Oratio Homomorphico: Creating a Lecture for Homomorphic Encryption by applying Design-Based Research

Thomas Prantl[1][0000−0003−4044−8494], Tobias Schneider[1][0009−0008−3023−5755], Lukas Horn[1][0009−0004−4959−1371], Simon Engel[1][0009−0005−3354−4746], Lukas Iffländer[2][0000−0002−8506−2758], Christian Krupitzer[3][0000−0002−7275−0738], Rafael Bonilla[4][0000−0003−1205−3583], and Samuel Kounev[1][0000−0001−9742−2063]

[1] University of Wuerzburg, Germany
{firstname.lastname}@uni-wuerzburg.de
[2] HTW Dresden - University of Applied Sciences, Germany
lukas.ifflaender@htw-dresden.de
[3] University of Hohenheim, Germany
christian.krupitzer@uni-hohenheim.de
[4] Escuela Superior Politecnica del Litoral, ESPOL
rabonilla@espol.edu.ec

Abstract. The ever-increasing pace of digitalization, with its collection of data and the training of artificial intelligence on this data, is bringing the topic of privacy-preserving computing more and more into focus. One promising technology for implementing the privacy-preserving computing paradigm is homomorphic encryption, which has been driven forward in recent years and still holds great potential for improvement in the future. The training of future computer scientists in this field is therefore of crucial importance in order to put this key technology into practice in the future. However, to the best of the authors' knowledge, there is currently no course that provides students with insights into the theoretical foundations of homomorphic encryption and teaches the practical applicability and extensibility of existing homomorphic cryptosystems. To close this gap, we present our own lecture in this paper, which focuses on the combination of theory and application of homomorphic encryption. We created our lecture material according to the Design-Based Research principle. We held and evaluated our lecture at three German universities. Our lecture evaluations have shown that we have been able to create suitable lecture material on homomorphic encryption and its application, which we would like to share with other lecturers.

Keywords: Homomorphic Encryption · Lecture · Privacy · Newton-Raphson · CKKS

1 Introduction

On the 23rd of April 2025, WhatsApp announced that they are adding an "Advanced Chat Privacy" mode to their product [25]. Aside from preventing chat

exportation, this mode also disallows using the messages to train Artificial Intelligence (AI) models. The latter is something that became relevant in recent years with the enormous growth of AI. Where before data was mainly used to create detailed profiles of users and use them, for example, to tailor ads, this data is now used to build and improve AI models. Therefore, data privacy becomes even more relevant. Although a lot of people say "what does XY care about my data?" – where XY can be replaced with an arbitrary tech company – most of them still do not want these companies to know everything. For example, information about the financial situation or health is a topic that is often not even communicated to the family. WhatsApp also mentions health challenges as an example of message contents that are not determined for everyone. However, more data, especially in the medicine context, would very likely help improve therapies, medications, etc. Therefore, there is a conflict between using data and respecting the individuals' privacy. One solution to this is Homomorphic Encryption (HE). This concept allows performing operations on encrypted data. In classical cryptography, the data can not be used in a meaningful way when it is in an encrypted state.

This changes with HE, which is why it is considered the holy grail of cryptography. A typical practical use would be that there is a data owner who wants to process the data, but does not have enough computational power. Therefore, the process is outsourced to a cloud provider. In the classical way, the provider also knows the data and may make use of it. However, when the data gets encrypted using HE technologies first, this is not the case. The calculation can then be done encrypted, and the result is sent back to the owner, who is the only one able to decrypt it. Then, both key points are achieved, the privacy of the data is ensured, and the results that can be obtained with this data are also there. This topic was first proposed in 1978 by Ronald Rivest et al. [22]. However, it was unknown for a long time whether there were cryptosystems that had this property. In 2009, there was a breakthrough as Craig Gentry found such a cryptosystem [12]. Thus, this topic is rather new. However, when we look at the statistics in Figure 1, we see that the number of publications in this field is growing rapidly. This shows that this topic will be highly relevant in the future.

It is of key importance not only to spread this topic but also to make university students aware of it and teach the foundations. This can, for example, be done by offering a lecture for master's students. A special focus should be on the practical application of the technologies and the awareness of possible pitfalls. We aimed to build and share a self-contained course, which requires only marginal prior knowledge, with well-known educational concepts like Design-Based Research. Our lecture materials are available in our public GitHub project[5]. In this paper, we want to present the details and give insights into the design process. The remainder of this document is structured as follows. In Section 2, we introduce basic concepts of homomorphic encryption. This is followed by an overview of related work, including differentiation in Section 3. In Section 4, we present our approach for the creation and evaluation of the lecture. An overview of the

[5] `https://github.com/DescartesResearch/he_lecture`

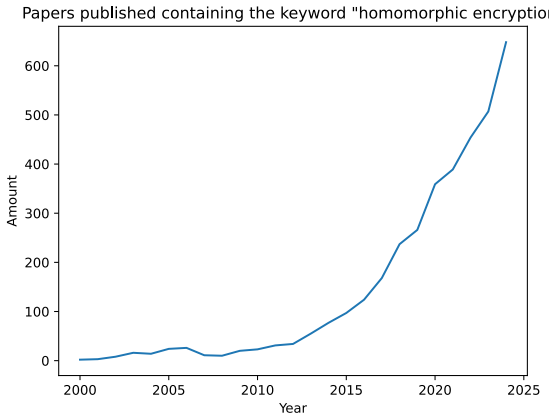

Fig. 1. Amount of publications containing the keyword "homomorphic encryption" per year from 2000 to 2024 [8]

content of the lecture and the exercise concept is provided in Sections 5 and 6. Section 7 rounds off the paper with a summary.

2 Background

Since there are already various definitions with different focuses, we want to provide a formal definition of HE that serves as the foundation of the lecture. The idea is derived from [13, 14] and similar to [19, 20].

Definition 1. *We define the following sets and words:*

- $\mathcal{P}$ *is a finite, non-empty set. It is called "plaintext space" and elements of it are named "plaintext". All data that will be encrypted is drawn from this set.*
- $\mathcal{K}$ *is a finite, non-empty set. It is called "key space" and elements of it are named "key". They are used as secrets for encryption and decryption.*
- $\mathcal{C}$ *is a finite, non-empty set. It is called "ciphertext space" and elements of it are named "ciphertext". If* $p \in \mathcal{P}$ *gets encrypted, its result is an element of* $\mathcal{C}$*.*

Definition 2 (Cryptosystem). *A cryptosystem is defined as a tuple* $(\Sigma, \mathcal{G}, \mathcal{E}, \mathcal{D})$ *with the following properties:*

- $\Sigma = \mathcal{P} \cup \mathcal{K} \cup \mathcal{C}$ *is the alphabet*
- $\mathcal{G}$ *is a probabilistic algorithm that returns a key pair* (pk, sk)
- $\mathcal{E} : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$ *is a non-deterministic function taking a key and a plaintext that outputs a ciphertext*

- $\mathcal{D} : \mathcal{C} \times \mathcal{K} \to \mathcal{P}$ *is a deterministic function taking a key and a ciphertext that outputs a plaintext*
- $\forall m \in \mathcal{P} : \forall (pk, sk) \in \mathcal{K} : \mathcal{D}(sk, \mathcal{E}(pk, m)) = m$

The last claim ensures that every ciphertext can be properly decrypted again. Otherwise, the encryption would not make a lot of sense as encrypted messages can no longer be restored.

Definition 3 (Homomorphic Cryptosystem). *A homomorphic cryptosystem is defined as a tuple* $(\Sigma, \mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{F})$ *with the following properties:*

- $(\Sigma, \mathcal{G}, \mathcal{E}, \mathcal{D})$ *is a cryptosystem according to Definition 2*
- $\mathcal{F}$ *is a set of functions that the cryptosystem can compute on the ciphertexts*

If $\mathcal{F}$ *contains all possible functions and can calculate them an unbounded number of times, the cryptosystem is called "fully homomorphic".*

The problem for many years was that there were no such fully homomorphic cryptosystems. Often, they were able to calculate all or a lot of functions, but only for a limited number of times. If this limit was exceeded, the ciphertexts could no longer be decrypted to the correct plaintexts. This happened due to the addition of noise in the encryption/calculation process to achieve the homomorphic property. Gentry's breakthrough [12] was a method called "bootstrapping" that allowed lowering the noise level without decryption and thus allowing for unlimited calculations in theory.

At the moment, there are several cryptosystems with different domains like integer [4, 3, 11], boolean/circuit [10, 7], and real numbers [6]. Floating point numbers are the most common domain and also allow the usage of the traditional algorithms, which is why we focused the lecture on this domain and the Cheon-Kim-Kim-Song (CKKS) cryptosystem.

3 Related Work

There are, of course, already a number of lectures dealing with the topic of homomorphic encryption. For example, the lecture "An Intensive Introduction to Cryptography" [2] was held at Harvard in 2021. The focus of the lecture is not directly on the topic of homomorphic cryptosystems but gives a good, broad overview of various cryptographic topics, from pseudorandomness to homomorphic encryption and quantum cryptography to anonymous communication. The lecture shows how homomorphic cryptosystems can be built from the Learning With Errors (LWE) problem and proves that they are correct and secure. In contrast to this lecture, our lecture focuses solely on the topic of homomorphic encryption and shows how the homomorphic cryptosystem CKKS works and how its range of functions can be extended by the following functionalities: conditional branching, binary step function, division function, square root function, logarithm function, exponential function, logarithm function, minimum and maximum function, as well as the equals and smaller than function. Our lecture

also provides the students with the opportunity to gain practical experience with what is currently probably the best open source library for homomorphic encryption by implementing many of the lecture contents themselves. The benefit of our course is that students are better prepared to use the CKKS in practical applications.

The lecture "Advanced Cryptography" [17], which was held at UC San Diego in 2023, focuses on homomorphic cryptography. The lecture introduces a number of different homomorphic cryptosystems, such as GHS, BFV, TFHE or CKKS, and explains how multiplication can be implemented homomorphically or how the bootstrapping algorithm works. We differ from this lecture in that with CKKS, we focus on a single homomorphic cryptosystem and emphasize how CKKS can be extended by the above-mentioned functionality in practice.

In 2024, the lecture "Data Privacy and Security" [24] was held at the Sapienza University of Rome. It covers a wide range of topics, including secure multiparty computation, differential privacy, and post-quantum cryptography. The lecture also covers the topic of homomorphic encryption in the context of post-quantum cryptography. Specifically, the GSW scheme is introduced, and how it can be extended by key switching. In contrast to this lecture, we focus on the homomorphic cryptosystem CKKS and show how it can be extended by the aforementioned functionalities in practice.

Another lecture is the lecture "Cryptography and Protocol Engineering" [15], which was held at the University of Cambridge in 2025. The lecture provides a broad overview of various topics such as secure implementation of cryptographic protocols and possible attack vectors on them, elliptic curve cryptography, and homomorphic encryption. With respect to homomorphic encryption, the lecture explains how homomorphic cryptosystems can be created from the LWE problem and how they can be used for private information retrieval applications. We differ from this lecture in that we only focus on the topic of homomorphic encryption and also use CKKS as a cryptosystem. Also new in our lecture is that the focus is on extending CKKS with the previously mentioned functionalities in practice.

Also in 2025, the lecture "Cryptography" [26] was held at the University of Texas. Similar to the other lectures, it provides a broad overview of a wide range of cryptographic topics. For example, zero-knowledge proofs, post-quantum cryptography, and collision-resistant hash functions are presented. Homomorphic encryption is also covered in the lecture. Specifically, the construction of a homomorphic cryptosystem is discussed, and its security and correctness are justified. Analogous to the other lectures, we differ again in that we focus only on the topic of homomorphic encryption and consider how we can extend the homomorphic cryptosystem CKKS with the above-mentioned functionalities in practice.

4 Approach

In this section, we want to explain the organizational setting of the lecture and the fundamental approach to its design.

4.1 Organizational Setting

The audience that the lecture targets is Master's students. Although it may also be possible to present it for undergraduates, we did not focus on this group - this could be interesting for future work. First, the mathematical foundations require an elaborate understanding of linear algebra and number theory. This exceeds the number of topics that are typically taught in introductory mathematics courses. Thus, undergraduate students in the early semesters may be overstrained. Secondly, the lecture shows the application of HE in the data science domain. Due to time constraints, the concepts and techniques used there can not be explained in this lecture, and thus, some prior knowledge is required. Typically, such knowledge is built throughout the undergraduate studies and can then be used in a Master's programme.

The general structure is one 90-minute lecture and one 90-minute weekly exercise session. The latter is not compulsory and does not introduce new concepts, but focuses on practicing everything learned in the previous week's lecture. This setting is typical for universities in Germany. The lecture itself then consists of eight separate units. However, not all of them can be held in a ninety-minute session. Therefore, some of them take two or even three sessions. They are put together as a unit as they are thematically connected. This is, for example, the case for the detailed explanation of the algorithms in the CKKS cryptosystem [6]. Although it is impossible to cover Encoding, Decoding, Encryption, and Decryption in one lecture, it does not make sense to put them in several units, as these four operations are strongly connected.

The lecture was held three times between winter 2024/2025 and spring 2025. Once as a regular lecture in the winter term of 2024/2025 at the University of Würzburg. Afterwards, it was offered as a block course in the form of a "winter school" at the University of Hohenheim. The third iteration was also a block course at the University of Frankfurt. These block courses lasted one week, and multiple units were held in one day. Typically, there were lectures in the morning, and in the afternoon, there were exercise sessions.

We selected these two forms for organizational reasons and to test out different settings as suggested by the design-based research principles covered in the next section. Besides being squeezed into a single week, the module's content was identical.

4.2 Design-Based Research

A concept that is frequently applied in higher education is the so-called "Design-Based Research". It originates from ideas by Brown and Collins [5, 9]. This term, however, is not used consistently in the literature. As McKenney and Reeves [16] mention, there are multiple different names with also slightly different definitions. We follow the definitions and ideas from McKenney and Reeves [16] and Scott et al. [23]. The concept consists of four phases that are iterated. There is no predefined endpoint; they end when the researchers are satisfied with the results. The process is also depicted in Figure 2, and explained below:

1. **Identify, Reflect, Adapt:**
   This phase is twofold. The initial phase consists of identifying the problem that needs to be addressed. During the iteration, this phase takes the results from the evaluation phase, reflects on the decision, and spins up new ideas and methods on how everything can be adapted.
2. **Design:**
   In this phase, participants design a solution to the problem identified in the previous phase. This can have various forms, such as a lecture, an exercise sheet, or even minor changes in just a single slide.
3. **Test:**
   This phase brings the materials from the phase before to the real world. Depending on the form, this can be a classroom, an online test, or other types of assessment.
4. **Evaluate:**
   In the final phase, the researchers evaluate the material on its effectiveness. That can be done, e.g., using tests, surveys, or interviews.

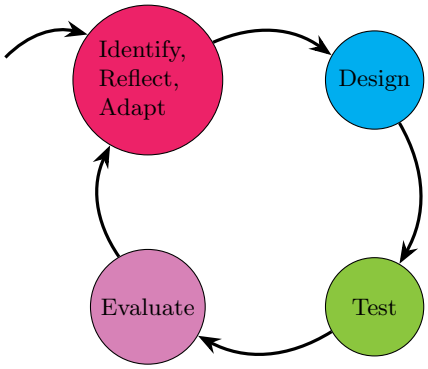

Fig. 2. The general principle of Design-Based Research, adapted from [23]

Applying this principle to the design of our lecture was done in multiple granularities. On a high level, we first identified the lack of a lecture that not only teaches the theory of HE but also focuses on applying it in practice. Out of this problem, we designed a whole lecture with corresponding exercise sheets, tested it with a real audience, and implemented their feedback for the next two tests at different universities as mentioned above. During the design phase, we stumbled upon more problems that required the implementation of this concept. For example, the existing tutorials that present the mathematical foundation of HE either require a lot of prior knowledge or are often incomplete. Therefore, we thought about how this can be circumvented and what level of detail is needed to understand, and how it can be presented. In order to do so, different methods were tested, for example, presenting toy examples at the blackboard or just relying on presentation slides.

The evaluations we conducted of our lecture material for each iteration show an overall positive picture. For example, on a Likert scale of 1 (=good) to 5 (=bad), we asked the participating students to rate the slide design in each iteration. The average rating of an iteration was between 2.3 and 1.4, and the average rating of the slide design across all iterations was 1.8. In our opinion, these numbers mean that the slide design was generally well done. In addition to the slide design, we also had the students rate the comprehensibility of the structure of the lecture in each iteration on a scale from 1 (= understandable structure) to 5 (= not understandable structure). The average rating of the comprehensibility of the individual iterations was between 2.2 and 1.3, averaged over all iterations at 1.8. In addition to the lecture structure, we also asked the students whether our lecture material was able to convey the concepts covered in an understandable way. For this purpose, students were able to give the ratings "yes" or "no". The average rating of the comprehensibility of the concepts varied between 87.4% and 97.4%, depending on the iteration, with a value of 91.3% across all iterations. In our opinion, these values show that we were able to convey the complex content on homomorphic encryption and its application in a comprehensible way with our lecture material.

4.3 Structure of the Lecture

Each of the lecture units deals with one topic, so that each of them is a closed unit and can, in theory, even be held without the other units. Of course, sometimes some prior knowledge is required, but this design allows for the simple integration of certain topics into other lectures. They all then follow the same general structure. This means that in the first part, the (mathematical) background that is required to understand the topic is presented and explained. Depending on the complexity, an example application of it is presented. For example, after the LWE problem is introduced, a toy example is presented together with an example construction of a simple cryptosystem before the connection to HE is presented. The overall topic is then the "highlight" of each unit and typically is either the end or only followed by a few slides discussing possible applications or showing connections to other topics. The last slide is then always a summary of the topics discussed in the unit. An example is given in Figure 3. These slides are there for two reasons. First, they help the students to quickly get an overview of what was done in the unit. Second, they foster sustainable learning since key concepts are reiterated, emphasizing the most important part of the unit.

5 Lecture Content

Our lecture consists of 8 slots structured as follows. The first lecture motivates the need for homomorphic encryption and provides a high-level overview of the idea of homomorphic encryption. We refresh the student's knowledge on basic security principles like data privacy, confidentiality, data integrity, and access control, and follow up with a refresher on the very basic mathematical concepts

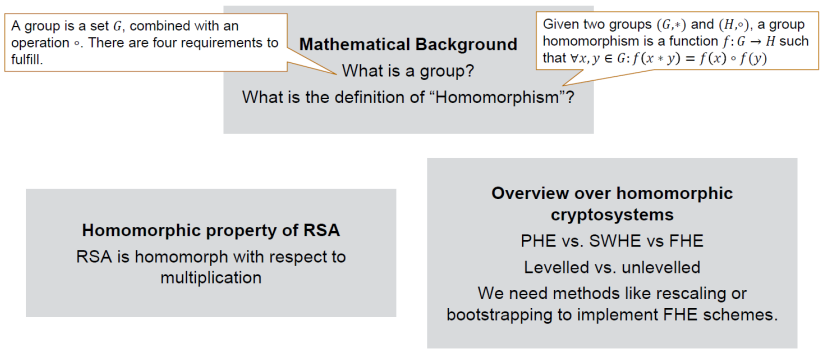

Fig. 3. Summary slide of the first lecture unit

like group axioms and common basic algebraic structures. Then the definition of a group homomorphism is introduced. As a practical example we show the homomorphic property of the well known RSA cryptosystem [21] followed by a short history section of homomorphic cryptosystems, starting with Partially Homomorphic Encryption (PHE) schemes supporting only a single mathematical operation up to Fully Homomorphic Encryption (FHE) schemes we know today.

In the second lecture, we introduce basic mathematical concepts from the ground up, in order for students to be able to follow the more advanced mathematical concepts that the modern CKKS cryptosystem is based on. Beginning with an introduction to modular arithmetic, we define the LWE search and decision problems and show how one can build a crude homomorphic cryptosystem on the basis of a toy example. The explanations are assisted with visual aids to make them easy to follow. As most modern homomorphic cryptosystems are based on so-called lattice problems, we define and explain the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP) lattice problems and emphasize their importance in post-quantum cryptography.

In the third lecture, we introduce the core mathematical concepts required for understanding the CKKS cryptosystem. We start by introducing polynomial rings and polynomial residue class rings, followed by complex numbers and the root of unity. The lecture is concluded with the concepts of antisymmetrical vectors and Vandermonde matrices, as well as the concept of coordinate-wise random rounding.

Starting with lecture three, the students have obtained the required knowledge to learn about the actual algorithms used in the CKKS cryptosystem. We start with the encoding and decoding algorithms by explaining every step with the help of a numerical example on the blackboard, allowing students to fully understand the details and get a feel for the complexity behind each operation. Following that, the key generation step, as well as the encryption and decryption of vectors, is explained in the same manner. Throughout the lecture, we make sure the domains the functions map between are always kept in mind. The

overview slide that is depicted in Figure 4 is therefore shown multiple times. In the beginning, not all blocks may make sense, but the more algorithms are explained, the better the students understand the colors and domains. Next, addition and multiplication, as well as the associated relinearization algorithm, are explained in detail. For all the algorithms, we do not include fully mathematically sound proofs; rather, we show by example that the algorithms work as expected. This is because, to understand the proofs, more mathematical and theoretical knowledge is required, and thus, some units are needed to explain them. In our lecture, however, the focus is more on applying and using the techniques instead of their theoretical fundamentals.

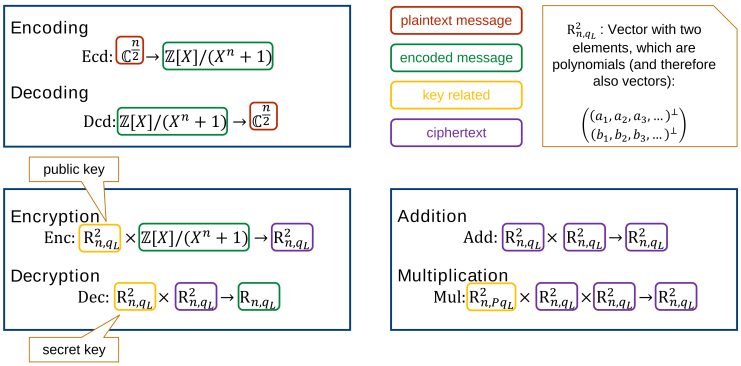

Fig. 4. Summary slide of the CKKS algorithms

In the fourth lecture, the rescaling and bootstraping concepts are explained on a higher level without diving into the mathematical details. Instead, we follow up the CKKS scheme with a guide on the right choice of security parameters. This is done with the exercises in mind because – as explained in Section 6 – the lecture and the exercise sheets are strongly coupled and the OpenFHE C++ library [1] is frequently used there. To use it, the security parameters need to be set properly. To avoid just copying them from tutorials, the lecture introduces them together with their reasoning.

As one limitation of the CKKS cryptosystem is the missing support of computing higher-order mathematical functions natively, we introduce the students to different methods for computing functions like the square root or division by only using addition and multiplication. The lecture slot is based on the findings of Prantl et al. [20], who used a modified Newton-Raphson method as well as Taylor series to calculate these functions, as well as the exponential and logarithm functions, and find the minimum and maximum of a series of numbers.

Lecture slot number six takes it one step further by explaining ways to implement the binary step function as well as approaches for allowing limited conditional branching by showing how equals and smaller than functions can be

implemented homomorphically. The goal is for students to learn to think with those limitations in mind. Ultimately, they will understand the possibilities and challenges of current state-of-the-art homomorphic cryptosystems.

The last slot of our course concludes the lecture by showcasing three more complex real-world algorithms, which combine the knowledge of concepts and functions acquired throughout previous lectures. Namely, the Box-Cox transformation [20], Linear Regression learning [19], and the k-means clustering algorithm, their implementation details, and performance measurements are demonstrated.

6 Exercises

In the approach, we mentioned that there are exercise sheets for every unit. Since some units are split into multiple lectures, there are nine sheets, which is one more than the number of units. They are all designed to cover the material from the previous lecture session. In a classic setting where the lecture is held over the course of one term, the students have one week to work on the tasks. In each week, there is a tutorial session where the solutions to the tasks are presented and the students can ask questions about the tasks or the lecture in general. The presentation of the solutions is not restricted to the lecturer. As many of the tasks are designed to get a hands-on approach, the students shall be encouraged to present the solutions themselves. There may also be some variations in the solutions since there are often multiple ways to solve and especially in HE the number of multiplications makes a key difference. If multiple students present their solution, the lecturer is able to show where there may be opportunities to save multiplications or which implementation techniques can cause problems in this context, although they may work without a problem in the non-encrypted context. The general structure of the exercise sheets consists of exercises from three categories: Practicing, Applying, and Transferring. The first consists of only calculation tasks. This could be, for example, the execution of some algorithms that are used in the CKKS cryptosystem. We consider this to be important for a better understanding of the algorithms. It is one thing to see and read them, but another to calculate them themselves. Afterwards, it is easier to remember them, and one also gets a feeling of what the pitfalls are in these algorithms and why they work the way they do. The second is almost the same as the first. Here, it is not just simple calculating but taking something that has been presented in the lecture and applying it to a new problem. An example is the following task:

> *Task Description:*
> Show that the function $f\colon \mathbb{R} \to \mathbb{R},\ f \mapsto ax + b$ with $a, b \in \mathbb{R}$ and $b \neq 0$ is not a homomorphism between the groups $(\mathbb{R}, +)$ and $(\mathbb{R}, +)$.

The students need to apply the concept of the homomorphism to this new problem. While doing so, they learn how the homomorphism works and develop a

better understanding of it. Furthermore, they are getting a more general overview outside of the HE context so that it is easier for them later to use it in other contexts and in the more complex topics of the lecture. The third is transferring the knowledge obtained in the lecture to other domains and developing new ideas. An example task for that is the following:

> *Task Description:*
> The Paillier scheme is a partial homomorphic scheme introduced in 1999 [18]. The key generation yields a public key (n, g) and a secret key (p, q). Encryption Algorithm: For each message $m \in \mathbb{Z}_n$ the number $r \in \mathbb{Z}_{n^2}^*$ is randomly chosen and the encryption works as follows:
>
> $$c = E(m) = g^m r^n \mod n^2$$
>
> Proof that the Paillier scheme is a partial homomorphic scheme.

Hereby, the students are invited to think about the concepts presented in the lecture. There, the definition of partial systems was given on a high level, and the students were introduced to groups. Now, they have to think about how they can connect all these concepts and transfer them to this new problem. Although this concrete exercise is not overly complicated, it can not be solved by just copying something from the lecture. Without having a basic understanding of what was presented there, it can not be solved. The combination of these three types of exercises leads to a better understanding of the students of the topic. This is because they not only repeat what has been done but also build up new connections by transferring their knowledge. A lot of the exercises from the second category require the usage of the OpenFHE library [1]. This is the current state-of-the-art library in HE that implements the most frequently used cryptosystems and is thus used in the majority of the recently published publications. Therefore, using it in the exercise sheets has the big advantage that the students are getting familiar with it and can use it later to do their own research and implement projects. If we had used another platform that may have been easier to start with, the hurdle to do something in the field of HE would have been much higher, as a new library would have needed to be learned. This corresponds with the general aim of the lecture, which is to focus on practical usability. This is also the reason why a lot of the tasks focus on implementation. Specifically, all algorithms that apply the cryptosystem and use it to calculate something, for example, the Newton-Raphson principle, are covered in the sheets. However, not only are they present, but also variations and other algorithms. The goal is to showcase why they are not as suitable as the other ones. For example, one task asks to implement the Taylor series to approximate the square root. The students see that this method is good around the development point, but fails to be precise the further one goes away. This raises awareness for the reasons why some methods are better than others and when they should be chosen. If the students do all the exercise sheets, they will have implemented some of the most important functions and can use them later on in their own research projects.

Overall speaking, the exercise sheets provide a mix of deepening the knowledge, getting a hands-on experience and also transferring the knowledge of the lecture to other domains.

7 Conclusion

Advancing digitalization is increasingly permeating all aspects of our lives, and the collection and processing of large amounts of data to train AI models is playing an ever more central role. With this trend, the paradigm of privacy-friendly computing is increasingly coming into focus, in order to protect personal data and, at the same time, to enable its use for training AI models. One promising technology for implementing privacy-friendly computing is homomorphic encryption. This cryptographic technique allows us to perform calculations directly on encrypted data without having to decrypt it first. In order to leverage the potential of homomorphic encryption and facilitate the transfer of knowledge from research to practice, it is essential to familiarize computer science students with the basic theoretical ideas behind homomorphic cryptosystems and how they can be extended for practical applications. However, there is currently a lack of lectures on this topic that combine the theoretical and practical aspects. For this reason, in this paper, we present a newly developed lecture that aims to close this gap. The aim of our lecture is to provide a balanced combination of theory teaching and practical application. We designed our lecture material according to the principle of Design-Based Research - an iterative approach that aims to iteratively design and evaluate the created materials. We held our lecture at three German universities, accompanied by comprehensive evaluations. The feedback from students shows that we have succeeded in creating a suitable lecture that meets our predefined goals. With this paper, we would like to share our lecture materials and findings with the professional community and lecturers to contribute to the further development of teaching in the field of applied privacy-preserving computing.

References

1. Al Badawi, A., Bates, J., Bergamaschi, F., Cousins, D.B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., Liu, Z., Micciancio, D., Quah, I., Polyakov, Y., R.V., S., Rohloff, K., Saylor, J., Suponitsky, D., Triplett, M., Vaikuntanathan, V., Zucca, V.: OpenFHE: Open-Source Fully Homomorphic Encryption Library. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. pp. 53–63. WAHC'22, Association for Computing Machinery, New York, NY, USA (Nov 2022). https://doi.org/10.1145/3560827.3563379
2. Barak, B.: An Intensive Introduction to Cryptography. `https://cs127.boazbarak.org/` (2021), Accessed: 07.05.2025
3. Brakerski, Z.: Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. pp. 868–886. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_50

4. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. pp. 309–325. ITCS '12, Association for Computing Machinery, New York, NY, USA (Jan 2012). https://doi.org/10.1145/2090236.2090262
5. Brown, A.L.: Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. Journal of the Learning Sciences **2**(2), 141–178 (Apr 1992). https://doi.org/10.1207/s15327809jls0202_2
6. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 409–437. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
7. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: Fast Fully Homomorphic Encryption Over the Torus. Journal of Cryptology **33**(1), 34–91 (Jan 2020). https://doi.org/10.1007/s00145-019-09319-x
8. Clarivate: Web of Science: Citation Report: Homomorphic Encryption
9. Collins, A.: Toward a Design Science of Education. In: Scanlon, E., O'Shea, T. (eds.) New Directions in Educational Technology. pp. 15–22. Springer, Berlin, Heidelberg (1992). https://doi.org/10.1007/978-3-642-77750-9_2
10. Ducas, L., Micciancio, D.: FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015. pp. 617–640. Springer, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_24
11. Fan, J., Vercauteren, F.: Somewhat Practical Fully Homomorphic Encryption (2012)
12. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. pp. 169–178. ACM, Bethesda MD USA (May 2009). https://doi.org/10.1145/1536414.1536440
13. Gentry, C.: Computing arbitrary functions of encrypted data. Communications of the ACM **53**(3), 97–105 (Mar 2010). https://doi.org/10.1145/1666420.1666444
14. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman & Hall/CRC Cryptography and Network Security, CRC Press, Boca Raton London New York, second edition edn. (2015)
15. Kleppmann, M., Hugenroth, D.: Cryptography and Protocol Engineering. `https://www.cl.cam.ac.uk/teaching/2425/P79/` (2024), Accessed: 07.05.2025
16. McKenney, S., Reeves, T.C.: Educational Design Research. In: Spector, J.M., Merrill, M.D., Elen, J., Bishop, M.J. (eds.) Handbook of Research on Educational Communications and Technology, pp. 131–140. Springer, New York, NY (2014). https://doi.org/10.1007/978-1-4614-3185-5_11
17. Micciancio, D.: Advanced Cryptography (FHE). `https://cseweb.ucsd.edu/classes/wi23/cse208-a` (2023), Accessed: 07.05.2025
18. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) Advances in Cryptology — EUROCRYPT '99. pp. 223–238. Springer, Berlin, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
19. Prantl, T., Engel, S., Horn, L., Kaiser, D., Iffländer, L., Bauer, A., Krupitzer, C., Kounev, S.: Performance Impact Analysis of Homomorphic Encryption: A Case Study Using Linear Regression as an Example. In: Meng, W., Yan, Z., Piuri, V. (eds.) Information Security Practice and Experience. Lecture Notes in Computer Science, vol. 14341, pp. 284–298. Springer Nature Singapore, Singapore (2023). https://doi.org/10.1007/978-981-99-7032-2_17

20. Prantl, T., Horn, L., Engel, S., Iffländer, L., Beierlieb, L., Krupitzer, C., Bauer, A., Sakarvadia, M., Foster, I., Kounev, S.: De Bello Homomorphico: Investigation of the extensibility of the OpenFHE library with basic mathematical functions by means of common approaches using the example of the CKKS cryptosystem. International Journal of Information Security **23**(2), 1149–1169 (Apr 2024). https://doi.org/10.1007/s10207-023-00781-0
21. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (Feb 1978). https://doi.org/10.1145/359340.359342
22. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Foundations of secure computation **4**(11), 169–180 (1978)
23. Scott, E.E., Wenderoth, M.P., Doherty, J.H.: Design-Based Research: A Methodology to Extend and Enrich Biology Education Research. CBE—Life Sciences Education **19**(3), es11 (Sep 2020). https://doi.org/10.1187/cbe.19-11-0245
24. Venturi, D.: Data Privacy and Security. `https://dventuri83.github.io/projects/2_dps` (2024), Accessed: 07.05.2025
25. WhatsApp: Introducing Advanced Chat Privacy: Enhanced Protection for Your Most Sensitive Conversations. `https://blog.whatsapp.com/introducing-advanced-chat-privacy?lang=en` (Apr 2025)
26. Wu, D., Bradley, E.: Cryptography. `https://www.cs.utexas.edu/~dwu4/courses/sp25/index.html` (2024), Accessed: 07.05.2025