Thesis for The Degree of Doctor of Philosophy

Cryptographic Tools for Privacy Preservation

Carlo Brunetta

Department of Computer Science & Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2021

Cryptographic Tools for Privacy Preservation

Carlo Brunetta

ISBN 978-91-7905-528-8
Doktorsavhandlingar vid Chalmers tekniska högskola, Ny serie nr 4995.
ISSN 0346-718X

Technical Report No 205D
Department of Computer Science & Engineering
Chalmers University of Technology
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2021.

"Breathe, breathe in the air...
...don't be afraid to care..."

"Breathe" - Pink Floyd

Abstract

Data permeates every aspect of our daily life and it is the backbone of our digitalized society. Smartphones, smartwatches and many more smart devices measure, collect, modify and share data in what is known as the Internet of Things.

Often, these devices don't have enough computation power/storage space thus outsourcing some aspects of the data management to the Cloud. Outsourcing computation/storage to a third party poses natural questions regarding the security and privacy of the shared sensitive data.

Intuitively, Cryptography is a toolset of primitives/protocols of which security properties are formally proven while Privacy typically captures additional social/legislative requirements that relate more to the concept of *"trust"* between people, *"how"* data is used and/or *"who"* has access to data. This thesis separates the concepts by introducing an abstract model that classifies data leaks into different types of breaches. Each class represents a specific requirement/goal related to cryptography, *e.g.* confidentiality or integrity, or related to privacy, *e.g.* liability, sensitive data management and more.

The thesis contains cryptographic tools designed to provide privacy guarantees for different application scenarios. In more details, the thesis:

(a) defines *new encryption schemes that provide formal privacy guarantees* such as theoretical privacy definitions like Differential Privacy (DP), or concrete privacy-oriented applications covered by existing regulations such as the European General Data Protection Regulation (GDPR);

(b) proposes new tools and procedures for providing *verifiable computation's* guarantees in concrete scenarios for post-quantum cryptography or generalisation of signature schemes;

(c) proposes *a methodology for utilising Machine Learning* (ML) for analysing the effective security and privacy of a crypto-tool and, dually, proposes a secure primitive that allows computing specific ML algorithm in a privacy-preserving way;

(d) provides an *alternative protocol for secure communication* between two parties, based on the idea of communicating in a periodically timed fashion.

Keywords

Cryptography, Privacy, Outsourced Computation, Cloud Computing, Verifiability

Acknowledgment

Let me start by thanking my supervisor, **Katerina**. The many conference's rejections, the long research visiting, the pandemic and many other complications. It was definitely tough (for both of us) but it was fun and educational!

Next, I would like to thank my co-supervisor **Bei**. Always prepared and ready to keep the work going and, additionally, an amazing office-mate with whom share mundane discussions regarding food, politics, economics and food (again, yes). I'm looking forward to coming to visit you in Beijing, either for work or for (food) holidays!

It is really hard to write a complete list of names, but I want to deeply thank all the many people in **my division/unit, Network and System**, for sharing the good/bad moments of the daily work. Additionally, thanks to all the **administration "moms and dads"** for all the support that they gave me either *"work-related"* or *"it's a blue day"*. **Tack <3**

A big thank you to the uncountable number of **friends** that crossed my life here in Chalmers. Thank you for all the good Fika, the beers and all the afterworks that made the journey a little more chilled.

A special *"efharisto poli!"* goes to a (crazy) friend and co-worker, **Georgia**. Thank you for all the laughs, drinks and philosophical discussions! It was really nice to share all these crazy years together. I wish you to finish the PhD soon, the best for your future, a lot of luck and to continue travelling the world!

Another special thanks go to **Pablo, Lara, Oliver and Erik**. It is definitely a pleasure seeing an amazing family grow and I really wish you the best of luck for all the future challenges!

An enormous *graxie* goes to **Elena**, my unofficial tutor, guide and co-worker and, mainly, an amazing Friend, with capital *"F"*. You helped me a lot at the beginning of my *crazy journey*. You and your amazing wife **Hedvig** were always there to give good, sincere advice and meaningful help. It is definitely hard to explain our (Aura's and mine) gratitude for how amazing you two are and I will not even try. I prefer to promise that we will continue sharing our path, share the good and bad moments and try to get together and having a good meal, a couple of drinks and enjoy every moment together, anywhere on Earth. (==)

Outside work, I was incredibly lucky to have found a multitude of incredibly amazing Friends, again with capital *"F"*. We shared different hobbies, interests, opinions but, most of all, we shared many unforgettable, meaningful and once-in-a-lifetime moments. Thank you to all my Sahlgrenska's real-science friends **Tugce, Lydia, Eleni, Axel, Giacomo, Alina, Masako and many more** and the *"climbing monkeys"* **Jasmine, Eridan, Clement and Katja**. Of course, I cannot forget to thank new friends like **Martin, Simon, Johan, Isabel, Veronica** and the old ones like **Alberto "Benjo", Davide, Kevin, Kekko, Andrea, Giorgia, Mia, Marta, Gloria, Silvia, Fede, Gloria, Alice, Mattia, Dylan, Seba, Casa, Costa, Tommy, J, Anto, Maru, Fox and many, many, many others**. Furthermore, a special thanks go to the *"Band with Many Names"* composed by **Marco, Enzo, Pier, Evgenii and Grischa**. We

shared a lot of adventures and I'm really grateful for all the good bohemian moments and improvised jams. Our band will always be a beautiful memory in my musical career.

Thank you, my Friends, for every moment. It is indeed the $\mathbb{R}$-life and I know that our path might diverge. Already many of us are getting married, having our first baby and/or moving to other countries. Our lives are slowly turning to different paths and I feel a little sad about it. But if I feel sad, it means that I cared a lot about our Friendship thus I wish you all the best for your career, family, happiness and any type of goal. We will definitely meet again, one day, and just *"synchronize"* our new experiences, adventures and achievements!

Before moving to the emotional side of the section, I want to thanks Aura's family: **Paolo, Manuela and Alice**. Grazie per avermi accolto nella vostra famiglia e per tutto il sostegno e l'aiuto che date a me e ad Aura. Grazie mille per tutto!

I have to switch to my dialect to properly thank my family, **Bepi, Reza and Gigi**.

Prima de tut, Graxie, con la *"G"* granda. So de esar al fiol pì casinaro e so benisimo che no le stat facile vedarme volar via all'estero, cresar così velocemente quasi da no riconosérme pì. Ma savee benisimo che se son così bravo in tel me laoro, rispetà e amà da tuti i me Amighi, a le solo graxie a come che me avé cresest. Graxie par averme soportà, par averve *"cavà al pan dala boca"* par darme an futuro diverso, milior de quel che avé pasa voi. So che mi e Gigi sion i vostri punti de orgoglio. Dove saver anca voi che son sempre fiero, orgoglioso e content de chi che sie, dei vosti enormi sacrifici e dela enorme umiltà che ve rende così unici. Se son quel che son, a le solo graxie a voi. E ora che la pension se avicina, vede de goderve al vostro meritato riposo. Graxie de tut.

As you might expect, I left the best for the end!

Thank you, my love. Really really thank you, **Aura**. You fill my days of love, energy and (good) no-sense, you give me reasons for fighting for a better Universe, you definitely make me a better human being (a quite awkward but still a better one!). I love you and you know it. No infinite amount of ink can describe how important you are. We share the best moments of our lives and I'm so eager to see where our future will bring us. All the new adventures, achievements and challenges.

You always call me your *"Mountain"* because I sometimes make you mad when I'm too introverted, cold and harsh. But, on the positive side, I'm there, stable and calm, ready to shift the whole Universe only for seeing you happy. Coming from the Alps and by looking at my personality and hobbies, I definitely feel like a Mountain.

You are my precious *Stella Alpina*.

You are part of me and you make me important, you make me proud of who I am, you make me want to protect you from all the tourists that are trying to pick you up and that doesn't know how strong you really are[1]. I believe that the *"Sea"* better represents who you are. Peaceful but incredibly strong, deep but calm on the surface. We complete each other, I'm the Mountain, you are the Sea.

I'm not a good swimmer (as we can agree from this last holiday) but there is something that I love doing: I love to look at the horizon, being from the top of a mountain or the shoreside. It makes me think of the past, the present and the future. It brings me peace, as you do, every day. And whenever you bring me peace, I'm able to appreciate all the love you give me and, to me, *our love is all that matters.*

[1] Stelle alpine are astonishingly strong and brave! Like, deciding to live in between harsh rocky terrain, under the freezing winter snow and strong winds only to pop out in the late summer, to enjoy the sun and the immense silence and peace that only the highest mountains can provide. That's hardcore!

I may have forgotten amazing people that crossed my work and personal life. I'm technically writing this section during my holidays, so sorry about my bad memory! If you are not on this list, don't feel angry. Just let me know and I will offer you a drink!

I concluded my licentiate acknowledgement with a quote from the masterpiece *"Dark Side of the Moon"* and I admit that it is a perfect summary even now:

> For long you live and high you fly,
> and smile you'll give and tears you'll cry,
> and all you touch and all you see,
> is all your life will ever be.
>
> *Breathe* - Pink Floyd

Appended Publications

This thesis is based on the following publications:

Paper A: **C. Brunetta**, C. Dimitrakakis, B. Liang, A. Mitrokotsa
"A Differentially Private Encryption Scheme"
20-th Information Security Conference (ISC), 2017, Ho Chi Minh city (Viet Nam). Spinger, LNCS, Vol. 11124, 2017, pg. 309326. [BDLM17]

Paper B: E. Pagnin, **C. Brunetta**, P. Picazo-Sanchez
"HIKE: Walking the Privacy Trail"
17th International Conference on Cryptology And Network Security (CANS), 2018, Naples (Italy). Springer, LNCS, Vol. 10599, 2018, pg. 4366 [PBP18]

Paper C: **C. Brunetta**, B. Liang, A. Mitrokotsa
"Lattice-Based Simulatable VRFs: Challenges and Future Directions"
1st Workshop in the 12th International Conference on Provable Security (PROVSEC), 2018, Jeju (Rep. of Korea) and *Journal of Internet Services and Information Security, Vol. 8, No. 4 (November, 2018).* [BLM18]

Paper D: **C. Brunetta**, B. Liang, A. Mitrokotsa
"Code-Based Zero Knowledge PRF Arguments"
22-th Information Security Conference (ISC), 2019, New York (USA). Spinger, LNCS, Vol. 11723, 2019, pg. 171-189. [BLM19]

Paper E: **C. Brunetta**, B. Liang, A. Mitrokotsa
"Towards Stronger Functional Signatures"
Manuscript.

Paper F: **C. Brunetta**, P. Picazo-Sanchez
"Modelling Cryptographic Distinguishers Using Machine Learning"
Journal of Cryptographic Engineering (July 2021), [BP21].

Paper G: **C. Brunetta**, G. Tsaloli, B. Liang, G. Banegas, A. Mitrokotsa
"Non-Interactive, Secure Verifiable Aggregation for Decentralized, Privacy-Preserving Learning"
To appear in 26th Australasian Conference on Information Security and Privacy (ACISP), 2021, Perth (Australia).

Paper H: **C. Brunetta**, M. Larangeira, B. Liang, A. Mitrokotsa, K. Tanaka
"Turn Based Communication Channel"
Manuscript under submission.

Other publications

The following publications were published during my PhD studies, or are currently under submission. However, they are not appended to this thesis.

(a) **C. Brunetta**, M. Calderini, and M. Sala
"On hidden sums compatible with a given block cipher diffusion layer"
Discrete Mathematics (Journal), Vol. 342 Issue 2, 2018 [BCS19]

(b) G. Tsaloli, B. Liang, **C. Brunetta**, G. Banegas, A. Mitrokotsa
"DEVA: Decentralized, Verifiable Secure Aggregation for Privacy-Preserving Learning"
Manuscript under submission.

Research Contributions

Paper A: I was involved in the initial brainstorming with Aikaterini and Christos who proposed me the idea of including differential privacy in the cryptographic domain. I had the idea of relaxing the correctness property of an encryption scheme, the key idea that allows defining differentially private encryption schemes. I further formalized, defined and proved all the contents of the paper. In the final stage, I wrote the implementation and the statistical tests.

Paper B: after many fruitful morning-fika and brainstorming with Elena and Pablo (and Oliver!), we all together traced the main structure and motivation for the HIKE protocol. During the development of the paper, I was the relay figure for the translation between theory and implementation. More specifically, I wrote the draft of some proofs and I was responsible for the theoretical aspects necessary for the implementation. Finally, I am the corresponding author of this work and I finalised the camera-ready version.

Paper C: I participated in the initial brainstorming discussion with Bei and Aikaterini after Bei's suggestion on the specific topic of constructing a post-quantum verifiable Pseudo-Random Function. I completely wrote the first draft of the paper. After receiving some useful external feedback on the paper, I participated in finding different possible solutions while Bei and Aikaterini revised the draft. In this final and much shorter version, I conceived the summary of the entire research-exploration and I was responsible for the introduction-background sections of the final paper.

Paper D: Bei, Aikaterini and I jointly discussed the possibility of extending Paper C's methodology for code-based cryptographic assumptions. I discovered and developed the content of the paper, wrote proofs and I completely wrote the first draft of the paper. After receiving some useful feedback on the paper from Bei and Aikaterini, I finalised the paper.

Paper E: I participated in the initial brainstorming discussion with Bei and Aikaterini after Bei's suggestion on the specific topic of providing construction for extending the Functional Signature primitive with a verifiability property. I was responsible for designing the Strong Functional Signature (SFS) instantiation with the related security proofs. In the first draft, I wrote the instantiation, security proofs and general introduction. After receiving some useful external feedback on the paper, I took the responsibility of revising the SFS's primitive, security model/properties and the application described in the introduction.

Paper F: after several discussion with Pablo, we together traced the main structure and motivation for a methodology for generating cryptographic distinguishers using machine learning. I was leading the project and developing the theoretical framework. I designed and performed the statistical analysis of our framework's experiment. I wrote the majority of the first draft and handled the journal communications.

Paper G: I joined the discussion with Georgia, Bei, Aikaterini and Gustavo regarding distributed federated learning. Concurrently, I designed a non-interactive primitive while Georgia and Bei defined DEVA (Paper b). After receiving some useful external feedback on the first paper, we jointly decided to split the constructions into two papers (Paper G and b) and I took the responsibility for my construction's paper. Thus, I defined and proved the security of NIVA, I wrote the first paper draft and I helped to debug minor problems in the implementation.

Paper H: I had the original idea of developing a turned communication channel which I later developed initially with Aikaterini and Bei and later with Mario and Keisuke during a research visit. I led the project and, in the first paper version, I wrote the initial draft of the protocol's construction and introduction's section. I double-checked the fairness security and proof that Mario and Keisuke wrote. After receiving some useful external feedback on the paper, we decided to split the paper into concrete instantiation and the theoretical implications of our construction. Currently, Paper H contains the protocol instantiation that I initially wrote.

All the co-authors agree on the preceding statements.

Thesis Contents

Introduction

Every single day
Every word you say
Every game you play
Every night you stay
I'll be watching you

Every Breath You Take - The Police

Our society lives in an era where every device, electronic or not, is becoming *"smart"*. Smartphones, smartwatches, smart glasses are examples of many new devices that are continuously being constructed and introduced in our daily life. All these *smart devices* are designed to improve productivity, automatise tasks and track complex procedures. This is possible by providing the devices with the ability to manage **data** by providing them with *computational power* and the ability to *communicate* with each other.

More precisely, the adjective *"smart"* relates to the device's ability to handle *"data management"* which can be classified into the actions of *(i)* **generating**; *(ii)* **communicating**; *(iii)* **storing**; and *(iv)* **computing/manipulating** data. In other terms, a smart device is a *"standard"* device that incorporates a computer-like microcontroller able to capture the device status, manipulate the information and communicate it to other smart devices.

This simple concept allows the consideration of *hyperconnected networks* of (often low) computational devices, better known as the *Internet of Things* (IoT). The IoT principle is based on the ubiquitous presence of cheap and low-computational devices that constantly generate, collect, manipulate and share data locally between themselves or with a *"higher entity"* called *the Cloud*.

For example, consider the thesis' writer, Carlo, that lives in a *smart home*, *i.e.* a home where lights, smart electro-domestic and more sensors/actuators are interconnected on the same home-local network. All the data collected throughout the house is, often, centrally collected on a house-router that later uploads part of the data to an external service *"on the Cloud"*. Abstractly, the Cloud is an *interface of data management services* that any authorised smart device contacts via the Internet and utilises to *"simplify"* the data processing. Despite the Orwellian feeling of massively collecting data and centralising it into a single external entity, the Cloud provides useful analysis to the router and allows Carlo to better control every measurable aspect of the home.

For example, Carlo might be highly interested in maintaining high-quality air in his home. To do so, Carlo's house is filled with air-quality sensors that collect pollution data, send it to the central router which later *"ask the Cloud"* for an analysis. Since this collecting-analysis is continuously executed, Carlo has the power to check the air-pollution in his house at every moment. This means that Carlo can voice-activate its home-assistant device and ask *"which room has the cleanest air?"*, the device will record Carlo's command and upload the recording to some voice-recognition service *"on the Cloud"* that will transliterate the command's request.

Whenever the home assistant receives the request transcription, it will ask the home-router an answer which will, most probably, *"contact the Cloud"* that will analyse the request and reply to the router with the answer. After all this back and forth, the router will provide the assistant with the answer that can effectively be announced verbally to Carlo after just a couple of seconds.

The careful reader might notice the writer's highlight of actions referred to *"into/to the Cloud"*. The reason for such pedant highlight is the necessity to take a step back and precisely delineate the concrete reality of the Cloud's *"composition"*. Similarly to the atmospheric homonymous and depicted in Fig. 1, the Cloud is a network conglomeration of smaller networks of computers, all interconnected and orchestrated to appear as a *"hyper-computer"*, *i.e.* a computer with incredible computational power, unimaginable storage capacity, extremely efficient communication bandwidth and always available. The quintessential aspect is that *"to use the Cloud"*, the user **does not** need to know where these computers are, their characteristics, how they operate or how they are organised. The writer's highlight wants to point out that *"uploading to the Cloud"* is, fundamentally, semantic sugar for *"uploading to some unknown-but-retrievable computer on the Internet"*.

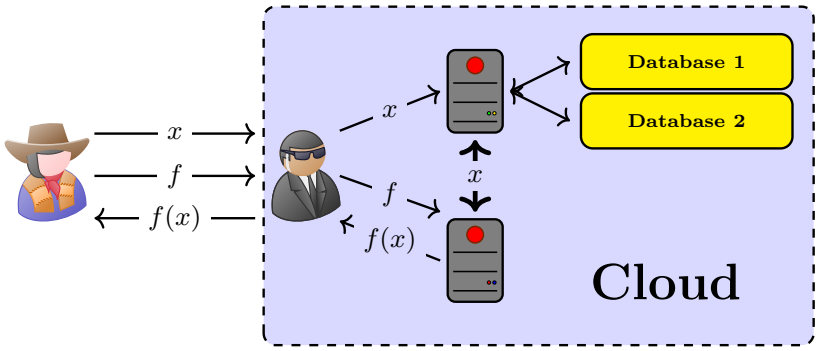

Figure 1: Picturesque representation of the Cloud's composition.

Data is **the** fundamental element of our digital society and imposes a remarkable role on our digital identity. Generated data can either be *public* or *sensitive/private* depending on the *data owner* thus requiring different confidentiality guarantees whenever handled. The IoT paradigm is based on having the smart devices execute part of the data management via cloud computing which, concretely, can be seen as simply requiring the devices to *outsource computation* to a more powerful computer. In other words, all the devices' data is handled by *unknown* computers on the Internet.

How is it possible to ***trust*** *the Cloud to* ***properly handle*** *user's sensitive data?*
What does it mean "to trust someone" *and* "properly handle data"*?*

Throughout history, humans evolved their *secrecy's needs* into the **cryptography** discipline. Figuratively, cryptography is *the toolset* of algorithms and protocols that allows the user to provide confidentiality, integrity, authenticity and many other properties that handle sensitive data. As in any proper toolset, there are several tools from *must-have screwdrivers*, such as the Diffie-Hellman's key-agreement protocol, to multi-purpose *Swiss Army-knifes*, such as the Fully Homomorphic Encryption (FHE) schemes. The main objective for all cryptographic tools is to avoid any data leaks, *i.e.* each one of these tools is designed to provide precise *security guarantees* which are formally defined and mathematically proven, *e.g.* confidentiality, integrity, authentication, anonymity

and many more. The use of formal modelling is fundamental to unequivocally describe how a cryptographic tool must be used to achieve the security guarantees when it can be used and all the limitation that it might have. The usage of mathematics for describing the cryptographic elements allows us to firmly state that a provably secure crypto-tool can not be the cause of a *data leak*, *i.e.* the scenario in which a malicious entity can disrupt/break the provided tool's security guarantees. On the contrary, if an adversary can *"break the crypto-tool"*, then either the cryptographic primitive/protocol or the security model used is not secure thus it is impossible to formally prove the tool's security or model's usefulness.

Often used in daily conversations, a different concept to consider is **privacy**. The main goal for privacy is complex and it is highly related to *how data is used* and *how to prevent data to be harmful* which require an extensive analysis of the application that requests privacy guarantees. Each privacy guarantee is an *"interdimensional"* requirement that spans from cryptographic security requirements to real juridical liability, business' responsibility or human necessities. In a nutshell, the concept of privacy is *"the framework"* that provides real/legal guarantees to people that their data is not misused in a harmful way.

Privacy and cryptography define a *spectrum* of requirements that describes the trade-off between *security and usefulness* and can be associated with the concept of *trust*. On one side of the spectrum, we have the *"no-trust"* scenario where the user's data is required to be secret, where no one else than the data owner can access the data. On the other side, the *"only-trust"* scenario where the same user's data might be communicated unencrypted with the only requirement of *"not misusing this information"*.

Hidden in the scenario's description, the spectrum naturally introduces the concept of *shared data* between users, *i.e.* someone else's private data which shouldn't be misused. Any privacy guarantees require shared data to be protected because it requires the data owner to trust the receiver not to misuse such sensitive information. At a first glance, protecting shared data might appear as a different way to name private/secret data but it is essential to understand that it is possible to lose all the privacy guarantees without breaking any used cryptographic tool. Consider a user that securely uploads to the Cloud a private photo of him/her and let the user fully trust the Cloud to maintain the necessary secrecy. Despite the cryptographic guarantees that the communication is secure, the photo is most probably unencrypted for the Cloud which utilises the photo for improving its services, *e.g.* trains classifiers for better face recognition. Without breaking any crypto tool, the Cloud can break the user's trust and publicly release the private photo thus breaking the trust agreement between itself and the user.

This discrepancy between cryptographic and privacy requirements is described in several legal regulations such as the California Consumer Privacy Act (CCPA) of 2018 [Par18] or the European General Data Protection Regulation (GDPR) [Cou16]. These regulations, and many more, provide a *legal foundation* that precisely state which user's data is *sensitive* thus requiring the Cloud's special care while handling the data. The regulations further describe precise liability penalties whenever a user's data is misused. For example, the user's IP address is sensitive information that can be maliciously used to approximately geo-localise the user or track him/her throughout the web. It is fundamentally impossible to navigate the web without revealing the personal IP address thus the servers must correctly handle this, and other, sensitive data. Otherwise, the users can bring the server's owner to court for misusing sensitive data.

To understand the differences between cryptographic and privacy guarantees and further provide future research directions in the intersection area of cryptography and privacy, it is mandatory to provide an abstract analysis of all the possible data leakage that might occur between any interaction of two entities.

1 Abstract Model for Data Leaks

People own collections of personal data and each one of them partitions the collection based on the specific data's **sensitivity**. More formally each person P_A classifies data into the collections of:

- **private data** $\mathfrak{C}$ that contains any information that P_A is **not willing to share** with anyone else. These are highly sensitive data that a malicious entity can use to seriously harm P_A thus must be carefully handled;
- **shared data** $\mathfrak{S}$ that contains P_A's private data that is **consensually shared** with a different person P_B. Because such data is technically private, P_A must trust P_B to not misuse/publish the shared data. On the other hand, P_B uses the data to provide some form of benefit to P_A, *e.g.* a personalised service. This data collection is strictly connected to trust and the concept of privacy;
- **public data** $\mathfrak{P}$ that contains P_A's public data that is **freely shared** with anyone. Ownership of such data cannot be used to harm P_A and are therefore easily retrievable.

For example, Carlo considers the data x = "work email address" to be public while ξ = "personal email address" is more sensitive so it is only shared with selected other people and web services. Consider the last example where Carlo considers ξ = "personal email address" $\in \mathfrak{S}$ and uses ξ to register to a generic social network $\mathcal{N}$. A (quite typical) scenario is that the social network $\mathcal{N}$ will publicly display ξ by default because $\mathcal{N}$ considers $\xi \in \mathfrak{P}$. This notion is condensed into the following axiom:

Informal Axiom 1. *Data partitioning is* ***subjective****,* i.e. *every person P has his/her way of partition data into* $(\mathfrak{C}_P, \mathfrak{S}_P, \mathfrak{P}_P)$.

Sadly, Informal Axiom 1 implies that deciding the sensitivity of a specific data is **ill-defined**, *i.e.* it is not possible to uniquely identify the correct partition to which data belongs, as previously described.

Additionally, data appears to be *"naturally entangled"* with other data, as if it is semantically interconnected. Intuitively, from big sets of information, it is possible to **infer** new information, maybe without absolute certainty thus requiring some probabilistic discussion. For example, if Carlo would present itself with a wet umbrella, the reader can deduce that it is raining outside. Or, by observing Carlo's smartphone screen, the reader can infer his usage pattern by analysing the *"oily"* residues left on the screen. Furthermore, Sherlock Holmes might be able to deduce the pin-code digits' used to unlock the phone by analysing the shape of the oily fingerprints. By carefully reading the examples, observe that Carlo might be unaware of how his data can be maliciously used when combined with *"advanced detective's knowledge"*.

Informal Axiom 2. *Data is* ***always dependent*** *on other data: for every information z, there always exists a set $\{x_i\}_{i\in I}$ that infers about z,* i.e. $\{x_i\}_{i\in I} \to z$.

Informal Axiom 2 describes two negative corollaries which state, from some known information x, the impossibility to compute *(i)* all the *inferable* data z, *i.e.* all the z such that $x \to z$; and *(ii)* all the data-sets $\{z_i\}_{i\in I}$ that infers about x, *i.e.* $\{z_i\}_{i\in I} \to x$.

The axioms allow the analysis of all the possible inference between the different sensitivity partitions, *e.g.* the inferences that take private data $\{s_i\}_{i\in I} \subseteq \mathfrak{C}$ and infers a public information $y \in \mathfrak{P}$. By conceptually reasoning on the empirical meaning of such deductions, the final result is an abstract model that describes a classification of any

data leak into four semantically different breaches, represented in Fig. 2 and named: *(i)* security breach; *(ii)* direct breach; *(iii)* coercion breach; *(iv)* indirect breach.

Before moving to a precise analysis of each breach, it is important to remark on an indirect consequence of Informal Axiom 1. As in any good model, the data leak classification into breaches is *relative to the observer*, *i.e.* the leak might hurt P_A but benefit P_B and it is caused by their different data sensitivity partitioning.

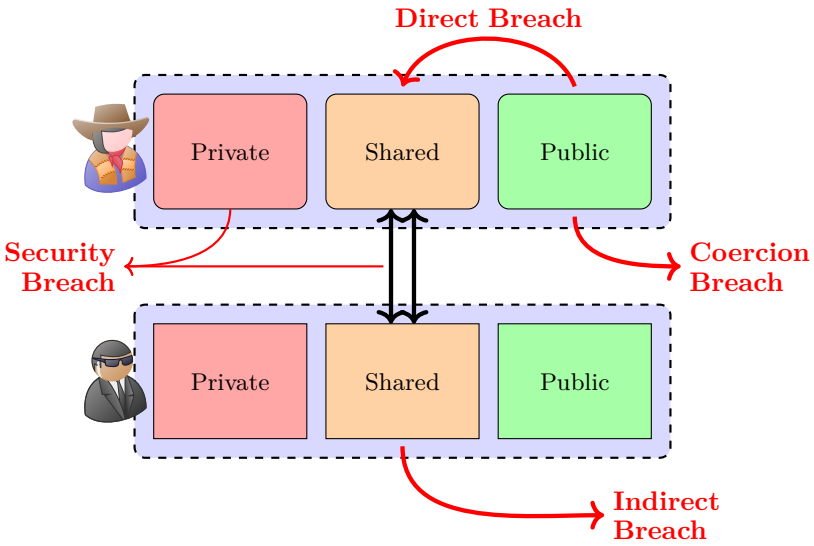

Figure 2: Data leak's model from the cowgirl's point of view. The black arrows indicate the communication between the parties. The red arrows indicate all the possible data leaks.

1.1 Security Breach

Security breaches are defined whenever an adversary $\mathcal{A}$ can *"break"* the cryptographic primitives/protocols used and the security properties requested, *e.g.* $\mathcal{A}$ decrypts an encrypted database of private data or can compromise the integrity of a secure communication channel.

A historical and didactical example is the cryptanalysis advances that, during the Second World War, allowed the Allied powers to break the encrypting machine *Enigma* used by the Axis powers. Preceding and motivating the development of the first computers, Enigma is an electro-mechanical encrypting device that appears to have a physical typewriter-like keyboard and display of light-emitting characters representing the keyboard. To encrypt, the operator presses a single character key which closes an internal electrical circuit that lights up a precise character in the display. Internally, the machine is composed of rotors that rotate at every typed character, modifying the circuit and the highlighted encrypted output, as represented in Fig. 3. The security of the device is due to the immense amount of possible starting combinations of the rotors and other external additional modifications of the circuit made via a plugboard. Enigma was considered *unbreakable*.

During the war, the Allied power developed the theoretical foundations of Inform-

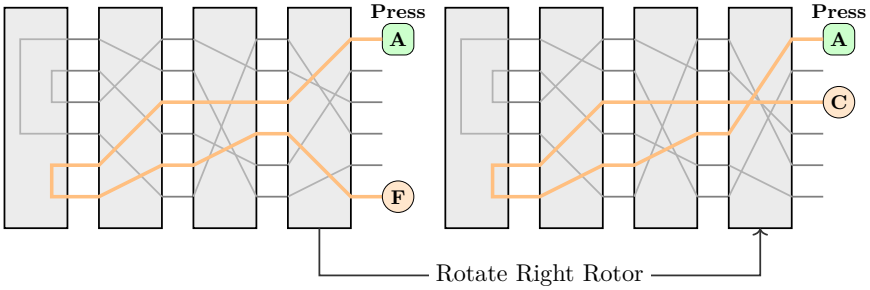

Figure 3: Conceptual illustration of the Enigma machine's encryption principle.

ation Theory [Sha48] and Cryptanalysis. Briefly speaking, together with practical examples of correct decryption, code-books and capturing some Enigma machines, this new knowledge allowed a refinement on the brute-forced decryption attacks which allowed to decrypt the secret communication and provide useful intelligence on the field. In other words, Enigma was *broken*.

In the same spirit, security breaches happen because either the cryptographical knowledge evolves and new successful attacks are being developed or, more simply, the wrong crypto-tool is used. The state-of-the-art primitives/protocols are secure up until the hypothesis used to formally prove the tools' security guarantees holds. This requires researchers to constantly check that new attacks don't break such hypothesis and promptly report to the community whenever a crypto-tool is broken.

1.2 Direct Breach

Direct breaches are defined whenever it is possible to deduce private/shared data from public ones. Despite the simple definition, these breaches are intrinsically sneaky to identify and prevent.

In October 2006, the on-demand streaming service Netflix released a dataset containing hundreds of millions of *private* movie ratings generated by half a million subscribers. The release's purpose was to allow the development of an improved movie recommendation system. To guarantee privacy, the dataset was *anonymised*, *i.e.* the subscriber's sensitive data such as user id, email addresses and even the timestamp of the rating submission was removed. In principle, *only public data was released*.

A couple of years later, Narayanan and Shmatikov [NS08] were able to de-anonymise the identity of known subscribers from Netflix's dataset and obtain his/her movie ratings, thus discovering unexpected sensitive information such as political preferences. Such a surprising result was possible by considering *additional information* such as the one retrievable by personally asking naive questions like *"what do you think about this movie genre?"* or, more systematically, utilise the *public* movie ratings provided by the Internet Movie DataBase (IMDB). The reader might argue that *"de-anonymising movie ratings don't sound harmful"* but consider the scenario where a malicious adversary $\mathcal{A}$ can de-anonymise the identity of the ratings. Only because $\mathcal{A}$ can de-anonymise people from their *"movie tastes"*, $\mathcal{A}$ can *profile* the unlucky subscriber and increase the ability to track him/her throughout the Internet.

Direct breaches are caused by the Informal Axiom 2 and the impossibility to conceive all the possible deductions that public information can provide. Conceptually, note that it is not obvious *how* cryptographic tools can protect from such breaches. For

this reason, the state-of-the-art solution is found in the concept of **Differential Privacy** [DMNS06] (DP) which provides a formal framework to measure the privacy loss of publishing specific data related to a dataset. To understand how DP works, consider a private dataset of values $\{x_i\}_{i=1}^n$ on which it is required to compute the known function f. The computed output $\mu = f(x_1, \cdots, x_n)$ is publicly released thus meaning that $\{x_i\}_{i=1}^n \to \mu$. Without loss of generality, by cleverly modifying the function's input, it might be possible to obtain the public value $\mu' = f(x_2, \cdots, x_n)$ in which the private data x_1 is **not used**. The direct breach, as represented in Fig. 4, is caused by considering the function f and the public outputs μ, μ' and observing that any *difference* between outputs must relate with x_1, *i.e.* the breach tries to deduce $\{\mu, \mu', f\} \to x_1$.

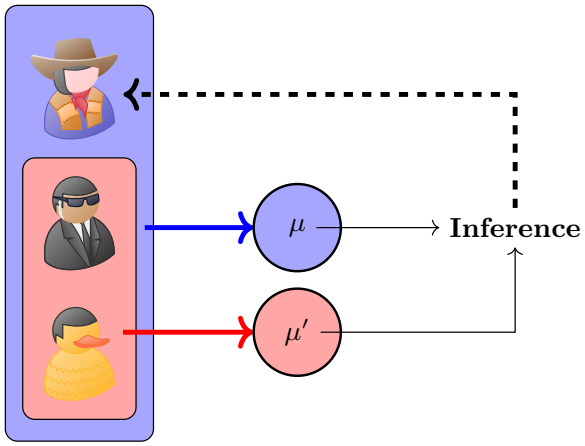

Figure 4: Depiction of the problem solved by the differential privacy framework.

DP provides a methodology to *measure* the privacy loss caused by releasing f's outputs and, to avoid the breach, a DP mechanism adds **noise** which is sampled by a cleverly selected distribution based on the previous measurements. The key concept of *adding cleverly selected noise* might sound counterproductive but finds roots in the idea of *"degrading the information accuracy"*. For example, by publishing Carlo's birth season instead of the month, the probability of guessing his birthdate is degraded thus a loss inaccuracy.

1.3 Coercion Breach

To understand what coercion breaches are, consider public information x related to some private data of the person P_A. Since x is public, a malicious adversary $\mathcal{A}$ might voluntarily advertise a *false*-statement x' that hurts P_A's image/reputation. The *"coercion"* adjective appears whenever considering that, to clarify that x' is false and x is true, P_A must provide private data y so to allow the inference $y \to x$ thus the adversarial coercion.

A real example of such malicious persuasion can be found in the widespread phenomenon of **media distortion** in which fake news are most probably the easier attack vector. Without entering the immense domain of human psychology, it is well-known that people can easily be influenced by only providing modified photos or provide emotionally intense messages. These cheap modifications are repeatedly shown to allow people to unconsciously change their mind regarding, *e.g.* political beliefs [AG17] or

memories of well-known historical events [SAL07]. The social damaging impact of media distortion through fake news is massive and must be prevented.

Coercion breaches are an undesired consequence of Informal Axiom 2 and the fact that often private data is necessary to understand how public data is deduced. Avoiding these breaches is a *tricky problem* that requires taking into consideration the social aspects of human psychology and it seems counter-intuitive that a cryptographic tool might help.

A possible solution would require appropriate experts to educate people on *digital etiquette* and *critical thinking*, *e.g.* by teaching the importance of source verification and awareness of possible media distortion practices. Observe that the appropriate usage of crypto-tools can help to discover data misuse by providing specific security guarantees or, naively, people might be aware of the *meaning* of the tools guarantees.

1.4 Indirect Breach

The last class in our model are the indirect breaches which are a negative consequence of sharing private data x to some other person P which is trusted to not misuse x. Whenever P misuses x, the assumed trust is lost and there is a data leak and the indirect breach. Whenever reading, in our daily life, news about data leaks and related privacy loss, often the news describes an indirect breach.

Purely for explanatory reasons, consider a *run tracking application*, *i.e.* web application that allows users to collect data, such as their heartbeat, pace and much more, from their running activities with the benefit of providing statistics, professional training advices and more user control on their activities. One such application is Strava [Str18] which allows users to provide precise geo-localisation data, *i.e.* GPS-data. Later on, the users visualise the GPS-data on a map thus allowing each user to correlate, *e.g.*, their pace with the topological morphology of the terrain. Strava, like all the others, is often trusted by its users to securely handle the sensitive data, *e.g.* GPS-data is commonly accepted and shown to be incredibly sensitive data [SSM14].

Having a lot of data allows providing interesting features to the users. One of them is Strava's *"popular routes"* which collects the users' GPS-data, finds highly popular routes and provides a popularity list where users now can find each other and share a training session. The feature has the noble motivation of creating a healthy community and increasing the social interaction between the users.

At the beginning of 2018, the noble feature showcased as a popular route a too-regularly shaped one in a scarcely populated, almost desertic, part of Afghanistan. By carefully checking the satellite image of the route, it was possible to discover a *secret* military base [Her28]. An unaware American soldier was periodically training inside the military base, running around an aircraft's runway thus creating a regularly shaped route. Strava's popular route algorithm worked as intended: the soldier was one of the few people in the whole area using the app which implied that his periodically tracked route was the most popular. The indirect breach, consequent trust-loss and legal cost for the data leak's harmful potential were caused by the soldiers' unawareness of Strava's feature **and** Strava's misjudgement on the sensitivity of using the soldier's GPS-data.

In general terms, it is easy to see that indirect breaches are caused by Informal Axiom 1 and the fact that different people have a different opinion regarding data sensitivity. Trust is a difficult concept to generally formalise thus, to avoid such costly damages, many state-of-the-art cryptographic protocols provide some specific privacy guarantees that allow preventing the leak.

A noticeable mention, of a whole research field that tries to avoid indirect breaches, is the research in *Information Flow Control* (IFC). IFC is based on the simple principle that whenever computing an algorithm on data, the algorithm must not be able to output private data given in input, depicted in Fig. 5. In other words, whenever the input is private, specific computational operations are *"prohibited"* because they might be reverted to get the input. By studying the *"allowed"* operations, it is possible to check which algorithms are immune to indirect breaches and are therefore safely executable.

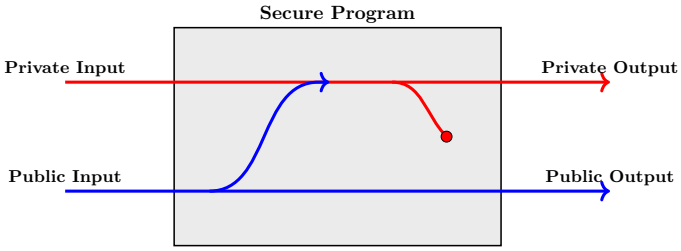

Figure 5: Conceptual representation of the Information Flow Control principle: a secure program does not manipulate the private input and reveals it into the public output.

> Gentlemen. Your communication lines are vulnerable, your fire exits need to be monitored, your rent-a-cops are a tad under-trained...
> Outside of that, everything seems to be just fine. You'll be getting our full report and analysis in a few days, but first, who's got my check?
>
> *Sneakers (1992)* - Martin Bishop (Robert Redford)

As previously stated, it is the research community goal to provide solutions that allow to *"trust the Cloud"* or to avoid any possible data leaks.

The **quintessential research goal** for **any** cryptographic solution that handles people's data is to **avoid data leakages, of any form**.

In other words, *ideal* cryptographic privacy-preserving tools must guarantee *(1)* a tamper-proof data generation; *(2)* secure data communication; *(3)* confidential and privacy-oriented data storage; and *(4)* data computation with *measurable* privacy guarantees, *i.e.* the computed outputs must not reveal *"too much"*.

A key concept that allows reducing the gap between ideal and real solutions is **verifiability**, *i.e.* the property of providing a tangible value used as *"proof"* of either the knowledge of specific information or *certification* of approval. Many existing cryptographic tools already provide verifiability-like guarantees such as:

- signature schemes allow a signer to attach a *signature* to the outgoing messages which can be seen as proof that *"the signer notarises the message content"*. The message-signature pair verification strictly relates to some form of liability that the signer obtains in the act of signing;
- authenticated communication channels, *e.g.* TLS, allow the communicating parties to securely communicate **and** provide the guarantees that only the intended/authorised parties participate in the communication. This is possible by the combination of several different cryptographic tools that are singularly correct and verifiable and that guarantees the confidentiality of the communication and the authenticity of the parties identities;
- in applications, zero-knowledge proofs allow a prover to prove a public statement without revealing the knowledge of a secret witness that easily proves the statement. Being able to provide such verification has profound application scenarios connected to privacy, liability, anonymity and more.

All the described examples provide verifiability for what the user sees or knows and can easily provide verifiability guarantees to data generation, storage and communication. *"Securing data computation"* and providing *"measurable privacy guarantees"* are the missing requirements to tackle.

Data manipulation transforms potentially sensitive information into *new* data that might get published thus having the potential of creating privacy concerns. Quantifying the privacy loss from publishing a computational output is generally hard to compute and/or to correctly and practically handle. For this reason and by observing the problem from a different perspective, it is easier to request **proofs of correct computation** on the data and **control which computation is performed**. It is trivial to see that providing a refined control on the computable functions allows to bound the complexity of computing the privacy loss. Indeed, a trade-off between functionalities and privacy must be considered whenever effectively implementing the system.

Verifying the correct computation of a function allows the verifier to check that the results are indeed correct **and** the correct function was computed. In other words, if something went wrong and the verification fails, the verifier can *identify* the problem, *e.g.* the verifier can precisely shift the data-misuse liability to some entity that later must defend against accusations in the court and not in the cryptographic domain.

To guarantee any form of privacy, it is fundamental to identify any data misuse which is only possible if *every step of the data management is verified*. Ideally, providing (formally provable) verification to every cryptographic tool allows to prevent any data leak:

- any direct breach is caused by a careless release of outputs which allows inferring sensitive data. Requiring the verifiability of the output computation **does not** directly avoid such privacy loss **but** it limits the available computable functions, thus limiting the possible malicious inferences, *and* completely shifts the liability to the publisher. In a sense, these data leaks are solved with the mantra: *"Be aware of what they publish"*;

- verifiability completely solves any coercion breaches since it allows to correctly pinpoint the trustworthiness of the provided data. It must be said that it is always important to **provide a proof** for the computed results and, respectively, to always **request proofs** of the content authenticity;

- security breaches are directly related to the formal security properties that the cryptographic primitives/protocols should achieve. Technically, verifiability is often an additional security property with a really specific description. In other words, the motto is *"always use proven secure and verifiable cryptographic tools"*;

- indirect breaches are always caused by breaking the data owner's trust. Verifiability can prevent these breaches whenever privacy is considered such as **design principles** for new cryptographic tools by providing certainty that the tools are correctly used.

The reality is that to avoid unexpected data leaks, cryptographic tools must be correctly implemented and used as theoretically intended, *i.e.* the purpose they are designed for. The *purpose* is important: there might exist a cryptographic tool that is considered highly secure by the research community, but it is not designed for privacy-oriented applications.

This thesis' goal is to investigate and design new cryptographic primitives/protocols that consider privacy as a fundamental design requirement. By increasing the crypto-toolset with new privacy-preserving crypto-tools, it is possible to choose the appropriate primitive/protocol for real applications thus guaranteeing privacy and security for everyone.

2 Thesis Contributions

This thesis considers several privacy-oriented problems and proposes solutions that formally provide security and privacy-preservation guarantees.

2.1 Differential Privacy and Cryptography

A fundamental principle in Cryptography is that an encryption scheme has to be *correct* and *confidential*, *i.e.* the ciphertext's decryption **must** be the original message and the message cannot be inferred by the ciphertext. Differently, a differentially private (DP) mechanism allows data to maintain privacy when revealed and this is done by introducing a cleverly sampled random noise. Observe that a DP mechanism does not require any confidentiality requirement. This observation brings up the question of combining the two feature:

Question A: A Differentially Private Encryption Scheme

*Is there a way to define/construct a differentially private encryption scheme that guarantees confidentiality while data is encrypted **and** afterwards provides a measurable privacy guarantee?*

Paper A consider an encryption scheme and a DP mechanism as a framework and it studies the relation between them to merge them into a single cryptographic primitive.

Contribution: we *relax* the encryption scheme's correctness property. Intuitively, the encryption scheme has to *"wrongly decrypt"* with some bounded and predefined probability, *i.e.* the ciphertext's decryption can return a wrong message m' with some probability $\alpha_{\mathsf{m},\mathsf{m}'}$ that depends on the original message m and the final wrong message m'. The knowledge of such probabilities allows us to prove that the *"faulty"* encryption scheme indeed achieves differential privacy. Additionally, an implementation is provided as a proof-of-concept.

To complete the study, we prove that using such *"faulty"* encryption schemes is equivalent to sequentially using a correct encryption scheme **and** a DP mechanism as two separate frameworks, as depicted in Fig. 6.

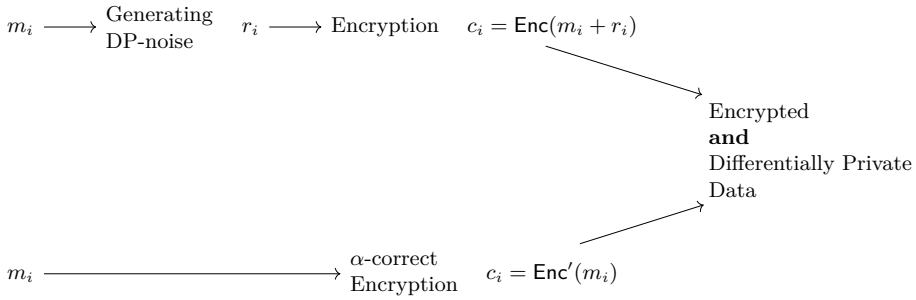

Figure 6: Paper A: The difference between the DP-then-Encrypt (on the top) and our solution (at the bottom).

This means that if we want to introduce differential privacy to already existing products/protocols, it is not required to change the already existing cryptographic

primitives but it is only necessary to introduce a DP mechanism in the system and correctly compose it with the encryption scheme.

2.2 Real Privacy Guarantees by Design

The main goal of Paper B is to provide a model/scheme with an implementation designed to provide privacy guarantees concerning privacy policies/regulations, such as the GDPR, that are not always described in mathematical formalism. By considering the scenario of a user uploading data to a trusted database that can be queried by third parties, the paper answers the following question:

Question B: HIKE: Walking the Privacy Trail

Is it possible to design privacy-preserving protocols that comply with some privacy policies, such as the European GDPR?

We start by selecting some specific articles contained in the GDPR and describe them as formal cryptographic properties:

(a) data has to be encrypted when stored;

(b) the user decides to selectively allow third parties to access his/her data; and

(c) the user can always delete his/her data from the database (*right to be forgotten*).

Contribution: to describe the *"client, cloud and service provider"* model, we use the concept of a *labelled encryption scheme* [BCF17] in which every message, or ciphertext, has a *label* that can be seen as a unique public identifier for that message. With these labels and the associativity and commutativity of the underlying group, we can define *decryption tokens* that can be generated by the client. This allows the user to create decryption tokens for specific label-ciphertexts and provide them to a service provider.

We exploit the additive homomorphic property of the encryption scheme to allow homomorphic evaluations on the client's ciphertexts. In this context, the client can generate decryption tokens for *labelled-programs*, *i.e.* the token necessary to decrypt a specific homomorphic evaluation and defined by the list of inputs, related labels and function to be computed. Since the function must be known to produce the decryption token, the clients can refuse to provide the token and therefore **not disclose** their data.

More concretely, we start from the ElGamal encryption scheme [ElG85], we describe the scheme as a labelled encryption scheme called LEEG, expand it with some specific features regarding the decryption token into FEET and finally obtaining the HIKE protocol, depicted in Fig. 7, that is then proven secure in the GDPR-oriented security model we defined.

As a final contribution, all our ideas are implemented and our code for the HIKE protocol is publicly available.

2.3 Post-Quantum Verifiable Pseudorandomness

Quantum computers are the currently accepted future of computation. Despite the engineering challenges of constructing such a revolutionary machine, the cryptographic research community is interested in providing new primitives that are guaranteed to be secure even against adversaries that use a quantum computer.

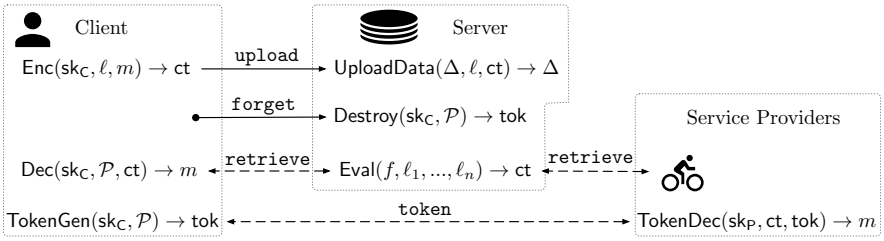

Figure 7: From Paper B: The HIKE protocol.

In particular, we focus on *verifiable random functions* (VRFs) and in particular on **simulatable VRFs** (sVRFs). In a nutshell, sVRFs are a family of VRFs in a public parameter security model, such as the common reference string.

Question C: Lattice sVRF: Challenges and Future Directions

*Is it possible to define a **post-quantum** sVRF, based on lattice assumptions?*

Contribution: Paper C proposes the possibility of defining a **lattice-based membership hard with efficient sampling** language which can be used to define a lattice-based *dual-mode commitment scheme*. We partially conjecture the possibility to combine the dual-mode commitment scheme with Libert *et al.*'s protocol [LLNW17] and Lindell's transformation [Lin15] and obtain an sVRF under post-quantum assumptions, as represented in Fig. 8. Given the non-triviality of the task, we raise and identify different open challenges in lattice-based cryptography and possible future directions for achieving a post-quantum sVRF.

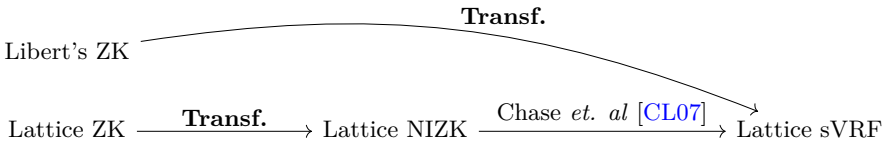

Figure 8: Paper C: A roadmap to lattice-based sVRF.

On a similar note, we ask ourselves:

Question D: Code-Based Zero Knowledge PRF Arguments

*Is it possible to utilize a similar methodology as for Question C to define a **code-based** post-quantum zero-knowledge argument protocol?*

Contribution: Paper D utilizes the idea underlying Paper C by transforming a code-based PRG into a PRF for then introducing a methodology to effectively provide a zero knowledge argument for the code-based PRF evaluation. We propose a concrete construction and theoretically estimate the communication cost of our construction. Additionally, we introduce the *whistle-blower notary problem*, represented in Fig. 9, of which Paper C and D's results are possible solutions.

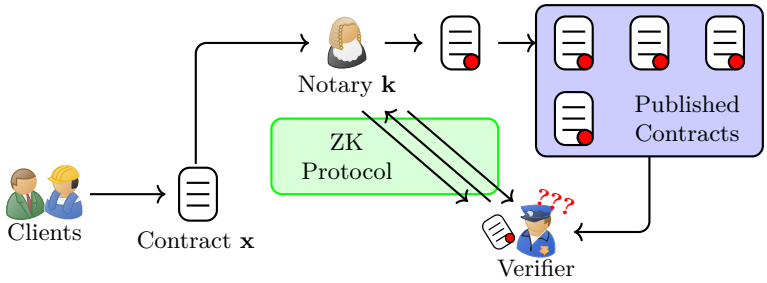

Figure 9: Paper D: The whistle-blower notary problem.

2.4 Verifying Functional Signature Evaluation

Signature schemes are a fundamental tool in today's application. They allow using a signing secret key to compute a signature from any message which later can be publicly verified with a public verification key and prove the authenticity of the content and the signer identity. A generalization of signature schemes is proposed by *Functional Signatures* (FS) in which the signer owns a *functional* signing key that allows signing a *specific function evaluation*. In other words, a functional signature allows authenticating the output of the function evaluation, therefore, hiding the original input.

An additional property provided by FS is *function hiding* in which it is impossible to infer which function got evaluated during the signature phase. In this way, verifying the signature correctness has two meanings: *(a)* the signature somehow verifies the correct evaluation of **a** function; and *(b)* the signature does not reveal **which** function got evaluated.

In a real application, often the signing key must be revoked which introduces a fundamental problem for FS: the function hiding property makes it impossible to know *which* signing key was used which means that the verification algorithm cannot effectively alert that a specific signature is generated from a revoked key.

Question E: Towards Stronger Functional Signatures

Is it possible to design a functional signature-like scheme that allows a more refined function evaluation verification ***but*** *preserves function privacy?*

Contribution: Paper E introduces the concept of *Strong Functional Signatures* (SFS), an FS-like scheme that introduces a public functional verification key that is publicly available and used during the verification phase. In a realistic application, such as the one represented in Fig. 10, all such public keys can be collected and publicly maintained by a trusted curator and allow key revocation by simply removing (or similar) the specific public key. SFS provides function hiding by requiring that both the signature **and** any functional verification public key hides which function is evaluated during the signing phase.

Our instantiation merges the Boneh-Lynn-Shacham's signature (BLS) scheme [BLS04] and Fiore-Gennaro's publicly verifiable computation (VC) scheme [FG12] under a shared *master* key pair used for the functional key generation and the final verification. Whenever generating the functional key pair, our instantiation first generates the VC's keys for the requested function and obtains the secret, evaluation and verification keys. Afterwards, the BLS's signing keys are generated with the addition of including additional information regarding the function **and** the VC's secret key. In this way, all the generated

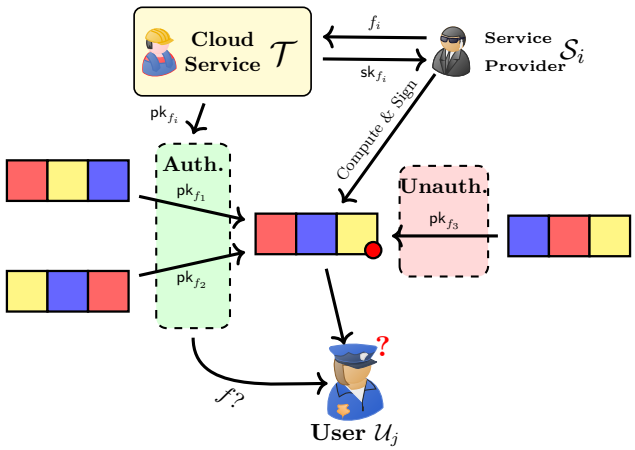

Figure 10: Paper E: Strong functional signatures in the cloud computational authentication scenario.

function's VC and BLS keys are related to each other.

The SFS's signing algorithm computes the VC evaluation and computes the BLS signature of the result which is later verified during the final verification. Our instantiation provides unforgeability by exploiting a design trick: a tamper must be a *"wrong evaluation"* which is signed with a BLS's key. Since the keys are all related, signing the wrong result will always create a wrong signature and if the BLS signature has correctly tampered with, then the tampered result must be the correct function evaluation which is not a tamper.

2.5 Machine Learning as a Tool for Cryptanalysis

Security is a complicated matter that can often be abstracted into *"hiding data's patterns"* while preserving some *"recovery"* property. Cryptanalysis is the research branch that applies several statistical, algorithmic and/or mathematical methodologies to find patterns in data to weaken or even destroy any security claim. The simplest form of such a methodology is based on solving a *distinguishing problem* in which an algorithm can classify the inputs between two (or more) different classes. The classical example is the *ciphersuite distinguishing problem* in which an algorithm takes in input a ciphertext and must output *"which is the encryption scheme used"*.

Machine Learning (ML) is a growing research area that provides a framework for investigating statistical correlations on specific datasets, often to extrapolate a classifier later used for analysing a new dataset.

Question F: Modelling Cryptographic Distinguishers Using Machine Learning

Can machine learning be used to automatize cryptanalysis?

Contribution: Paper F proposes an abstract methodology that allows to effectively use of ML for creating cryptographic distinguishers and provides some simple technique to improve the efficiency of such ML classifiers. Our methodology is depicted in Fig. 11.

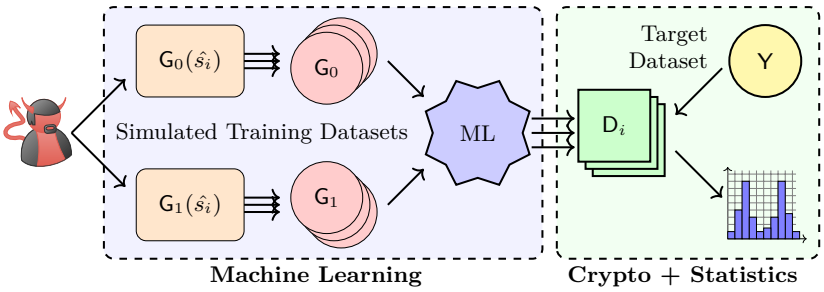

Figure 11: Paper F: Abstract representation of our methodology.

We implement our methodology in an expandable framework and create a simple proof-of-concept experiment in which we study the possibility of utilizing an ML generated distinguisher for distinguishing between several National Institute of Standard and Technology (NIST) Deterministic Random Bit Generators.

2.6 Secure Aggregation for Federated Learning

Federated Learning (FL) is a novel paradigm oriented to allow the aggregation of ML classifiers between several users with special consideration in achieving high privacy guarantees. The first privacy-preserving design concept is that each user pre-computes its ML model locally and it is not required to provide the raw data to the aggregating server. Only the computed model is used in the aggregation, therefore requiring the aggregation protocol to protect the user's model privacy.

Current solutions are focused on providing an interactive protocol between the users and a *single* central server that facilitates communication coordination. The interactivity of the protocol handles users that *drop out* from the protocol execution because either they lose their connection or they are maliciously trying to deny the service execution. Furthermore, the aggregating server is a *single-point-of-failure*. In an extreme scenario, an adversary might crash the central server and the protocol will abort without any recovery possibility.

Our specific interest is to additionally require the aggregating server to provide a proof that allows the users to verify the correctness of the servers computation.

Question G: Non-Interactive Secure Verifiable Aggregation for Decentralized, Privacy-Preserving Learning

> *Is it possible to distribute the secure aggregation between several servers* ***and*** *remove the necessity of the user's interaction* ***and*** *provide verification of the server evaluation correctness?*

Paper G proposes NIVA, a non-interactive primitive inspired by Shamir's secret sharing scheme that allows users to distribute the aggregation between several servers of which a threshold amount is needed to correctly reconstruct the final output, as depicted in Fig. 12 We implement NIVA and compare the communicational costs against some state-of-the-art protocol.

Contribution: our construction extends the standard additive homomorphic secret sharing scheme by introducing a *"verification token"* that the user computes and which is related to the secret input and the servers. During the aggregation phase, the servers compute and release the secret-sharing partial aggregation value and a proof of correct

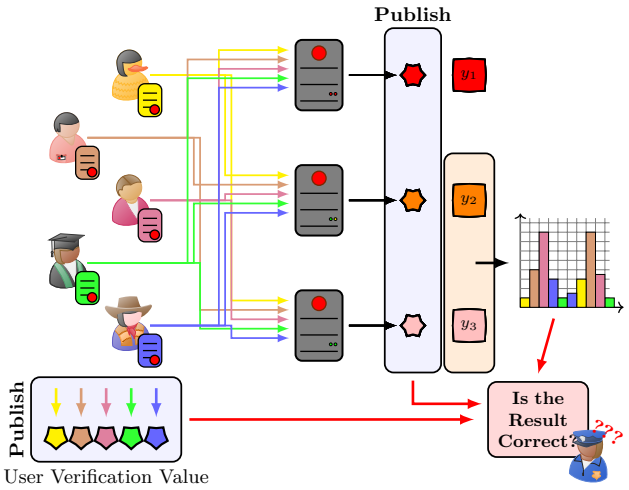

Figure 12: Paper G: Several users delegate the secure aggregation of their inputs to independent servers. A threshold amount of server's outputs is necessary to publicly reconstruct and verify the resulting aggregated value.

computation. The verification algorithm requires at least a threshold amount of server to be used to reconstruct the final aggregation **and** verify the computation correctness.

The confidentiality of the secret inputs is guaranteed by the underlying secret sharing scheme and the computational assumption used by the verification token. Differently, the scheme is proved to *never* be tamperable, *i.e.* any adversary is unable to provide a verifying wrong final aggregation result. The verification algorithm design allows to easily prove such a strong statement which boils down to an algebraic *"trick"*: the existence of an adversarial tamper depends on a pre-defined linear system which is easy to prove to **never** have a solution.

2.7 Alternative Communication Channels

The fundamental medium required for communicating is the *communication channel*. Different applications might require different *features*, *e.g.* we are interested in *consistent channels*. This means that the communication transcript is constantly verified during communication to prevent any *future* tampering of the *past* exchanged messages.

Blockchain is a novel technology that allows the creation of such a consistent channel. The only requirements are the *"complex"* assumptions necessary to create and use such a channel. Many blockchains require extensive use of signature schemes, public-key cryptography, hash functions and a *consensus mechanism*, often based on game-theoretic assumptions based on economical strategies.

Question H: Turn Based Communication Channel

> *Is it possible to create a consistent communication channel based on a minimal set of assumptions?*

Paper H assumes the existence of a timed hash function, *i.e.* a hash function that is computable always in the same amount of time Δ. With such a primitive, we describe a *turn-based communication channel* (TBCC), depicted in Fig. 13

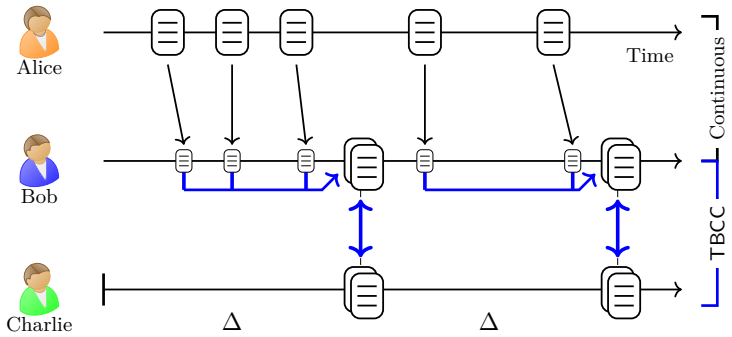

Figure 13: Paper H: A continuous and TBCC channel, the messages are gathered in "blocks", and each block, and its set of messages, is confirmed only at the end of each turn.

Contribution: we base our TBCC protocol on the idea of creating a verifiable *"commitment"* that can be verified only after solving a puzzle that requires a designed amount of time to be solved. Both the parties set up the communication by committing to a list of sequential puzzles which can only be solved in sequence. In this way, the parties start communicating committed messages that can only be periodically verified thus *emulating* a real turned communication where all the messages are exchanged periodically.

We provide a construction of the TBCC channel and prove that it provides communication consistency. This is possible because each exchanged message contains a *digest* of the previous communication thus making it impossible to tamper the communication without being noticed by the other party.

3 Summary and Future Directions

The papers contained in this thesis are testimony of the possibility of improving the crypto-toolset to incorporate privacy preservation and further allowing more secure solutions for real applications that requires to carefully handle people's sensitive data. Each one of the papers provides a novel cryptographic tool's instantiation that tackles a specific data leak, as summarised in Fig. 14.

Inevitably, data will increasingly be consumed by our evolving digital society and human understanding of data sensitivity will evolve accordingly, posing new security and privacy challenges to solve. For this reason, the research community **must** continue to develop new *verifiable* cryptographic tools that empower people and protect them from any harm caused by such a strong *data centricity*. Security and privacy are long-term requirements that must be incorporated in all the aspect of our society. More modest, shorter-term research directions would consider improvements such as:

- **Paper A** describes the possibility of easily introducing differential privacy in any cryptographic encryption scheme. On the other hand, it is left open the possibility to design different crypto-primitives that provides DP by design, *e.g.* would it be possible to create a DP signature scheme and which practical opportunities would it provide?

- **Paper B** provides a tailored GDPR-oriented solution called HIKE for a specific realistic application of outsourcing of both storage and computation. A direct

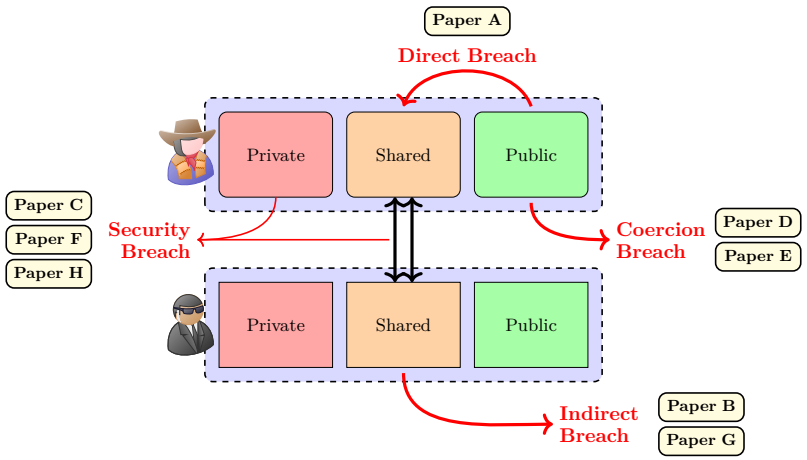

Figure 14: Paper's contributions and the correspondent data leak considered.

improvement would consist in either *(i)* further increasing HIKE's privacy requirements to cover more GDPR principles; *(ii)* simplifying the construction to improve efficiency; or *(iii)* introduce precise computation's verification requirements to guarantee security and privacy against stronger adversaries.

- **Paper C** and **Paper D** focus on the same goal of instantiating a simulatable verifiable random function. For both the papers, more research is necessary to allow them to be efficiently usable in practice. Additionally, different post-quantum cryptographic assumptions, *e.g.* isogenies, might be considered with the purpose of increasing the number of choices and allow the application to select the best-fitting primitive.

- **Paper E** provides the concept of strong functional signatures (SFS) and introduces an instantiation of SFS. A possible future direction would be to simplify the current instantiation, provide an efficient implementation and further investigate the possibility to define a general transformation that allows the instantiation of SFS from well-known cryptographic primitives.

- **Paper F** describes a methodology that enables the creation of crypto-distinguishers by utilising machine learning. It further provides an experimental analysis and an implementation. This paper's next step would be to improve the implementation by supporting additional machine learning algorithms and design a more practical and automatised framework. Of different motivation, it is of major interest the possibility to apply our methodology and check the concrete security of a real cryptographic system.

- **Paper G** introduces NIVA which is designed for federated learning's applications. Future directions would be focused on improving the primitive's efficiency and lowering the application requirements for secure usage of NIVA. From the practical side, NIVA should be implemented to be usable by the popular machine learning framework used by developers, *e.g.* TensorFlow.

- **Paper H** instantiates the concept of turn based communication channel (TBCC) and proves that the TBCC protocol achieves communication consistency. The

next step for TBCC would be to understand if it formally provides any cryptographic fairness property. Furthermore, a major investigation should be conducted to incorporate into the protocol more realistic assumptions, *e.g.* unpredictable communication delays.

Bibliography

[Adl83] Leonard M. Adleman. Implementing an Electronic Notary Public. In *Advances in Cryptology*, 1983.

[AFS05] Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A Family of Fast Syndrome Based Cryptographic Hash Functions. In *Progress in Cryptology – Mycrypt 2005*, 2005.

[AG17] Hunt Allcott and Matthew Gentzkow. Social Media and Fake News in the 2016 Election. *J. Econ. Perspect.*, 31(2), May 2017.

[AGM+13] Joseph A. Akinyele, Christina Garman, Ian Miers, Matthew W. Pagano, Michael Rushanan, Matthew Green, and Aviel D. Rubin. Charm: A framework for rapidly prototyping cryptosystems. *J Cryptogr Eng*, 3(2), June 2013.

[AGP16] Pablo Daniel Azar, Shafi Goldwasser, and Sunoo Park. How to Incentivize Data-Driven Collaboration Among Competing Parties. In *ITCS*, 2016.

[AGS11] C. Aguilar, P. Gaborit, and J. Schrek. A new zero-knowledge code based identification scheme with reduced communication. In *2011 IEEE Information Theory Workshop*, October 2011.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, July 1996.

[Alp14] Ethem Alpaydin. *Introduction to Machine Learning*. Third edition edition, 2014.

[AT17] Joël Alwen and Björn Tackmann. Moderately Hard Functions: Definition, Instantiations, and Applications. In *TCC*, 2017.

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable Delay Functions. In *CRYPTO*, volume 10991. 2018.

[BBCD20] Anubhab Baksi, Jakub Breier, Yi Chen, and Xiaoyang Dong. Machine learning assisted differential distinguishers for lightweight ciphers (extended version). 2020.

[BBF19] Dan Boneh, Benedikt Bünz, and Ben Fisch. Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains. In *CRYPTO*, 2019.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, January 2012.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, June 2013.

[BCF17] Manuel Barbosa, Dario Catalano, and Dario Fiore. Labeled Homomorphic Encryption. In *Computer Security – ESORICS 2017*, 2017.

[BCS19] Carlo Brunetta, Marco Calderini, and Massimiliano Sala. On hidden sums compatible with a given block cipher diffusion layer. *Discrete Math.*, 342(2), February 2019.

[BDD+20] Carsten Baum, Bernardo David, Rafael Dowsley, Jesper Buus Nielsen, and Sabine Oechsner. TARDIS: Time And Relative Delays In Simulation. Technical Report 537, 2020.

[BDL+16] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More Efficient Commitments from Structured Lattice Assumptions. Technical Report 997, 2016.

[BDLM17] Carlo Brunetta, Christos Dimitrakakis, Bei Liang, and Aikaterini Mitrokotsa. A Differentially Private Encryption Scheme. In *Information Security*, 2017.

[BEB13] Robert G Brown, Dirk Eddelbuettel, and David Bauer. Dieharder: A random number test suite. *Open Source Softw. Libr.*, 2013.

[Bei11] Amos Beimel. Secret-Sharing Schemes: A Survey. In *Coding and Cryptology*, 2011.

[Ben87] Josh Cohen Benaloh. Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret (Extended Abstract). In *Advances in Cryptology — CRYPTO' 86*, 1987.

[BF14] Mihir Bellare and Georg Fuchsbauer. Policy-Based Signatures. In *Public-Key Cryptography – PKC 2014*, 2014.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional Signatures and Pseudorandom Functions. In *Public-Key Cryptography – PKC 2014*, 2014.

[BGI17] Elette Boyle, Niv Gilboa, and Yuval Ishai. Group-Based Secure Computation: Optimizing Rounds, Communication, and Computation. In *Advances in Cryptology – EUROCRYPT 2017*, 2017.

[BGJ+16] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-Lock Puzzles from Randomized Encodings. In *ITCS*, 2016.

[BGM16] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies Without Proof of Work. In *FC*, 2016.

[BIK+17] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, October 2017.

[BK15] Elaine B. Barker and John M. Kelsey. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Technical Report NIST SP 800-90Ar1, National Institute of Standards and Technology, June 2015.

[BK16] Elaine Barker and John Kelsey. Recommendation for Random Bit Generator (RBG) Constructions. Technical Report NIST Special Publication (SP) 800-90C (Draft), National Institute of Standards and Technology, April 2016.

[BKLP15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient Zero-Knowledge Proofs for Commitments from Learning with Errors over Rings. In *Proceedings, Part I, of the 20th European Symposium on Computer Security – ESORICS 2015 - Volume 9326*, 2015.

[BLM18] Carlo Brunetta, Bei Liang, and Aikaterini Mitrokotsa. Lattice-Based Simulatable VRFs: Challenges and Future Directions. *J. Internet Serv. Inf. Secur. JISIS*, 8(4), November 2018.

[BLM19] Carlo Brunetta, Bei Liang, and Aikaterini Mitrokotsa. Code-Based Zero Knowledge PRF Arguments. In *Information Security*, 2019.

[BLMR13] Dan Boneh, Kevin Lewi, Hart Montgomery, and Ananth Raghunathan. Key Homomorphic PRFs and Their Applications. In *Advances in Cryptology – CRYPTO 2013*, 2013.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. *J. Cryptol.*, 17(4), September 2004.

[BMS16] Michael Backes, Sebastian Meiser, and Dominique Schröder. Delegatable Functional Signatures. In *Public-Key Cryptography – PKC 2016*, 2016.

[BMv78] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (Corresp.). *IEEE Trans. Inform. Theory*, 24(3), May 1978.

[BN00] Dan Boneh and Moni Naor. Timed Commitments. In *CRYPTO*, 2000.

[BNO11] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed Private Data Analysis: On Simultaneously Solving How and What. *ArXiv11032626 Cs*, March 2011.

[BP21] Carlo Brunetta and Pablo Picazo-Sanchez. Modelling cryptographic distinguishers using machine learning. *J. Cryptogr. Eng.*, July 2021.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom Functions and Lattices. In *Advances in Cryptology – EUROCRYPT 2012*, volume 7237. 2012.

[Bri90] Ernest F. Brickell. Some Ideal Secret Sharing Schemes. In *Advances in Cryptology — EUROCRYPT '89*, 1990.

[BRS+10] Lawrence Bassham, Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, N. Heckert, and James Dray. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Technical Report NIST Special Publication (SP) 800-22 Rev. 1a, National Institute of Standards and Technology, April 2010.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptology*, 4(1), January 1991.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2), 2014.

[CG86] Alfredo Capelli and Giovanni Garbieri. *Corso Di Analisi Algebrica: 1: Teorie Introduttorie*, volume 1. 1886.

[CGG07] Pierre-Louis Cayrel, Philippe Gaborit, and Marc Girault. Identity-Based Identification and Signature Schemes Using Correcting Codes. In *WCC*, volume 2007, 2007.

[CGMA85] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, October 1985.

[Cha95] Florent Chabaud. On the security of some cryptosystems based on error-correcting codes. In *Advances in Cryptology — EUROCRYPT'94*, 1995.

[CKP+20] S. Cohney, A. Kwong, S. Paz, D. Genkin, N. Heninger, E. Ronen, and Y. Yarom. Pseudorandom black swans: Cache attacks on CTR_DRBG. In *S&P*, May 2020.

[CL07] Melissa Chase and Anna Lysyanskaya. Simulatable VRFs with Applications to Multi-theorem NIZK. In *Advances in Cryptology - CRYPTO 2007*, August 2007.

[CLZ12] Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. A Non-interactive Range Proof with Constant Communication. In *Financial Cryptography and Data Security*, 2012.

[Con18] A. Connolly. Freedom of Encryption. *IEEE Secur. Priv.*, 16(1), January 2018.

[Cou16] Council of the European Union, European Parliament. *Regulation (EU) 2016/679 (General Data Protection Regulation)*. 2016.

[CPSV16] Michele Ciampi, Giuseppe Persiano, Luisa Siniscalchi, and Ivan Visconti. A Transform for NIZK Almost as Efficient and General as the Fiat-Shamir Transform Without Programmable Random Oracles. In *Theory of Cryptography*, 2016.

[CRRV17] Ran Canetti, Srinivasan Raghuraman, Silas Richelson, and Vinod Vaikuntanathan. Chosen-Ciphertext Secure Fully Homomorphic Encryption. In *Public-Key Cryptography – PKC 2017*, 2017.

[CV07] Dario Catalano and Ivan Visconti. Hybrid Commitments and Their Applications to Zero-knowledge Proof Systems. *Theor Comput Sci*, 374(1-3), April 2007.

[CVEYA11] Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem. In *Selected Areas in Cryptography*, 2011.

[CZD+19] Chengjun Cai, Yifeng Zheng, Yuefeng Du, Zhan Qin, and Cong Wang. Towards Private, Robust, and Verifiable Crowdsensing Systems via Public Blockchains. *IEEE Trans. Dependable and Secure Comput.*, 2019.

[DGKR18] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In *Advances in Cryptology – EUROCRYPT 2018*, 2018.

[DH76] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Trans. Inf. Theory*, 22(6), 1976.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, volume 3876. 2006.

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing Encryption Schemes from Decryption Errors. In *Advances in Cryptology - EUROCRYPT 2004*, 2004.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6), November 2004.

[DS06] A.D. Dileep and C.C. Sekhar. Identification of Block Ciphers using Support Vector Machines. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, July 2006.

[Dwo06] Cynthia Dwork. Differential Privacy. In *Automata, Languages and Programming*, 2006.

[EED08] Khaled El Emam and Fida Kamal Dankar. Protecting Privacy Using k-Anonymity. *J Am Med Inf. Assoc*, 15(5), 2008.

[ElG85] Taher ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Advances in Cryptology*, 1985.

[ELL+15] Martianus Frederic Ezerman, Hyung Tae Lee, San Ling, Khoa Nguyen, and Huaxiong Wang. A Provably Secure Group Signature Scheme from Code-Based Assumptions. In *Advances in Cryptology – ASIACRYPT 2015*, 2015.

[ETLP13] Z. Erkin, J. R. Troncoso-pastoriza, R. L. Lagendijk, and F. Perez-Gonzalez. Privacy-preserving data aggregation in smart metering systems: An overview. *IEEE Signal Process. Mag.*, 30(2), March 2013.

[EYACM11] Sidi Mohamed El Yousfi Alaoui, Pierre-Louis Cayrel, and Meziani Mohammed. Improved Identity-Based Identification and Signature Schemes Using Quasi-Dyadic Goppa Codes. In *Information Security and Assurance*, 2011.

[FFKB17] Andreas Fischer, Benny Fuhry, Florian Kerschbaum, and Eric Bodden. Computation on Encrypted Data using Data Flow Authentication. *CoRR*, abs/1710.00390, 2017.

[FG12] Dario Fiore and Rosario Gennaro. Publicly Verifiable Delegation of Large Polynomials and Matrix Computations, with Applications. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 2012.

[FGJS17] Nelly Fazio, Rosario Gennaro, Tahereh Jafarikhah, and William E. Skeith. Homomorphic Secret Sharing from Paillier Encryption. In *Provable Security*, 2017.

[FGP14] Dario Fiore, Rosario Gennaro, and Valerio Pastro. Efficiently Verifiable Computation on Encrypted Data. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[Fis18] Tilo Fischer. Testing Cryptographically Secure Pseudo Random Number Generators with Artificial Neural Networks. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, August 2018.

[FMNP16] Dario Fiore, Aikaterini Mitrokotsa, Luca Nizzardo, and Elena Pagnin. Multi-key Homomorphic Authenticators. In *Advances in Cryptology – ASIACRYPT 2016*, 2016.

[FS87] Amos Fiat and Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology — CRYPTO' 86*, 1987.

[FS96] Jean-Bernard Fischer and Jacques Stern. An Efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding. In *Advances in Cryptology — EUROCRYPT '96*, 1996.

[GAC18] Francisco-Javier González-Serrano, Adrián Amor-Martín, and Jorge Casamayón-Antón. Supervised machine learning using encrypted training data. *Int. J. Inf. Secur.*, 17(4), 2018.

[Gen09] Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD Thesis, Stanford University, 2009.

[GGG17] Zahra Ghodsi, Tianyu Gu, and Siddharth Garg. SafetyNets: Verifiable execution of deep neural networks on an untrusted cloud. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to Construct Random Functions. *J ACM*, 33(4), August 1986.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. In *Advances in Cryptology – CRYPTO 2010*, 2010.

[Gil52] E. N. Gilbert. A comparison of signalling alphabets. *Bell Syst. Tech. J.*, 31(3), May 1952.

[GJ11] Flavio D. Garcia and Bart Jacobs. Privacy-Friendly Energy-Metering via Homomorphic Encryption. In *Security and Trust Management*, 2011.

[GK06] S. Dov Gordon and Jonathan Katz. Rational Secret Sharing, Revisited. In *Security and Cryptography for Networks*, 2006.

[GKL15] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. In *EUROCRYPT*, volume 9057. 2015.

[GKM11] Johannes Gehrke, Daniel Kifer, and Ashwin Machanavajjhala. L-Diversity. In *Encyclopedia of Cryptography and Security*. 2011.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, February 1989.

[GLS07] P. Gaborit, C. Lauradoux, and N. Sendrier. SYND: A Fast Code-Based Stream Cipher with a Security Reduction. In *2007 IEEE International Symposium on Information Theory*, June 2007.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, May 1982.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2), April 1988.

[GNP+15] Sharon Goldberg, Moni Naor, Dimitrios Papadopoulos, Leonid Reyzin, Sachin Vasant, and Asaf Ziv. NSEC5: Provably Preventing DNSSEC Zone Enumeration. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, 2015.

[Goh19] Aron Gohr. Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning. In *Advances in Cryptology – CRYPTO 2019*, 2019.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, June 2011.

[GW13] Rosario Gennaro and Daniel Wichs. Fully Homomorphic Message Authenticators. In *Advances in Cryptology - ASIACRYPT 2013*, 2013.

[HAP17] 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*, October 2017.

[Her28] Alex Hern. Fitness tracking app Strava gives away location of secret US army bases. *The Guardian*, 2018.Jan.28.

[HGDM+11] Gabriel Hospodar, Benedikt Gierlichs, Elke De Mulder, Ingrid Verbauwhede, and Joos Vandewalle. Machine learning in side-channel analysis: A first study. *J Cryptogr Eng*, 1(4), October 2011.

[Hir09] Shoichi Hirose. Security Analysis of DRBG Using HMAC in NIST SP 800-90. In *Information Security Applications*, 2009.

[HMT13] Rong Hu, Kirill Morozov, and Tsuyoshi Takagi. Proof of plaintext knowledge for code-based public-key encryption revisited. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, May 2013.

[HPS14] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. An Introduction to Cryptography. In *An Introduction to Mathematical Cryptography*. 2014.

[HZ19] Xinyi Hu and Yaqun Zhao. Block Ciphers Classification Based on Random Forest. *J. Phys.: Conf. Ser.*, 1168, February 2019.

[JLE14] Zhanglong Ji, Zachary C. Lipton, and Charles Elkan. Differential Privacy and Machine Learning: A Survey and Review. *ArXiv14127584 Cs*, December 2014.

[Jou09] Antoine Joux. *Algorithmic Cryptanalysis*. 2009.

[KK06] Shri Kant and Shehroz S. Khan. Analyzing a class of pseudo-random bit generator through inductive machine learning paradigm. *Intell. Data Anal.*, 10(6), December 2006.

[KKG+09] Shri Kant, Naveen Kumar, Sanchit Gupta, Amit Singhal, and Rachit Dhasmana. Impact of machine learning algorithms on analysis of stream ciphers. In *2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS)*, December 2009.

[KL08] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. 2008.

[KLP07] Yael Tauman Kalai, Yehuda Lindell, and Manoj Prabhakaran. Concurrent Composition of Secure Protocols in the Timing Model. *J Crypto*, 20(4), October 2007.

[KMS14] Jonathan Katz, Andrew Miller, and Elaine Shi. Pseudonymous Broadcast and Secure Computation from Cryptographic Puzzles. Technical Report 857, 2014.

[KMTZ13] Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Universally Composable Synchronous Computation. In *TCC*, 2013.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Math. Comput.*, 48(177), January 1987.

[Koz91] John Koza. Evolving a computer program to generate random numbers using the genetic programming paradigm. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991.

[Kra94] Hugo Krawczyk. Secret Sharing Made Short. In *Advances in Cryptology — CRYPTO' 93*, 1994.

[KRDO17] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In *CRYPTO*, volume 10401, 2017.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In *Advances in Cryptology - ASIACRYPT 2008*, 2008.

[LABK17] Wenting Li, Sébastien Andreina, Jens-Matthias Bohli, and Ghassan Karame. Securing Proof-of-Stake Blockchain Protocols. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, volume 10436. 2017.

[Lin15] Yehuda Lindell. An Efficient Transform from Sigma Protocols to NIZK with a CRS and Non-programmable Random Oracle. In *Theory of Cryptography*, 2015.

[LLM+16] Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Zero-Knowledge Arguments for Matrix-Vector Relations and Lattice-Based Group Encryption. In *Advances in Cryptology – ASIACRYPT 2016*, 2016.

[LLNW17] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-Knowledge Arguments for Lattice-Based PRFs and Applications to E-Cash. In *Advances in Cryptology – ASIACRYPT 2017*, 2017.

[LLNW18] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-Based Zero-Knowledge Arguments for Integer Relations. In *Advances in Cryptology – CRYPTO 2018*, 2018.

[LLV07] N. Li, T. Li, and S. Venkatasubramanian. T-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, April 2007.

[LMA+18] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS*, 2018.

[LMS18] Russell W. F. Lai, Giulio Malavolta, and Dominique Schröder. Homomorphic Secret Sharing for Low Degree Polynomials. In *Advances in Cryptology – ASIACRYPT 2018*, 2018.

[LS07] Pierre L'Ecuyer and Richard Simard. TestU01: A C library for empirical testing of random number generators. *ACM Trans. Math. Softw.*, 33(4), August 2007.

[LW15] Arjen K. Lenstra and Benjamin Wesolowski. A random zoo: Sloth, unicorn, and trx. Technical Report 366, 2015.

[LYAX18] Kang Li, Rupeng Yang, Man Ho Au, and Qiuliang Xu. Practical Range Proof for Cryptocurrency Monero with Provable Security. In *Information and Communications Security*, 2018.

[MCEYA11] Mohammed Meziani, Pierre-Louis Cayrel, and Sidi Mohamed El Yousfi Alaoui. 2SC: An Efficient Code-Based Stream Cipher. In *Information Security and Assurance*, 2011.

[Mei12] Rebecca Meissen. *A Mathematical Approach to Fully Homomorphic Encryption*. PhD Thesis, Worcester Polytechnic Institute, 2012.

[MHC12] Mohammed Meziani, Gerhard Hoffmann, and Pierre-Louis Cayrel. Improving the Performance of the SYND Stream Cipher. In *Progress in Cryptology - AFRICACRYPT 2012*, 2012.

[MMR+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. of AISTATS*, 2017.

[MMV11] Mohammad Mahmoody, Tal Moran, and Salil Vadhan. Time-Lock Puzzles in the Random Oracle Model. In *Advances in Cryptology – CRYPTO 2011*, 2011.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with Small Parameters. In *Advances in Cryptology – CRYPTO 2013*, 2013.

[MR02] Silvio Micali and Ronald L. Rivest. Micropayments Revisited. In *Topics in Cryptology — CT-RSA 2002*, 2002.

[MT09] Ravi Montenegro and Prasad Tetali. How long does it take to catch a wild kangaroo? In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, May 2009.

[MT19] Giulio Malavolta and Sri Aravinda Krishnan Thyagarajan. Homomorphic Time-Lock Puzzles and Applications. In *CRYPTO*, 2019.

[MVR99] Silvio Micali, Salil Vadhan, and Michael Rabin. Verifiable Random Functions. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, October 1999.

[Nie02] Jesper Buus Nielsen. Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case. In *Advances in Cryptology — CRYPTO 2002*, volume 2442. 2002.

[NIS17] NIST STS. *Cryptographic Key Length Recommendation*. 2017.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (Sp 2008)*, May 2008.

[PAH+18] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Trans. Inf. Forensics Secur.*, 13(5), May 2018.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology — EUROCRYPT '99*, 1999.

[Par18] Stuart L Pardau. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J Tech Pol*, 23, 2018.

[PB10] K. Peng and F. Bao. An Efficient Range Proof Scheme. In *2010 IEEE Second International Conference on Social Computing*, August 2010.

[PBP18] Elena Pagnin, Carlo Brunetta, and Pablo Picazo-Sanchez. HIKE: Walking the Privacy Trail. In *Cryptology and Network Security*, 2018.

[Pei16] Chris Peikert. A Decade of Lattice Cryptography. *Found. Trends Theor. Comput. Sci.*, 10(4), March 2016.

[PJ17] M. Panjwani and M. Jäntti. Data Protection Security Challenges in Digital IT Services: A Case Study. In *2017 International Conference on Computer and Applications (ICCA)*, September 2017.

[PO14] Alberto Peinado and Andrés Ortiz. Prediction of Sequences Generated by LFSR Using Back Propagation MLP. In *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14*, 2014.

[Pol78] John M Pollard. Monte Carlo methods for index computation (mod p). *Math. Comput.*, 32(143), 1978.

[Pol00] J. M. Pollard. Kangaroos, Monopoly and Discrete Logarithms. *J. Cryptol.*, 13(4), September 2000.

[PRV12] Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. How to Delegate and Verify in Public: Verifiable Computation from Attribute-Based Encryption. In *Theory of Cryptography*, 2012.

[PST13] Charalampos Papamanthou, Elaine Shi, and Roberto Tamassia. Signatures of Correct Computation. In *Theory of Cryptography*, 2013.

[PWH+17] Dimitrios Papadopoulos, Duane Wessels, Shumon Huque, Moni Naor, Jan Včelák, Leonid Reyzin, and Sharon Goldberg. *Making NSEC5 Practical for DNSSEC*. 2017.

[RAD78] R L Rivest, L Adleman, and M L Dertouzos. On Data Banks and Privacy Homomorphisms. *Found. Secure Comput. Acad. Press*, 1978.

[Reg10] Oded Regev. The Learning with Errors Problem (Invited Survey). In *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity*, 2010.

[RSW96] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock Puzzles and Timed-release Crypto. Technical report, Massachusetts Institute of Technology, 1996.

[SAL07] Dario L. M. Sacchi, Franca Agnoli, and Elizabeth F. Loftus. Changing history: Doctored photographs affect memory for past public events. *Appl. Cogn. Psychol.*, 21(8), December 2007.

[Sha48] C. E. Shannon. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, 27(3), 1948.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11), November 1979.

[SS15] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[SSM14] Bas Stottelaar, Jeroen Senden, and Lorena Montoya. Online social sports networks as crime facilitators. *Crime Sci*, 3(1), August 2014.

[SSV19] Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Publicly Verifiable Proofs from Blockchains. In *PKC*, 2019.

[ST96] Moshe Sipper and Marco Tomassini. Generating Parallel Random Number Generators by Cellular Programming. *Int. J. Mod. Phys. C*, 07(02), April 1996.

[ST13] W. A. R. D. Souza and A. Tomlinson. A Distinguishing Attack with a Neural Network. In *2013 IEEE 13th International Conference on Data Mining Workshops*, December 2013.

[Sta96] Markus Stadler. Publicly Verifiable Secret Sharing. In *Advances in Cryptology — EUROCRYPT '96*, 1996.

[STBK+18] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry McKay, Mary Baish, and Michael Boyle. Recommendation for the Entropy Sources Used for Random Bit Generation. Technical Report NIST Special Publication (SP) 800-90B, National Institute of Standards and Technology, January 2018.

[Ste89] Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications*, 1989.

[Ste96] J. Stern. A new paradigm for public key identification. *IEEE Trans. Inf. Theory*, 42(6), November 1996.

[Str18] Strava. Strava. https://www.strava.com, November 2018.

[SUM13] Petr Svenda, Martin Ukrop, and Vashek Matyáš. Towards cryptographic function distinguishers with evolutionary circuits. In *2013 International Conference on Security and Cryptography (SECRYPT)*, July 2013.

[SV10] N. P. Smart and F. Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In *Public Key Cryptography – PKC 2010*, 2010.

[TB19] Florian Tramèr and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *Proceedings of ICLR*, 2019.

[THH+09] Brian Thompson, Stuart Haber, William G. Horne, Tomas Sander, and Danfeng Yao. Privacy-Preserving Computation and Verification of Aggregate Queries on Outsourced Databases. In *Privacy Enhancing Technologies*, volume 5672. 2009.

[TLM18] Georgia Tsaloli, Bei Liang, and Aikaterini Mitrokotsa. Verifiable Homomorphic Secret Sharing. In *Provable Security (ProvSec), 2018*, volume 11192, 2018.

[TM20] Georgia Tsaloli and Aikaterini Mitrokotsa. Sum it up: Verifiable additive homomorphic secret sharing. In *Information Security and Cryptology – ICISC 2019*, 2020.

[Var57] R. R. Varshamov. Estimate of the Number of Signals in Error Correcting Codes. *Docklady Akad Nauk SSSR*, 117, 1957.

[vGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully Homomorphic Encryption over the Integers. In *Advances in Cryptology – EUROCRYPT 2010*, 2010.

[Wes19] Benjamin Wesolowski. Efficient Verifiable Delay Functions. In *EUROCRYPT*, 2019.

[WF02] Ian H. Witten and Eibe Frank. Data mining: Practical machine learning tools and techniques with Java implementations. *SIGMOD Rec.*, 31(1), March 2002.

[WS19] Joanne Woodage and Dan Shumow. An Analysis of NIST SP 800-90A. In *Advances in Cryptology – EUROCRYPT 2019*, 2019.

[XEQ18] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*, 2018.

[XLL+20] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin. VerifyNet: Secure and Verifiable Federated Learning. *IEEE Trans. Inf. Forensics Secur.*, 15, 2020.

[YS16] Yu Yu and John Steinberger. Pseudorandom Functions in Almost Constant Depth from Low-Noise LPN. In *Advances in Cryptology – EUROCRYPT 2016*, volume 9666. 2016.

[ZZL18] Zhicheng Zhao, Yaqun Zhao, and Fengmei Liu. The Research of Cryptosystem Recognition Based on Randomness Test's Return Value. In *Cloud Computing and Security*, 2018.