuntanathan (GPV) Digital Signature and Identity-based Encryption [GPV08]
  - Ciphertext-policy Attribute-based Encryption (ABE) [ZZG12]
  - Key-policy ABE [BGG+14]
  - Conjunction obfuscation [BVWW16]
  - Token-based conjunction obfuscation [CC17]
  - Token-based branching program obfuscation [CC17, CVW18]

Yuriy Polyakov ypolyakov@dualitytech.com

INTRODUCTION TO RNS/DOUBLE-CRT: WHY IS IT IMPORTANT?

- The main HE schemes, such as BFV, BGV, and CKKS, work with large integers (polynomial coefficients), which are much larger than native word size. Many other non-HE primitives implemented in PALISADE also work with large integers.
  - Multiprecision arithmetic (implemented in software) is typically highly inefficient for these schemes.
- Instead we use Residue Number System (RNS) or Chinese Remainder Theorem (CRT) representation of polynomial coefficients for the following reasons:
  - RNS works with native (machine-word size) integers: faster (as high as 10x) than multi-precision integer arithmetic.
  - Runtime scales (quasi)linearly with integer size.
  - RNS dramatically improves memory locality.
  - Computations are easily parallelizable.
  - RNS supports efficient GPU/FPGA hardware implementations.

Yuriy Polyakov ypolyakov@dualitytech.com

INTRODUCTION TO RNS/DOUBLE-CRT: WHAT IS IT?

- In ring constructions, we use NTT to perform efficient polynomial multiplications (it is one of the "CRTs" in the Double-CRT concept).
- Large modulus q is also represented as a smooth integer

$$q = \prod_{i=1}^{k} q_i,$$

where q_i are pair-wise coprime, native integers (typically of size between 20 and 60 bits).

- The large numbers (polynomial coefficients) are represented using CRT as residues $\bmod q_i$. This CRT decomposition, which is often called the RNS representation, corresponds to the second "CRT" in the Double-CRT notion.

Yuriy Polyakov ypolyakov@dualitytech.com

CHALLENGES OF BFV SCHEME

While addition, multiplication, and automorphism in Double-CRT is trivial (done independently for each small residue), several procedures in BFV require special RNS algorithms to avoid the use of multiprecision arithmetic. These include:

- Scaling in decryption

$$m := \left[\left[\frac{t}{q} \cdot [\langle \mathbf{sk}, \mathbf{ct} \rangle]_q\right]\right]_t.$$

- Scaling in homomorphic multiplication (tensor product w/o modular reduction)

$$\mathbf{ct}^* := \left[\left[\frac{t}{q} \cdot \mathbf{ct_1} \otimes \mathbf{ct_2}\right]\right]_q.$$

- Ciphertext digit decomposition in key switching (e.g., in relinearization)

$$\mathbf{G}^{-1}(\mathbf{ct}[last]).$$

Yuriy Polyakov ypolyakov@dualitytech.com

BFV SCHEME IN RNS

- Two RNS variants of BFV are known (both are implemented in PALISADE)
  - The Bajard-Eynard-Hasan-Zucca (BEHZ) variant [BEHZ16] that uses only integer arithmetic (based on small Mongtomery reduction and auxiliary moduli).
  - The Halevi-Polyakov-Shoup (HPS) variant [HPS19] that uses both integer and floating-point arithmetic.
  - The main differences are in the RNS procedures used for decryption and homomorphic multiplication. RNS digit decomposition is done using the same technique.
- Until very recently, it was believed that
  - The HPS variant requires extended floating-point precision (long doubles and quadfloat) to support moduli > 45 bits.
  - The BEHZ variant has a higher noise growth than HPS [BPAVR19].

Yuriy Polyakov ypolyakov@dualitytech.com

HPS RNS VARIANT: MAIN DEFINITIONS

- We work with the following CRT reconstructions of a large integer x

$$x = \left(\sum_{i=1}^{k} [x_i \cdot \tilde{q}_i]_{q_i} \cdot q_i^* \right) - v \cdot q, \text{ where } v \in \mathbb{Z}$$

$$x = \left(\sum_{i=1}^{k} x_i \cdot \tilde{q}_i \cdot q_i^* \right) - v' \cdot q, \text{ where } v' \in \mathbb{Z}$$

$$q_i^* = q/q_i \in \mathbb{Z}_q, \;\; \tilde{q}_i = (q_i^*)^{-1} (\mathrm{mod}\, q_i) \in \mathbb{Z}_{q_i}$$

Yuriy Polyakov ypolyakov@dualitytech.com

HPS RNS VARIANT: SCALING IN DECRYPTION

$$y := \left[\left\lfloor \frac{t}{q} \cdot x \right\rceil\right]_t = \left[\left\lfloor \left(\sum_{i=1}^{k} x_i \cdot \tilde{q}_i \cdot q_i^*\right) \cdot \frac{t}{q} - v' \cdot q \cdot \frac{t}{q} \right\rceil\right]_t = \left[\left\lfloor \left(\sum_{i=1}^{k} x_i \cdot \left(\tilde{q}_i \cdot \frac{t}{q_i}\right)\right) \right\rceil\right]_t$$

We separate integer and fractional parts (both are pre-computed):

$$t\frac{\tilde{q}_i}{q_i} = \omega_i + \theta_i, \text{ where } \omega_i \in \mathbb{Z}_t \text{ and } \theta_i = \left[-\frac{1}{2}, \frac{1}{2}\right)$$

Main challenge is in the approximation error accumulating from rounding $\sum_{i=1}^{k} x_i \cdot \theta_i$

The goal is to choose such q_i that

$$\sum_{i=1}^{k} x_i \cdot \varepsilon_i < \frac{1}{4}, \text{ where } |\varepsilon_i| \leq 2^{-52}$$

If we limit k to 32, then the highest modulus bit size supported with doubles is 45.

Yuriy Polyakov ypolyakov@dualitytech.com

HPS VARIANT: NEW ALGORITHM (JOINT WORK WITH ANDREY KIM)

- Inspired by the Brakerski-Vaikuntanathan digit decomposition technique, originally proposed for key switching in the BV scheme [BV11].
- We decompose x_i into high and low digits:

$$x_i = x_i^{(h)} \cdot 2^r + x_i^{(l)}$$

- Precompute at parameter generation

$$t\frac{\tilde{q}_i}{q_i} = \omega_i^{(l)} + \theta_i^{(l)}, \text{ where } \omega_i^{(l)} \in \mathbb{Z}_t \text{ and } \theta_i^{(l)} = \left[-\frac{1}{2}, \frac{1}{2}\right)$$

$$t\frac{[2^r\tilde{q}_i]_{q_i}}{q_i} = \omega_i^{(h)} + \theta_i^{(h)}, \text{ where } \omega_i^{(h)} \in \mathbb{Z}_t \text{ and } \theta_i^{(h)} = \left[-\frac{1}{2}, \frac{1}{2}\right)$$

- During decryption, we compute

$$\sum_{i=1}^{k} x_i \cdot \left(t\frac{\tilde{q}_i}{q_i}\right) = \sum_{i=1}^{k} x_i^{(l)} \cdot \left(\omega_i^{(l)} + \theta_i^{(l)}\right) + x_i^{(h)} \cdot \left(\omega_i^{(h)} + \theta_i^{(h)}\right)$$

Yuriy Polyakov ypolyakov@dualitytech.com

HPS VARIANT: NEW ALGORITHM (CONT'D)

- The approximation error is now

$$\sum_{i=1}^{k} x_i^{(l)} \cdot \varepsilon_i^{(l)} + x_i^{(h)} \cdot \varepsilon_i^{(h)}, \text{ where } \left|\varepsilon_i^{(l)}\right|, \left|\varepsilon_i^{(h)}\right| \leq 2^{-52}$$

- We can easily support 60-bit (or higher) moduli using *doubles* by setting r to 30 bits as the sum will always be much smaller than 1/4.
- The technique is also efficient as we can perform only regular integer multiplications (no need for intermediate modular reductions in all practical cases).
- The technique is readily extendible to any size/number of digits and can be used to reduce the floating-point approximation error in any scenario where intermediate floating-point computations are used for modular integer computations.

Yuriy Polyakov ypolyakov@dualitytech.com

HPS VARIANT: CRT BASIS EXTENSION

- Extend to modulus p

$$[x]_p = \left[\left(\sum_{i=1}^{k} [x_i \cdot \tilde{q}_i]_{q_i} \cdot q_i^*\right) - v \cdot q\right]_p.$$

- Estimate v using floating-point arithmetic

$$v = \left\lceil \left(\sum_{i=1}^{k} [x_i \cdot \tilde{q}_i]_{q_i} \cdot q_i^*\right) / q \right\rfloor = \left\lceil \sum_{i=1}^{k} \frac{[x_i \cdot \tilde{q}_i]_{q_i}}{q_i} \right\rfloor.$$

- Compute

$$[x]_p = \left[\left(\sum_{i=1}^{k} [x_i \cdot \tilde{q}_i]_{q_i} \cdot [q_i^*]_p\right) - v \cdot [q]_p\right]_p$$

- The approximation error introduced by floating-point computations does not practically affect the noise in the homomorphic multiplication (less than 1 bit)

Yuriy Polyakov ypolyakov@dualitytech.com

HPS BFV VARIANT: CURRENT SINGLE-THREADED RESULTS

L	*n*	$\log_2 q$	*k*	*Encryption [ms]*	*Addition [ms]*	*Multiplication [ms]*	*Rotation [ms]*	*Decryption [ms]*
1	2^{11}	50	1	0.42	0.008	1.84	0.31	0.09
3	2^{12}	100	2	1.18	0.026	5.90	0.66	0.31
5	2^{13}	180	3	3.10	0.077	17.7	2.82	0.88
10	2^{13}	240	4	7.77	0.218	48.6	9.80	2.21
20	2^{14}	420	7	12.2	0.425	98.8	28.4	3.76
30	2^{15}	660	11	38.4	1.53	383	144	12.3
50	2^{16}	1020	17	118	5.72	1,556	717	39.0
100	2^{17}	2100	35	510	24.9	11,252	6,435	171

From the practical perspective, the BFV implementation in PALISADE is now used in a commercial product of Duality (for an encrypted SQL-like query).

Yuriy Polyakov ypolyakov@dualitytech.com

NOTES ON COMPARISON OF HPS AND BEHZ RNS VARIANTS

- Thanks to Vincent Zucca and Ahmad Q. A. Al Badawi, PALISADE has a highly optimized implementation of the BEHZ variant, and we can perform a fair comparison of both variants.
- Based on the current implementation of both in PALISADE and complexity analysis
  - The runtimes are approximately the same (with differences only within 10%).
  - The noise growth is the same in both variants (as recently suggested in [MENSZ19]).
  - The only significant (usability) difference is that HPS is simpler, and easier to implement.
- It should be noted that both variants can co-exist and their RNS procedures can be mixed in the same implementation of BFV.

Yuriy Polyakov ypolyakov@dualitytech.com

MAIN CHALLENGE OF CKKS SCHEME IN RNS: RESCALING

- Modulus switching, which is used for rescaling in CKKS, is defined as

$$[\mathbf{c}']_q = \left\lceil \frac{q}{Q} \cdot [\mathbf{c}]_Q \right\rfloor.$$

- In CKKS rescaling, $\Delta = Q/q$ corresponds to the scaling factor for encoding the floating-point data.
- In original (multiprecision) CKKS, Δ is typically 2^p , and both Q and q are a power of 2.
- In RNS variants of HE schemes, Q and q are decomposed into products of small primes to support fast NTT, and q typically divides Q.
- Rescaling requires scaling down by 2^p but we cannot choose such q and Q that will give us 2^p almost exactly. Our best approximation is a small prime Q/q that is close to 2^p.

Yuriy Polyakov ypolyakov@dualitytech.com

"APPROXIMATE" RESCALING RNS VARIANT OF CKKS

- Initial CKKS RNS variants, such as [CHKKS18] and [BGPRV19], treat the rescaling approximation error as extra "noise" in the CKKS scheme.
- The RNS moduli q_i are chosen such that $2^p/q_i$ stays in the range $(1-2^{-\varepsilon}, 1+2^{-\varepsilon})$, where $2^{-\varepsilon}$ is kept as small as possible.
- When we perform the first multiplication and rescale operation, the effective scaling factor changes from 2^p to $2^{2p}/q_L$.
  - We introduce an approximation error of roughly $p-\varepsilon$ bits.
- If we alternate the moduli around 2^p, i.e., $q_{i-1} < 2^p$ and $q_i > 2^p$, we can reduce the approximation error after multiple levels.
- Still, the error keeps growing as we increase the ring dimension and depth of computation, and the rescaling approximation error dominates over other CKKS noise (typically exceeding it by several bits). Can we avoid this approximation error?

Yuriy Polyakov ypolyakov@dualitytech.com

“EXACT” RESCALING RNS VARIANT OF CKKS (JOINT WORK WITH ANTONIS PAPADIMITRIOU)

- Yes, we can eliminate the approximation error but the cost can be too high.
  - The rescaling error is deterministic and depends only on 2^p and q_i’s.
  - Hence we can always adjust the scaling factor by performing an extra scalar multiplication.
- A naïve solution is to multiply by a scaling factor after every multiplication.
  - While preserving the original scaling factor 2^p, this seems to double the number of levels.
- Can we do better than this?

Yuriy Polyakov ypolyakov@dualitytech.com

"EXACT" RESCALING RNS VARIANT OF CKKS (JOINT WORK WITH ANTONIS PAPADIMITRIOU) – CONT'D

Challenge 1

How to efficiently adjust the scaling factors, minimizing the performance cost?

Our Solution

Assign a different scaling factor to each level:

- Use the "natural order" of CKKS

Level	Scaling factor
L	q_L
L-1	q_L
L-2	${q_L}^2/q_{L-1}$
L-3	$({q_L}^2/q_{L-1})^2/\,q_{L-2}$
…	…

- Avoids any scalar multiplication for ciphertexts following the natural order.
- We need to do scalar multiplication only for ciphertexts at different levels.

Yuriy Polyakov ypolyakov@dualitytech.com

“EXACT” RESCALING RNS VARIANT OF CKKS (JOINT WORK WITH ANTONIS PAPADIMITRIOU) – CONT’D

Challenge 2

How to hide scaling factor adjustments from the user, i.e., do them automatically?

Our Solution

- Keep track of the scaling factors for each level
- Automatically adjust the scaling factor when needed
- Automatically rescale to enforce the natural order of scaling factors

Yuriy Polyakov ypolyakov@dualitytech.com

“EXACT” RESCALING RNS VARIANT OF CKKS (JOINT WORK WITH ANTONIS PAPADIMITRIOU) – CONT’D

Challenge 3

What logic should we use for automated rescaling?

Our Solution

Rescale right before the next multiplication.

- Perform all computations between two multiplications using the “noise budget” of the first multiplication. So these intermediate operations are almost noise-free.
- There is a performance penalty as most operations work with an effective scaling factor of $\sim 2^{2p}$ (and we keep one extra modulus in the CRT basis of the current ciphertext), but it is relatively small for most cases.

Yuriy Polyakov ypolyakov@dualitytech.com

“EXACT” RESCALING RNS VARIANT OF CKKS (JOINT WORK WITH ANTONIS PAPADIMITRIOU) – CONT’D

Challenge 4

We noticed that the natural-order scaling factors start diverging either towards 0 or infinity after a certain number of levels (like 20), hence preventing the use of this RNS variant for deeper computations.

Our Solution

Choose q_i’s so that the scaling factor changes as little as possible from one level to the next.

SelectModuli(n,L,p):
- $q_L :=$ FirstPrime(p,n)
- $q_{next} := q_L$
- $q_{prev} := q_L$
- $sf_L := q_L; sf_{L-1} := q_L$
- $ctr := 0$
- **for** $i = L-2, \ldots, 1$
  - $sf_i := \frac{(sf_{i+1})^2}{q_{i+1}}$
  - **if** $ctr \bmod 2 = 0$
    - $q_{prev} := \lfloor sf_i \rceil - 2n - [\lfloor sf_i \rceil]_{2n} + 1$
    - $q_i :=$ PreviousPrime(q_{prev}, n)
  - **else**
    - $q_{next} := \lfloor sf_i \rceil + 2n - [\lfloor sf_i \rceil]_{2n} + 1$
    - $q_i :=$ NextPrime(q_{next}, n)
  - $ctr := ctr + 1$
- $q_0 :=$ PreviousPrime(60,n)
- **return q**

Yuriy Polyakov ypolyakov@dualitytech.com

“EXACT” RESCALING RNS VARIANT OF CKKS (JOINT WORK WITH ANTONIS PAPADIMITRIOU) – SUMMARY

Exact vs Approximate RNS Rescaling

- The rescaling error in the exact RNS variant is similar to the error in the “textbook” CKKS. In the approximate RNS variant, the rescaling error is often higher than CKKS noise.
  - Net effect: higher output precision (same as in “textbook” CKKS) can be supported.
- Rescaling and “modulus switching” (level reduction without changing the scale) are done automatically in the exact RNS variant. Rescaling is performed manually in the approximate RNS variant.
- The exact RNS variant is typically 1.1x to 1.5x slower (depending on the computations) than the approximate variant due to extra scalar multiplications and “delayed” rescaling. But it is much simpler for the user and provides a higher precision.

Yuriy Polyakov ypolyakov@dualitytech.com

KEY SWITCHING IN CKKS AND BGV

Key switching plays a crucial role in CKKS and BGV (and to some extent in BFV). It is often the main performance bottleneck of many computations. We consider the following two techniques as most practical options for RNS variants of CKKS and BGV:

- The RNS decomposition technique proposed in [BEHZ16] (similar in concept to Brakerski-Vaikuntanathan (BV) digit decomposition [BV11])
  - Often faster for a small number of levels.
  - May introduce significant noise when performing rotations.
- Hybrid key switching (a hybrid of Gentry-Halevi-Smart key switching [GHS12] and RNS decomposition) [HK20]
  - Often faster for deeper computations.
  - Introduces auxiliary moduli (thus increasing the effective ciphertext modulus used in estimating the LWE work factor).
  - Generally has lower noise growth.

Yuriy Polyakov ypolyakov@dualitytech.com

COMPARISON OF BGV VS BFV

Comparison of the single-threaded runtime for BGV and BFV to compute a chain of $2^{depth} - 1$ homomorphic multiplications (using the binary tree approach).
The security level was set to 128 bits according to the HomomorphicEncryption.org security standard.

t	depth	BGV [ms]	BFV [ms]	BGV Speedup
2	2	4.8	35.2	7.3x
2	3	34.8	82.1	2.4x
2	4	112	261	2.3x
2	5	318	554	1.7x
2	6	872	1,136	1.3x
65537	2	9.7	51.6	5.3x
65537	3	35.1	121	3.4x
65537	4	256	730	2.9x
65537	5	740	1,508	2.0x
65537	6	1,999	4,060	2.0x

Yuriy Polyakov ypolyakov@dualitytech.com

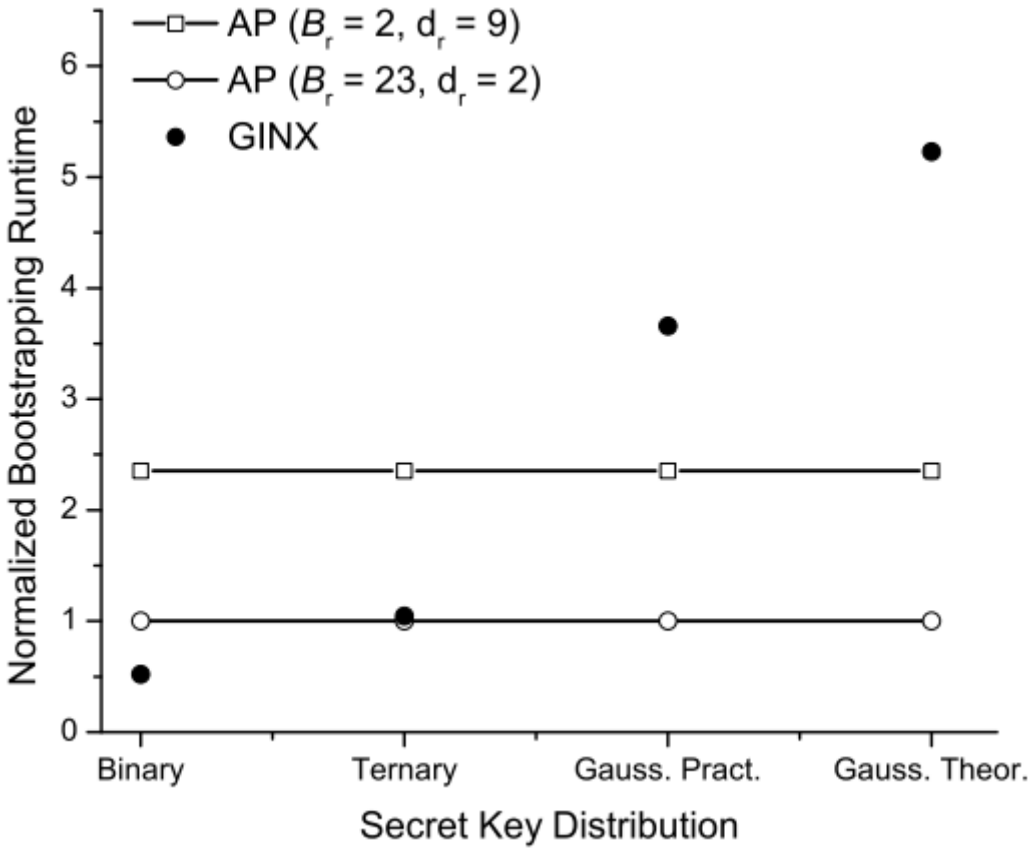
FHEW VS TFHE

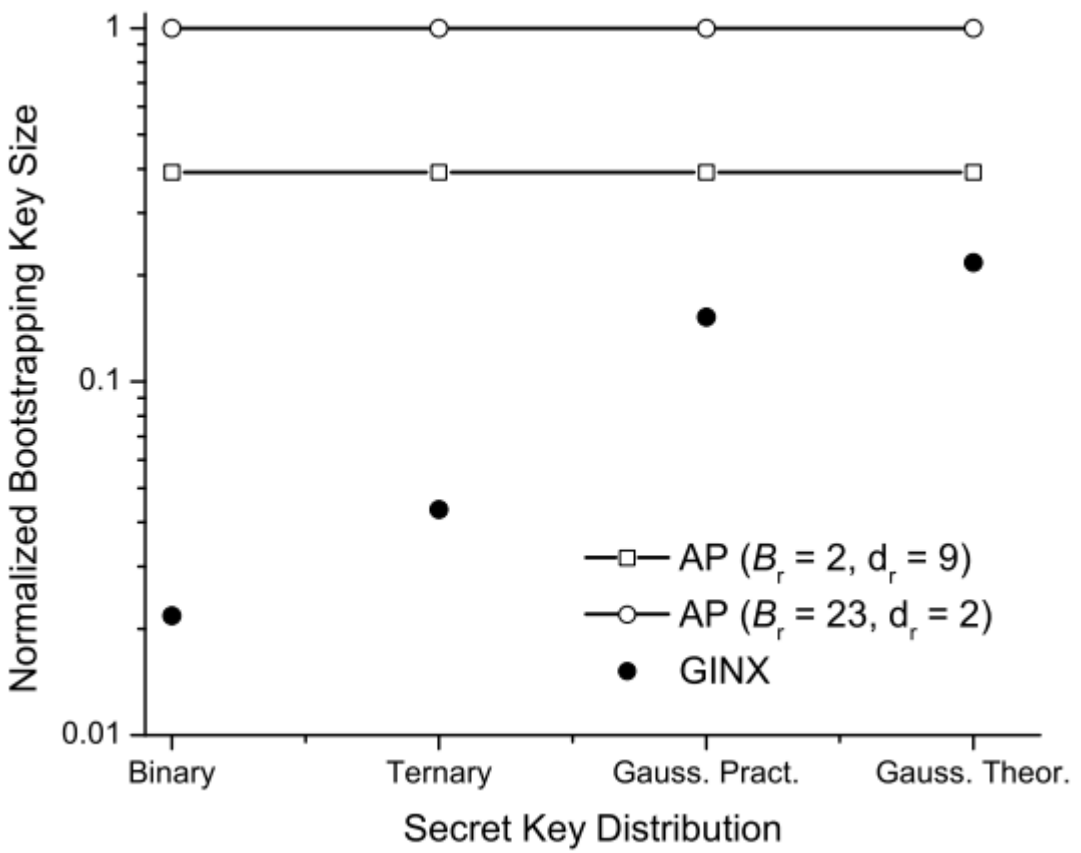
We recently presented a framework that integrates the Ducas-Micciancio (FHEW) and Chillotti-Gama-Georgieva-Izabachene (TFHE) schemes [MP20]. The results can be summarized as follows:

- When instantiated using LWE/Ring LWE, both schemes are the same except for the bootstrapping procedure: Alperin-Sherif—Peikert (AP) for FHEW vs Gamas-Izabachene-Nguyen-Xie (GINX) for TFHE.
- All other optimizations and enhancements, such as circuit bootstrapping, proposed in later TFHE papers equally apply to FHEW.
- Though the original TFHE scheme is formulated for binary secret distributions, it can be extended to ternary and Gaussian secrets.

Yuriy Polyakov ypolyakov@dualitytech.com

FHEW VS TFHE (CONT'D)





Yuriy Polyakov ypolyakov@dualitytech.com

MULTI-PARTY EXTENSIONS OF FHE

In [PRSV17], we showed that PRE can be built using the key switching procedure in BGV.

- We subsequently improved the security (changed from CPA to security under Honest Re-encryption Attacks [Coh19]) and extended it to BFV and CKKS.
- The current runtimes are one order of magnitude better than in [PRSV17] and look favorably compared to non-lattice constructions.
- There is an ongoing work with Aloni Cohen in this area.

Single-key FHE has also been extended in PALISADE to the multi-party threshold FHE setting using the ideas from [AJLTVW12, LTV11].

- Threshold FHE support has been added to BGV, BFV, and CKKS.
- The algorithms are being improved, and the enhanced implementation will soon be publicly available.

Yuriy Polyakov ypolyakov@dualitytech.com

LATTICE GADGET TOOLKIT

PALISADE provides an implementation of a recent lattice gadget toolkit developed in [GM18] and [GMP19]. The toolkit includes new gadget matrices and sampling algorithms:

- Efficient MP12 trapdoor sampling for lattices with arbitrary moduli, particularly, power-of-2 cyclotomic rings.
- Efficient subgaussian sampling for the same lattices.
- CRT (RNS) gadgets.
- RNS algorithms for discrete gaussian and subgaussian sampling.

The focus has been on general-purpose lattice trapdoor sampling algorithms that can support complex protocols, such as key-policy ABE and program obfuscation.

Yuriy Polyakov ypolyakov@dualitytech.com

PROTOCOLS BASED ON LATTICE TRAPDOORS

The following protocols are implemented in PALISADE (please see the references for implementation details):

- GPV digital signature and IBE, ciphertext-policy ABE [GPRRS18, GPRRSS19].
- Key-policy ABE in RNS [DDPRSSS18,GMP19].
- Conjunction obfuscation based on GGH15 in RNS [CDGKPRRS18].
- Token-based obfuscation of conjunctions and branching programs based on GGH15 in RNS [CGMPR18].

Yuriy Polyakov ypolyakov@dualitytech.com

MORE INFORMATION ON PALISADE

- PALISADE website
  - https://palisade-crypto.org/
- Stable release
  - https://gitlab.com/palisade/palisade-release
  - Current version: 1.9.2
- Development repo
  - https://gitlab.com/palisade/palisade-development
  - Includes latest code, preview releases, and additional lattice schemes
- Contact
  - contact@palisade-crypto.org

Yuriy Polyakov ypolyakov@dualitytech.com

REFERENCES

- [AJLTVW12] Asharov G., Jain A., López-Alt A., Tromer E., Vaikuntanathan V., Wichs D. (2012) Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE. In: Pointcheval D., Johansson T. (eds) Advances in Cryptology – EUROCRYPT 2012. EUROCRYPT 2012. Lecture Notes in Computer Science, vol 7237. Springer, Berlin, Heidelberg.
- [BEHZ16] Jean-Claude Bajard, Julien Eynard, M. Anwar Hasan, and Vincent Zucca, A Full RNS Variant of FV Like Somewhat Homomorphic Encryption Schemes, SAC'16.
- [BEMSZ19] Jean-Claude Bajard, Julien Eynard, Paulo Martins, Leonel Sousa, Vincent Zucca: Note on the noise growth of the RNS variants of the BFV scheme. IACR Cryptology ePrint Archive 2019: 1266 (2019)
- [BGG+14] Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology – EUROCRYPT 2014. Lecture Notes in Computer Science, vol. 8441, pp. 533–556. Springer (2014), http://dx.doi.org/10.1007/978-3-642-55220-5
- [BGPRV19] M. Blatt, A. Gusev, Y. Polyakov, K. Rohloff, V. Vaikuntanathan, Optimized Homomorphic Encryption Solution for Secure Genome-Wide Association Studies, Cryptology ePrint Archive, Report 2019/223, https://eprint.iacr.org/2019/223.
- [BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT), 6(3):13, 2014.
- [BPAVR19] A. Al Badawi, Y. Polyakov, K.M.M. Aung, B. Veeravalli, and K. Rohloff. Implementation and Performance Evaluation of RNS Variants of the BFV Homomorphic Encryption Scheme. IEEE Transactions on Emerging Topics in Computing (2020). https://eprint.iacr.org/2018/589.

Yuriy Polyakov ypolyakov@dualitytech.com

REFERENCES (CONT'D)

- [Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In Advances in Cryptology–CRYPTO 2012, pages 868–886. Springer, 2012.
- [BV11] Brakerski, Z., Vaikuntanathan, V.: Efficient Fully Homomorphic Encryption from (Standard) LWE. In: FOCS '11. pp. 97-106 (2011).
- [BVWW16] Z. Brakerski, V. Vaikuntanathan, H. Wee, and D. Wichs, "Obfuscating conjunctions under entropic ring lwe," in Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, ser. ITCS '16, 2016, pp. 147–156.
- [CC17] Ran Canetti and Yilei Chen. 2017. Constraint-hiding Constrained PRFs for NC1 from LWE. Cryptology ePrint Archive, Report 2017/143. https://eprint.iacr.org/2017/143.
- [CDGKPRRS18] Cousins, D. B., Di Crescenzo, G., Gür, K. D., King, K., Polyakov, Y., Rohloff, R., Ryan, G. W., and Savaş, E., "Implementing Conjunction Obfuscation under Entropic Ring LWE", 2018 IEEE Symposium on Security and Privacy (SP), pp. 354-371 [https://eprint.iacr.org/2017/844].
- [CGGI16] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Asiacrypt 2016 (Best Paper), pages 3-33.
- [CGGI17] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster Packed Homomorphic Operations and Efficient Circuit Bootstrapping for TFHE. ASIACRYPT (1) 2017: 377-408.
- [CGMPR18] Cheng, C., Genise, N., Micciancio, D., Polyakov, Y., and Rohloff, K., "Implementing Token-Based Obfuscation under (Ring) LWE", [https://eprint.iacr.org/2018/1222].

Yuriy Polyakov ypolyakov@dualitytech.com

REFERENCES (CONT'D)

- [CHKKS18b] J. H. Cheon, K. Han, A. Kim, M. Kim, Y. Song, A Full RNS Variant of the Approximate Homomorphic Encryption. In SAC 2018. Pages 347–368.
- [CKKS17] J. H. Cheon, A. Kim, M. Kim, Y. Song, Homomorphic Encryption for Arithmetic of Approximate Numbers. In ASIACRYPT 2017. Pages 409–437.
- [Coh19] Aloni Cohen: What About Bob? The Inadequacy of CPA Security for Proxy Reencryption. Public Key Cryptography (2) 2019: 287-316
- [CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. 2018. GGH15 Beyond Permutation Branching Programs: Proofs, Attacks, and Candidates. In CRYPTO 2018, Hovav Shacham and Alexandra Boldyreva (Eds.). 577–607.
- [DDPRSSS18] Dai, W., Doröz, Y., Polyakov, Y., Rohloff, K., Sajjadpour, H., Savaş, E., and Sunar, B., "Implementation and Evaluation of a Lattice-Based Key-Policy ABE Scheme", IEEE Transactions on Information Forensics and Security (IEEE TIFS), 2018, Vol. 13, No. 5, pp. 1169-1184 [http://eprint.iacr.org/2017/601].
- [DM15] L. Ducas and D. Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. EUROCRYPT 2015.
- [FV12] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, 2012:144, 2012.
- [GM18] Nicholas Genise, Daniele Micciancio: Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus. EUROCRYPT (1) 2018: 174-203
- [GMP19] Nicholas Genise, Daniele Micciancio, Yuriy Polyakov: Building an Efficient Lattice Gadget Toolkit: Subgaussian Sampling and More. EUROCRYPT (2) 2019: 655-684
- [GPRRS18] Gür, K. D., Polyakov, Y., Rohloff, K., Ryan, G. W., and Savaş, E., "Implementation and Evaluation of Improved Gaussian Sampling for Lattice Trapdoors", WAHC'18 Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, pp. 61-71 [http://eprint.iacr.org/2017/285].

Yuriy Polyakov ypolyakov@dualitytech.com

REFERENCES (CONT'D)

- [GPRRSS19] Gür, K. D., Polyakov, Y., Rohloff, K., Ryan, G. W., Sajjadpour, H., and Savaş, E., "Practical Applications of Improved Gaussian Sampling for Trapdoor Lattices", IEEE Transactions on Computers (IEEE TC), 2019, Vol. 68, No. 4, pp. 570 – 584 [https://eprint.iacr.org/2017/1254].
- [GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In STOC, pages 197–206, 2008.
- [HK20] K. Han, D. Ki, Better Bootstrapping for Approximate Homomorphic Encryption. In CT-RSA 2020. Pages 364-390.
- [LTV11] L´opez-Alt, A., Tromer, E., Vaikuntanathan, V.: Cloud-assisted multiparty computation from fully homomorphic encryption. Cryptology ePrint Archive, Report 2011/663 (2011). https://eprint.iacr.org/2011/663.
- [MP20] D. Micciancio and Y. Polyakov. Bootstrapping in FHEW-like Cryptosystems. Cryptology ePrint Archive. Report 2020/086, 2020. http://eprint.iacr.org/2020/086.
- [PRSV17] Polyakov, Y., Rohloff, K., Sahu, G., Vaikuntanathan, V.: Fast proxy reencryption for publish/subscribe systems. ACM Trans. Priv. Secur. 20(4), 14:1–14:31 (Sep 2017)
- [SS11] Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, Advances in Cryptology – EUROCRYPT 2011, volume 6632 of Lecture Notes in Computer Science, pages 27–47. Springer Berlin Heidelberg, 2011.
- [ZZG12] Jiang Zhang, Zhenfeng Zhang, and Aijun Ge. Ciphertext policy attribute-based encryption from lattices. In Heung Youl Youm and YoojaeWon, editors, 7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS '12, Seoul, Korea, May 2-4, 2012, pages 16–17. ACM, 2012.

Yuriy Polyakov ypolyakov@dualitytech.com

THANK YOU

Yuriy Polyakov

ypolyakov@dualitytech.com