NORSK REGNESENTRAL
NORWEGIAN COMPUTING CENTER

IARIA

Secure Data Processing in AI Applications Through Federated Learning and Homomorphic Encryption

SECURWARE 2024 - The Eighteenth International Conference on Emerging Security Information, Systems and Technologies

https://www.iaria.org/conferences2024/SECURWARE24.html

Svetlana Boudko svetlana@nr.no

Nice, France, Nov 03, 2024

whoami

- Dr. Svetlana Boudko is a Senior Research Scientist at the Norwegian Computing Center in Oslo, Norway.
- She defended her PhD in computer science at the University of Oslo in 2014.
- She has over 20 years of experience working on R&D projects.
- Her areas of interest include cybersecurity, privacy and data protection, secure multi-party computation, and federated learning.

Roadmap

- Background
- Federated Learning
  - Federated Aggregation
  - FL tools
  - Privacy & Security concerns
- Homomorphic Encryption
  - Different types, methods, and schemes
  - Open-source libraries
  - Standardization efforts
- When these technologies are used together
- Conclusion & Future

Various industries benefit from AI applications


Healthcare


Finance


Retail


Telecom

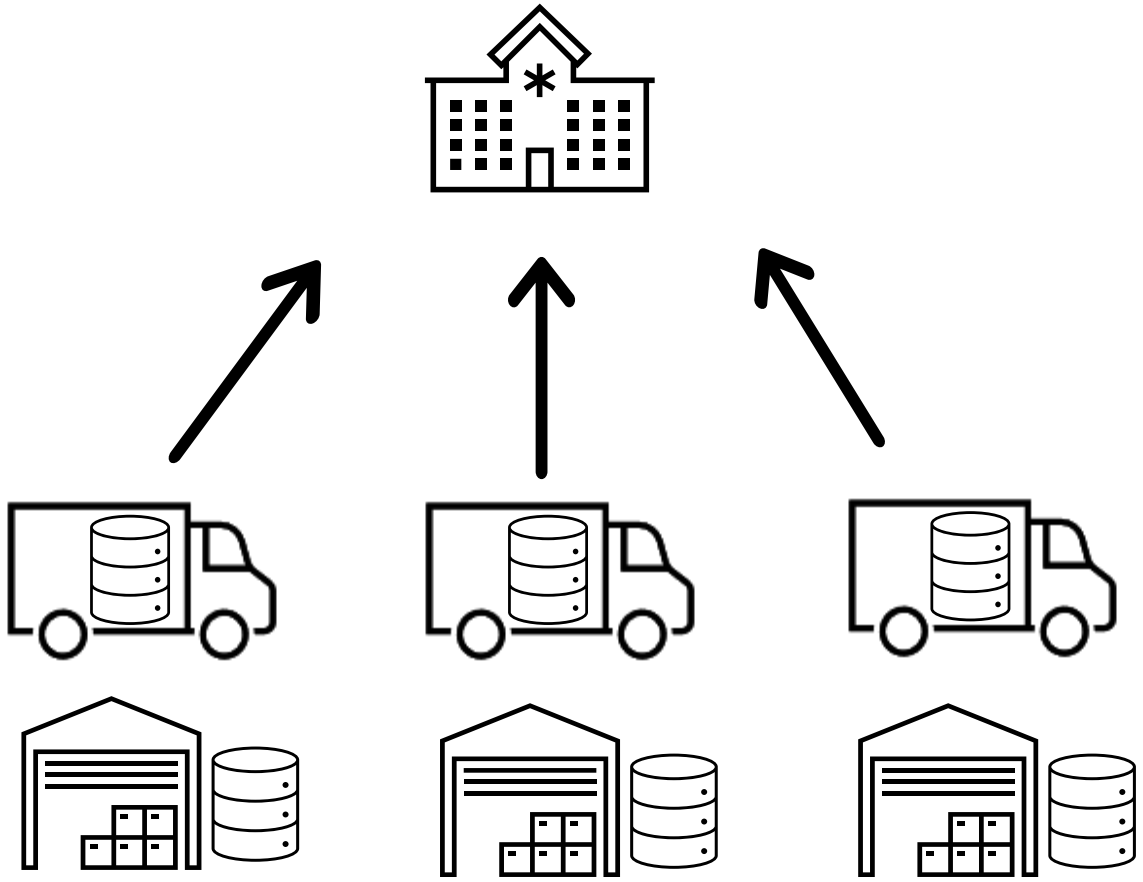

Manufacturing


Transportation and Logistics


Education


Agriculture

Challenges related to data sharing compromise centralised approach

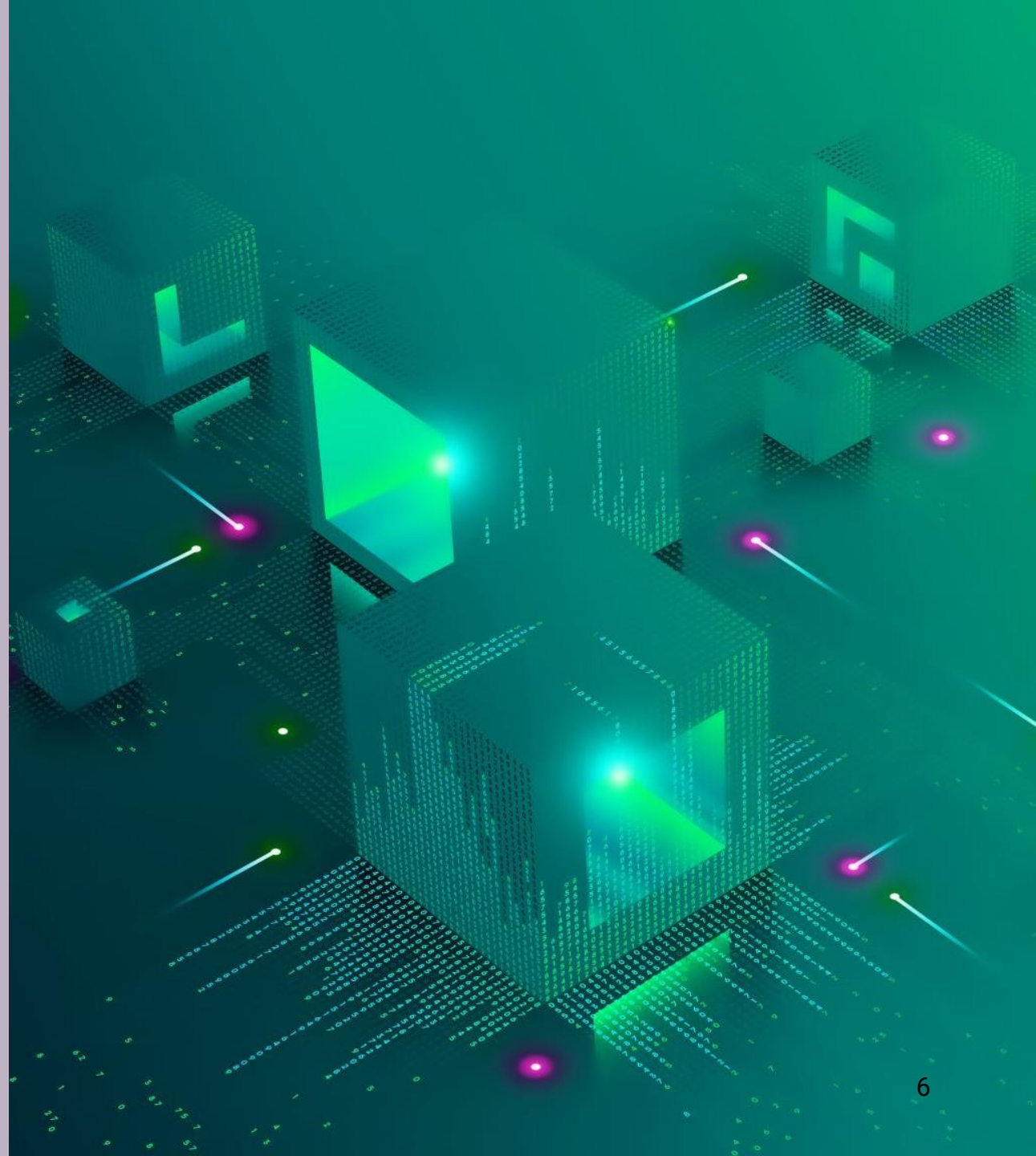


all data in single location

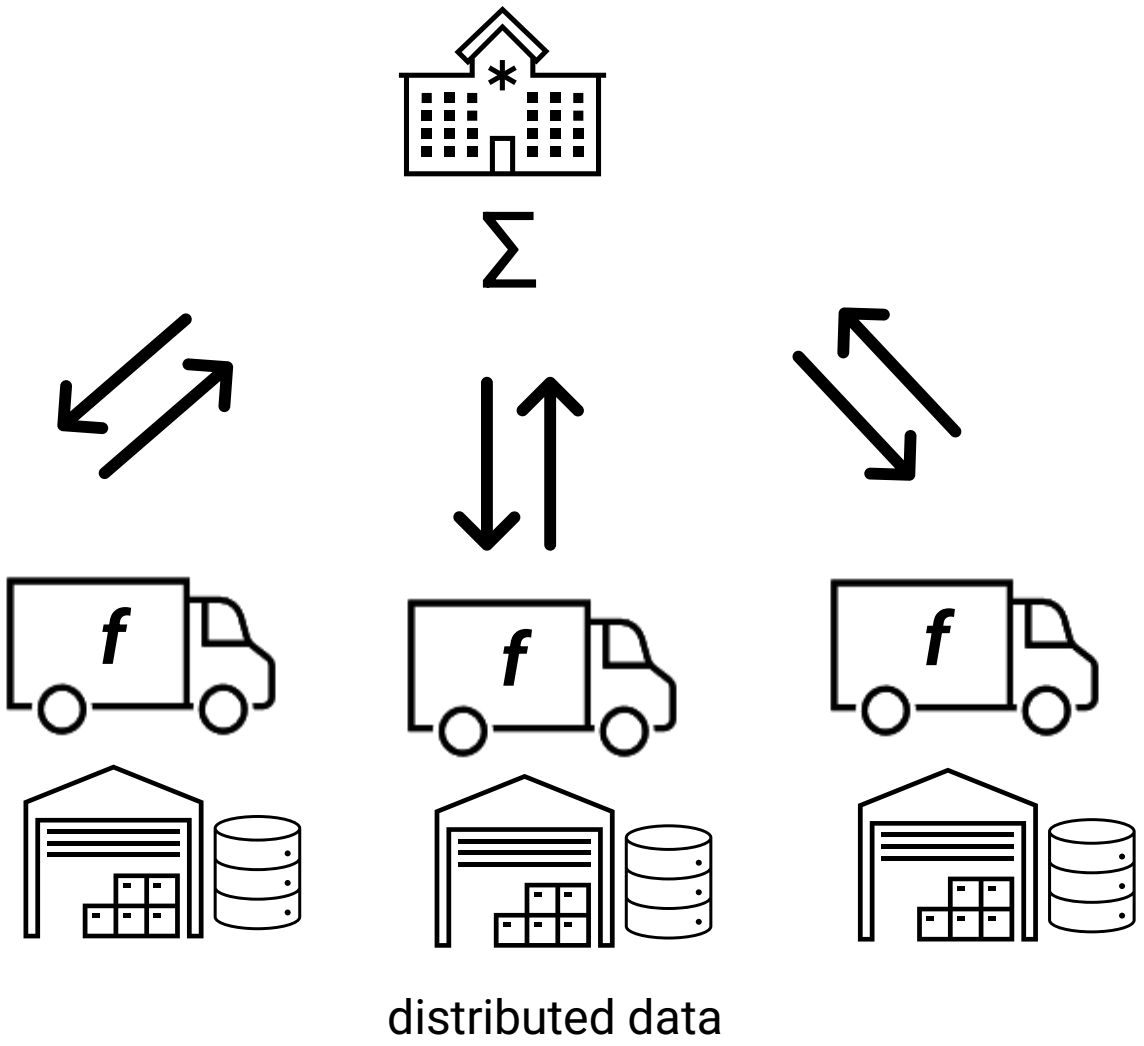
- \+ full control over data quality
- \+ data concistency
- − difficult to update data
- − compromised privacy
- − security risks / one point of failure
- − logistic challenges
- − storage

Federated approach

- Google introduced Federated Learning in 2016 aiming on
  - reduction of data transfer costs
  - protection of privacy-sensitive information
- enables on-device model training using client specific data
- further aggregation of the obtained local model updates on a central server
- cross-silo (organizations / jurisdictions) and cross-device
- data partitioning: horizontal FL (different sample spaces), vertical FL (different feature spaces), and hybrid FL
- IEEE Guide for Architectural Framework and Application of Federated Machine Learning (IEEE 3652.1-2020)



Federated learning enable data analytics across different silos



- ⊕ privacy issues are better addressed
- ⊕ logistic
- ⊕ data updates
- ⊕ cross-border cooperation: training of models with data coming from different jurisdictions

- ⊖ non-iid (independent and identically distributed) issues should be addressed, increasing complexity in model training and aggregation [Ma22, Ka21]
- ⊖ risk of introducing biases; methods for ensuring fairness and addressing sources of bias

Federated aggregation is a key process in federated learning

- Federated Stochastic Gradient Descent (FedSGD)
  - local gradients
- Federated Averaging (FedAvg)
  - most common/communication efficiency
  - differs from FedSGD in what/how information is aggregated from the local model
  - local stochastic gradient updates from several epochs of training
  - averaged weights for updates at the aggregator
  - assumption: all devices have an equal contribution to the global model

Federated aggregation methods, cont.

- Weighted Federated Averaging
  - modification of FedAvg
  - takes into account the number of data samples on each device
  - devices with more data samples have a higher contribution to the global model
- Hierarchical aggregation (FogL, MACFL, MaxQ, …)
  - across different devices
  - aggregates local models from devices closely located together before further aggregation on the server
  - reduces communication overhead and the number of model transfer rounds

FL tools

- TensorFlow Federated, Google
- Flower, Flower
- PySyft, OpenMind community
- FedML
- OpenFL
- IBM Federated Learning
- Microsoft's FL
- Vantage6

Vantage6

- supported by the Netherlands Comprehensive Cancer Organisation (IKNL)
- designed to facilitate collaborative data science / analysis / machine learning, cross-silo
- designed to be flexible and modular, allows users to create custom algorithms for specific tasks
- supports different programming languages such as Python and R, robust support for handling missing or inconsistent data
- great community support

from https://distributedlearning.ai/

Security & Privacy concerns

- Inference attacks
  - model inversion
  - membership inference
  - attribute inference
- Data/gradient leakage attacks
- Model/data poisoning
- Model extraction
- GAN-based attacks where an attacker
  - trains a GAN model to learn the distribution of the victim’s private dataset based on the shared model gradients
  - has a high chance to fully reconstruct users’ private data [Hi17]

conference dinner

Federated Learning can be further integrated with privacy-preserving technologies

- Anonymization
  - involves removing any identifying information from the data, such as names, addresses, or social security numbers
  - helps to protect individuals' privacy, but it's crucial to ensure that the data cannot be re-identified
  - risk of re-identification (K-anonimity, L-diversity, T-closeness)
- Differential Privacy
  - adds noise to the data to protect individual privacy while still allowing for overall trend analysis
  - provides a mathematical guarantee of privacy by ensuring that the addition or removal of a single database entry does not significantly change the output of a data analysis

Privacy-preserving technologies, cont.

- Trusted Execution Environments
  - are secure areas of a main processor
  - ensure that the data being processed is secure, even in the presence of a compromised operating system
  - are used in a wide range of applications, including secure payment applications and authentication
- Data masking
  - sensitive data is replaced with fictional yet realistic data
  - allows developers and testers to work with data close to the actual data without violating privacy rules

Secure Multi-Party Computation

- Multiple parties jointly compute a function over their inputs while keeping those inputs private
- Homomorphic encryption
  - allows computations to be performed on encrypted data without decrypting it; the decrypted result matches the result of operations performed on the plaintext
  - data privacy and trust: protection of data, ML models, model updates
  - preservation of data ownership
  - versatility : can be applied to many types of data and computations
  - computationally expensive

Homomorphic Encryption

- The idea of direct computation on encrypted data was first recognized in 1978 by Ronald L. Rivest, Len Adelman, and Michael L. Dertouzos
  - RSA (Rivest-Shamir-Adleman) encryption
- Partially Homomorphic Encryption (PHE): simplest form of homomorphic encryption
  - perform one type of operation either addition or multiplication an unlimited number of times. The most famous PHE scheme is the RSA algorithm, which is a multiplicative homomorphic encryption
- Somewhat Homomorphic Encryption (SHE)
  - perform both addition and multiplication operations, but only up to a certain level; the limitation: the noise introduced into the cipher text after each operation; grows exponentially with the number of operations

Fully Homomorphic Encryption

- Unlimited number of both addition and multiplication operations
- First practical scheme was proposed by Craig Gentry in 2009 [Ge09]
  - defined as a circuit of logic gates, and unrestricted computation occurred on encrypted data with results encrypted in the same way
  - extremely slow, taking about 30 minutes to complete a single logic gate on standard x86 hardware
- Continued research has resulted in four distinct generations of FHE and substantial speed-up on standard hardware platforms

Homomorphic Encryption can be further classified as:

- Single key
  - single key schemes, all clients use the same public-private key pair and can decrypt updates from other participants => no different from directly sending plaintext
- Multikey homomorphic encryption
  - supports a variable number of clients, the number of clients in one operation is bound
  - computationally inefficient, key generation is expensive, ciphertext expands proportionally to the number of clients, runtime - quadratically
  - new client can joint
- Threshold homomorphic encryption
  - relaxes the problem with ciphertext expansion, runtime and key generation
  - clients are predefined to generate a common public key and evaluation keys
  - new client cannot joint, requiring new key generation

Threshold FHE schemes can be separated into three categories

Brakerski-Gentry-Vaikuntanathan [BGV14] (BGV) and Brakerski [Bra12]/Fan-Vercauteren [FV12] (BFV)	• Support SIMD encrypted computations for arithmetic circuits modulo a prime power; integer
Ducas-Micciancio [DM15] (DM, also called FHEW)/Chillotti-Gama-Georgieva-Izabachene [CGGI16] (CGGI, also called TFHE)	• Support binary or small-precision arithmetic ; arbitrary functions are evaluated using lookup tables via functional/programmable bootstrapping; integer
Cheon-Kim-Kim-Song [CKKS17] (CKKS, also called HEAAN)	• Support SIMD fixed-point-like arithmetic circuits (for many real-number applications) , includes approximate bootstrapping

based on LWE/RLWE/MLWE , lattice-based cryptography, known to be quantum resistant

Open-Source Fully Maintained Libraries

- OpenFHE (former PALISADE)
  - Duality Technologies
  - C++ and Python bindings (Linux)
  - Threshold FHE for BGV, BFV, and CKKS schemes
  - Strong community support
- Lattigo: lattice-based multiparty homomorphic encryption library
  - The EPFL Laboratory for Data Security (2019-2022), Tune Insight SA (2022-)
  - Go
  - Threshold FHE for BFV, BGV, and CKKS schemes
- TFHE-rs
  - Rust
  - Zama.ai, community support
  - working on threshold FHE solutions, but not in the library

“Homomorphic encryption is already ripe for mainstream use, but the current lack of standardization is making it difficult to start using it.” [HES]

ISO standardization

- Targeting BGV, BFV, CKKS, and CGGI
- Technical Committee : ISO/IEC JTC 1/SC 27 Information security, cybersecurity and privacy protection

Homomorphic Encryption Standardization / HomomorphicEncryption.org

- open consortium of industry, government and academia to standardize homomorphic encryption
- founded in 2017
- 6 meetings held since 2017, next meeting: Sunday October 13, 2024
- security recommendations are available since 2018

NIST Workshops on Multi-Party Threshold Schemes

Threshold FHE representative scenario in the context of FL

- Each participant has some secret data
- Key generation routine is done for all participants
  - participants generate their secret key shares and jointly generate evaluation and public keys
  - evaluation key is sent to the central server
- An initial FL model is trained and sent to all participants
- Each participant evaluates the function, computes and encrypts model updates, sends the result to the central server
- The central server aggregates received updates
- Participants jointly decrypt the result
- A participant joins or leaves the group: new key generation routine

HE in Federated Learning

- supports only additions and multiplications
- polynomial functions can be implemented in a straightforward way
- averages, weighted-sum functions can be directly implemented, inverse function / Taylor series
- some schemes do not support floating-point numbers: convert real numbers to integers by proper scaling
- sigmoid and rectified linear activation (ReLU) functions are non-polynomial functions: approximate these functions with low-degree polynomials or replace them with polynomial activation functions (for models)

HE in Federated learning, contd

- communication overhead
  - higher with HE, generation of keys, joint decryption
- scalability is challenging for cross-device FL
  - involve up to millions of devices
  - more challenging using HE
  - cross-silo vs cross-device – different requirements
- model quality
  - non-iid issues
  - biases
- integration issues

Conclusion

- Federated Learning presents a promising approach to machine learning and data analysis
  - reduces the amount of data that needs to be transmitted and strengthens protection of sensitive data
  - robust AI models without the necessity of transferring raw data across the network
- Homomorphic encryption enhances data privacy and security in machine learning and AI
  - allows for private computation and protects data even if the computing environment is compromised
  - threshold HE: lattice-based schemes considered to be post-quantum resistant

There are still challenges and future steps to consider

- Efficiency and Performance: Federated Learning and Homomorphic Encryption require significant computational resources and can be slower than traditional methods. Future research should focus on improving efficiency and performance, making these technologies more practical for widespread use.
- Scalability: As the number of devices and the amount of data grows, it will be a challenge to scale these technologies. Research is needed to develop methods for scalable, decentralized learning and efficient homomorphic encryption.

There are still challenges and future steps to consider, cont.

- Standards and Regulations: As new technologies, standards and regulations for Federated Learning and Homomorphic Encryption are still being developed. It will be important to establish guidelines for their use to ensure privacy and security.
- Integration with Existing Systems: These technologies will need to be integrated with existing systems. This will require new tools and methodologies, as well as education and training for users.

Thank you!

References

[ABGS22] Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, and Tjerand Silde. Verifiable mix-nets and distributed decryption for voting from latticebased assumptions. IACR Cryptol. ePrint Arch., page 422, 2022.

[BGV14] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT), 6(3):1–36, 2014.

[Bra12] Z. Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In CRYPTO 2012. Pages 868 – 886.

[CCMS21] Kelong Cong, Daniele Cozzo, Varun Maram, and Nigel P. Smart. Gladius: LWR based efficient hybrid public key encryption with distributed decryption. In ASIACRYPT (4), volume 13093 of LNCS, pages 125–155. Springer, 2021.

[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.

[CKKS17] J. H. Cheon, A. Kim, M. Kim, Y. Song, Homomorphic Encryption for Arithmetic of Approximate Numbers. In ASIACRYPT 2017. Pages 409–437.

[DM15]: L. Ducas and D. Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. EUROCRYPT 2015.

References

[FV12] J. Fan and F. Vercauteren. Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive. Report 2012/144, 2012. http://eprint.iacr.org/2012/144.

[KS23] Kamil Kluczniak, Giacomo Santato: On Circuit Private, Multikey and Threshold Approximate Homomorphic Encryption. IACR Cryptol. ePrint Arch. 2023: 301 (2023)

[Ma22] Ma, X., Zhu, J., Lin, Z., Chen, S., Qin, Y.: A state-of-the-art survey on solving non-iid data in federated learning. Future Generation Computer Systems 135, 244258 (2022)

[Ge09] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, ser. STOC ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 169–178

[Mo20] Arturo Moncada-Torres, Frank Martin, Melle Sieswerda, Johan van Soest, Gijs Geleijnse. "VANTAGE6: an open source priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange". AMIA Annual Symposium Proceedings, 2020, p. 870-877

[Sm22] Djura Smits, Bart van Beusekom, Frank Martin, Lourens Veen, Gijs Geleijnse, Arturo. Moncada-Torres, “An Improved Infrastructure for Privacy-Preserving Analysis of Patient Data”, Proceedings of the International Conference of Informatics, Management, and Technology in Healthcare (ICIMTH), vol. 25, 2022, p. 144-147

References

[Ka21] P. Kairouz et al, Advances and Open Problems in Federated Learning, Foundations and Trends in Machine Learning Vol 4 Issue 1, 2021

[Hi17] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan:

information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 603–618, 2017

[Ri78] Rivest, R. L., Adleman, L., Dertouzos, M. L. 1978. On data banks and privacy homomorphism. Massachusetts Institute of Technology. Academic Press; https://people.csail.mit.edu/rivest/RivestAdlemanDertouzos-OnDataBanksAndPrivacyHomomorphisms.pdf.

[FV12] J. Fan and F. Vercauteren. Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive. Report 2012/144, 2012. http://eprint.iacr.org/2012/144.

[KS23] Kamil Kluczniak, Giacomo Santato: On Circuit Private, Multikey and Threshold Approximate Homomorphic Encryption. IACR Cryptol. ePrint Arch. 2023: 301 (2023)