NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

TIAGO ALEXANDRE PORTUGAL MORAIS

Bachelor in Computer Science and Engineering

CONSTRUCTING TRUSTED INPUTS FOR SECURE MULTI-PARTY COMPUTATION PROTOCOLS

EMBEDDING ZERO KNOWLEDGE PROOFS FOR TRUSTWORTHY AND CONFIDENTIAL INPUT GATHERING

Dissertation Plan
MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
July, 2023

DEPARTMENT OF
COMPUTER SCIENCE

CONSTRUCTING TRUSTED INPUTS FOR SECURE MULTI-PARTY COMPUTATION PROTOCOLS

EMBEDDING ZERO KNOWLEDGE PROOFS FOR TRUSTWORTHY AND CONFIDENTIAL INPUT GATHERING

TIAGO ALEXANDRE PORTUGAL MORAIS
Bachelor in Computer Science and Engineering

Adviser: Alexander Andrew Davidson
Assistant Professor, NOVA University Lisbon

Dissertation Plan
MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
July, 2023

Abstract

Throughout history, human collaboration has enabled us, as a species, to survive hardships, learn, and make better decisions. For example, nations have been shaped by the electorate vote, the process in which leaders are chosen to represent their citizens. These decision-making collaborations can be thought of as computations, which evolve with technology, allowing greater collaboration opportunities. In these computations, the inputs remain private while the overall result is public. This kind of computation is called Secure Multi-Party Computation.

One problem that exists with this computation, however, results from trust assumptions, in which malicious individuals, through their private input, provide disruptive inputs that skew the final result, affecting decision-making and everyone involved. While some solutions have been adopted to address this problem, overall they remain inefficient.

This dissertation will research how to solve this problem efficiently using Zero Knowledge Proofs that guarantee input legitimacy while keeping them private. We'll develop and implement a solution on an existing protocol named STAR, which will address this problem in user-provided individual information, available in Signed Identity Documents, which originate from a trusted source of information, such as the government.

This solution is of interest in today's widely available Internet services that require user authentication for content that is age-restricted, limited to citizens of a certain country, or that require absolute identity authentication. Further research might allow for general-purpose proofs that can be used for any kind of input, facilitating integration with existing protocols.

Keywords: Secure Multi-Party Computation, Zero Knowledge Proof, Privacy, Trusted Inputs

Resumo

Ao longo da história, a colaboração humana permitiu-nos, como espécie, sobreviver a dificuldades, aprender, e fazer melhores decisões. Por exemplo, nações foram moldadas pelo voto eleitoral, o processo em que líderes são escolhidos para representar os seus cidadãos. Estas colaborações de decisão podem ser interpretadas como computações, que evoluem com a tecnologia, permitindo um maior leque de oportunidades colaborativas. Nestas computações, os *inputs* permanecem privados enquanto que o resultado final é público. Este tipo de computação denomina-se Computação Multipartidária Segura.

Um problema existente com esta computação, contudo, resulta de suposições de confiança, nas quais indivíduos maliciosos, através dos seus *inputs* privados, fornecem *inputs* disruptivos que enviesam o resultado final, afetando decisões e todos os envolvidos. Enquanto que algumas soluções têm sido adotadas para resolver este problema, no geral permanecem ineficientes.

Esta dissertação irá investigar como resolver este problema de forma eficiente utilizando Provas de Zero Conhecimento que garantem legitimidade de *inputs* preservando a sua privacidade. Iremos desenvolver e implementar uma solução num protocolo existente intitulado STAR que irá encarar este problema, mais concretamente, em informação individual fornecida por utilizadores, disponível em documentos de identificação assinados, que originam de fontes de informação confiáveis, como o governo.

Esta solução é de todo o interesse nos dias de hoje em serviços de Internet amplamente disponíveis que exigem autenticação dos seus utilizadores de forma a poderem aceder a conteúdo que é restrito por idade, limitado a cidadãos de determinado país de origem, ou que exigem autenticação de identidade absoluta. Investigação aprofundada poderá permitir a utilização de provas de uso geral que podem ser utilizadas para qualquer tipo de *input*, facilitando a integração com protocolos existentes.

Palavras-chave: Computação Multipartidária Segura, Prova de Zero Conhecimento, Privacidade, *Inputs* Confiáveis

Contents

List of Figures

Acronyms

CA	Certifying Authorities *(pp. 8, 9, 13–17, 24)*
DID	Decentralized Identifiers *(p. 13)*
FCT	NOVA School of Science and Technology *(p. 21)*
NP	Nondeterministic Polynomial *(p. 10)*
PKI	Public-Key Infrastructure *(p. 8)*
QSCD	Qualified Signature Seal Creation Device *(pp. 9, 15)*
SCEE	System of Electronic Certification of the Portuguese State *(p. 9)*
SFE	Secure Function Evaluation *(p. 5)*
SMPC	Secure Multi-Party Computation *(pp. 1–3, 5, 6, 12–15, 21, 24)*
STAR	Secret Sharing for Private Threshold Aggregation Reporting *(pp. 2, 3, 7, 8, 15, 16, 19–22, 25)*
W3C	World Wide Web Consortium *(p. 13)*
zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge *(pp. 3, 10, 11, 16)*
zk-STARK	Zero-Knowledge Scalable Transparent Argument of Knowledge *(pp. 3, 10, 11, 16)*
ZKP	Zero Knowledge Proof(s) *(pp. 2, 3, 9, 10, 12–16, 18, 21, 22, 24)*

1

Introduction

DRAFT

1.1 Motivation

Collaboration is an essential and defining characteristic of humans as a whole, which has enabled our species to survive throughout history. Its purpose isn't limited to facing hardships, but also to learn and collect information that can allow for more thoughtful decisions regarding those involved in it. That learning, collection, and eventual decision, can be thought of as a computation.

Take the democratic vote as an example: Individuals of a nation make a single vote, which amassed allows them to elect a leader together. The process of collecting, counting, and deciding on that leader, can be referred to as the computation.

As technology evolved so did collaboration. With the development of computers and networks, more opportunities arose for collaboration that benefits from these technologies. Individuals can collaborate with one another over networks to compute through their machines some data that interests them alike.

In some cases of collaboration, individuals might have a need for computing some data collaboratively while keeping their inputs hidden. This type of collaboration uses a form of computation called Secure Multi-Party Computation (SMPC). Inputs provided by individuals remain hidden throughout the process of the computation and thus allow them to learn the result of the computation with guarantees of their inputs being hidden [16].

1.2 Problem Definition

One of the problems with this collaboration results from trust assumptions, however, where malicious individuals might provide invalid or disruptive inputs capable of manipulating or skewing the collaborative final result of the computation.

Real-world scenarios where this behavior might be exploited not only cause harm to the honest individuals of the computation, that might depend on that result for their own purposes but also third-party users that benefit from the honest result of the computation and are subsequently affected by wrong results.

Such examples where individuals can be affected directly and indirectly are sustainability metrics [26], in which gathering of fossil fuel emissions from a group of organizations can be manipulated to give an estimate of emissions that fails to accurately disclose its true magnitude to the wider population. The nature of the inputs provided by the organizations, which are associated with intellectual property, requires that they remain private, thus directly validating them through trusted parties becomes very invasive.

As outlined by [1], some proposed solutions implement protective measures such as digital signatures, encryption, and authorization. A known reputation table of participants might also be used for a participant to decide if it wants to engage with another. Another example is data use agreements that identify the purpose of the data, and have penalties enforced if the agreement is violated.

1.3 Solution

This dissertation will research how to solve these problems in an efficient way, however, through the use of Zero Knowledge Proof(s) (ZKP). These will be used as additional inputs to the SMPC protocols to ensure validity in the inputs given while maintaining their value, individual information, and related information hidden, other than the fact that they are legitimate.

To this effect, we will first specifically develop and implement a ZKP protocol to validate user-provided individual information, by guaranteeing that their input is valid according to their personal Signed Identity Document, without exposing it or any other private information. It's important that this information is valid according to such document, as it originates from a trusted source of information, in this case, the government, and can provide authenticity mechanisms to be used in the proof. This is an interesting case as nowadays many services on the web require that users authenticate themselves to prove they are above a certain age, from a certain country, or that they are who they claim to be.

The protocol will then be integrated with the Secret Sharing for Private Threshold Aggregation Reporting (STAR) protocol [14], a simple to adopt Private Threshold Aggregation System capable of solving the Private Heavy-Hitters Problem developed and used by Brave, which provides high efficiency and low cost. There's the additional factor that, given prior knowledge on STAR, we view it as a fit candidate for our solution. Besides, this dissertation will be done in collaboration with one of the authors and developers of STAR itself, Assistant Professor Alexander Davidson.

Further research may be done to allow for a general purpose ZKP protocol that can build and validate proofs for any type of input, thus facilitating integration with existing message schemes or integration for other types of SMPC.

1.4 Overview of Techniques

Our solution will make use of non-interactive ZKP such as Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) and Zero-Knowledge Scalable Transparent Argument of Knowledge (zk-STARK) by leveraging the Cartão de Cidadão's data, certificates, and signing key, all of which serve as important data points. These proofs will be generated by the client and presented along with their message as they begin the STAR protocol. At some point during the protocol, prior to the aggregation, our solution will verify the proof and decide whether it moves forward to the aggregation phase or is rejected altogether. Thus we ensure that, from the k measurements that are aggregated, all are legitimate and can be revealed.

1.5 Contributions

The following contributions are expected from the development of this dissertation:

1. **Contributing to the development of trust guarantees in SMPC protocols** - By addressing the problem with trust in Secure Multi-Party Computation and leveraging Zero Knowledge Proof properties, we wish to make contributions to the development of SMPC, building on existing properties, such that inputs can be trusted while remaining private.

2. **Development of a Zero Knowledge Proof of identity** - Development of a Zero Knowledge Proof capable of validating an individual's personal information embedded in smart cards, whilst keeping it hidden.

3. **Integration of STAR with smart card Zero Knowledge Proof validation** - Integration of STAR with a Zero Knowledge Proof protocol capable of validating id card information. There is also the prospect of developing a general-purpose protocol capable of validating other sources of inputs.

1.6 Document Structure

The document is structured in the following manner, divided into chapters:

- **Introduction** - Describes the context and motivation for the thesis along with a brief description of the proposed solution and what contributions are expected from it.

- **Background** - Establishes and provides a base understanding of the technologies and areas of work related to this dissertation, such as SMPC, STAR, Signed Identity Documents and ZKP.

- **Related Work** - Showcases work that has been done and is related to the dissertation, in order to support its solution proposal and establish the notoriety of the context.

- **Proposed Solution** - Goes into detail over the proposed solution, a description of its requirements, decisions made and system architecture along with other details.
- **Preliminary Work** - Explains the preliminary work done in preparation for the dissertation, such as acquaintance with the subject at hand and discussion regarding the proposal.

2

Background

This chapter provides knowledge about the concepts addressed in this dissertation that are essential to its development. It is divided into 4 sub-sections:

1. The first describes what Secure Multi-Party Computation is (in addition to what issues are still present);

2. The second describes STAR, the protocol that will be built on to implement the solution proposed by the dissertation;

3. The third describes what Signed Identity Documents are and how they achieve authenticity;

4. Lastly, the fourth describes what Zero Knowledge Proofs are and how they can contribute to the solution.

2.1 Secure Multi-Party Computation

First introduced as Two-Party Computation by Yao [43] through the *Millionaires Problem*, SMPC (or Secure Function Evaluation (SFE)) is defined by Goldreich [19] as the arbitrary function of m arguments in which m interested parties willingly provide their corresponding input in order to obtain the result of the function, whilst preventing other parties from gathering any information besides the result of said function, as figure 2.1 illustrates.

Instead of depending on a trusted third party that could receive the parties' inputs and compute the function for them, these parties attempt to achieve the same goal by communicating with one another instead. In what is called the *Real-World/Ideal-World* paradigm, protocols that implement SMPC try to establish some security conditions that are met in the *Real-World* model the same way they are met in the *Ideal-World* model, which usually uses the trusted third-party approach. This way, parties communicating with one another through the protocol are as secure as if they were using a trusted third-party [2].

SMPC protocols must satisfy two conditions:

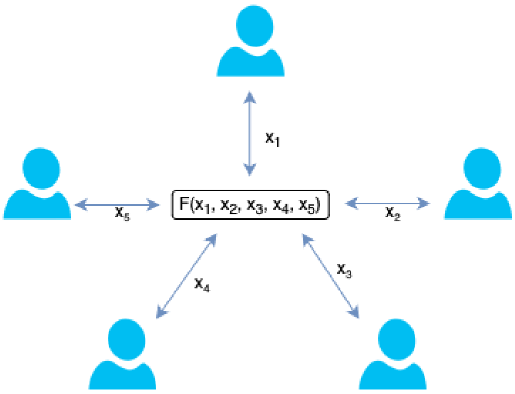


Figure 2.1: Example of SMPC

- **Correctness**: The correct result of the function is computed and captured by the participants;
- **Privacy**: The only new information revealed is the result.

These conditions are ensured through **Formal Security Models** and proofs of security that reveal a protocol's behavior is indistinguishable from one with Ideal Functionalities.

It must be acknowledged that participating parties might not follow the guidelines established by the protocols, and such scenarios are more than plausible to happen in the real world. The parties might manipulate their inputs such that they don't fit in accordance with the protocol, or they could deviate their behavior from what is expected from it. These parties are considered adversaries, parties that may try to corrupt or learn more than intended from a protocol. As Goldreich et al. put it [18], they are faulty machines that can be *passive* or *malicious*.

- **Passive Adversary**: A party that may not want to corrupt the computation but would use third-party methods to learn information about the parties' inputs.
- **Malicious Adversary**: A party that may break the privacy constraint but whose main goal is to change the outcome of the computation.

Many current SMPC protocols have some degree of protection against these adversaries by using multiple techniques such as majority quorums, secret-sharing, oblivious transfer, encryption, signatures, and zero-knowledge proofs [24].

As pointed out by [13] it is assumed that participating parties have an incentive to offer inputs that will result in an outcome they genuinely desire to acquire. The very nature of SMPC is to hide inputs, hence verifying inputs would counter privacy guarantees. Thus, it is difficult to determine if the inputs are reasonable without checking them. You may recognize that this problem is what we will be trying to solve.

It is difficult to resolve this issue no matter how secure a protocol is, but it's important to note that even though the end result might not hold some value we trust in, wrong inputs don't affect the conditions of security of SMPC or its behavior.

Multiple protocols have been developed to satisfy these conditions in each of their planned scenarios. STAR is an example of such a protocol and will be described in the following subsection. Other protocols include BaRK-OPRF for private set intersection and PRIO for computation of aggregate statistics [36].

2.2 STAR

Secret Sharing for Private Threshold Aggregation Reporting, or STAR for short, is a Private Threshold Aggregation System of user measurements, and a solution to the Private Heavy-Hitters Problem developed by Brave focused on high efficiency and low cost, all the while leveraging simple and well-established cryptography for easier adoption from developers scattered across a wide range of expertise [14]. A general depiction of STAR's architecture is given in figure 2.2.

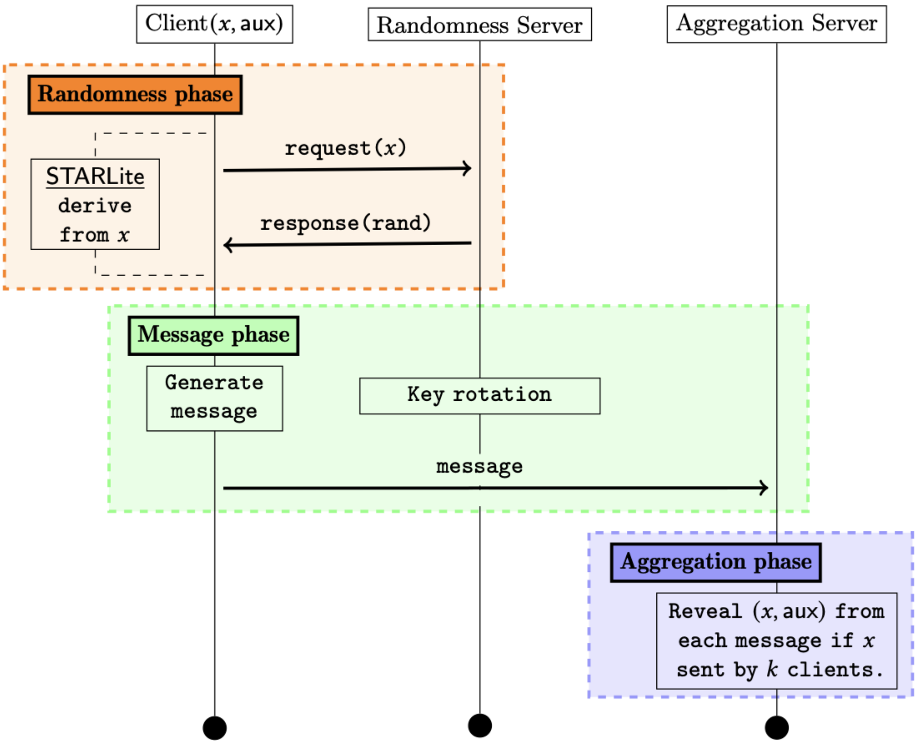


Figure 2.2: STAR architecture [14]

Threshold Aggregation Systems are collaborative computations in which N parties contribute their inputs or computations to a central entity called the *aggregator*, which after reaching a threshold of k inputs, computes the aggregation result. Generally, it is used for data collection, like the collection of client measurements such as cellphone brands of their personal smartphones.

As mentioned earlier, STAR is a solution to the Private Heavy-Hitters Problem, which is characterized by the need of gathering the set of all popular values for a certain parameter above a certain threshold, from a pool of clients, without learning anything else about any specific client's parameters. Another solution that addresses this problem is Poplar [8]. Use cases include, for example, the need to know the most commonly searched word within a search engine without learning which searches a client is making.

STAR ensures client measurements remain hidden until k equal measurements are collected, through compatible secret shares for a (k,N)-secret-sharing scheme. Thus, upon reaching k equal measurements, any correlation between clients and measurements is negligible.

STAR ensures the **correctness** property and further ensures the **privacy** property within the bounds of some leakage value, regarding client measurements that remain hidden, so that practical performance is achievable, a common trade-off used by other threshold aggregation systems. STAR's open source Rust implementation is available through the following GitHub repository.

2.3 Signed Identity Documents

Signed Identity Documents used to identify individuals are comprised of information regarding the individual they are assigned to. Name, date of birth, and a face picture are usually present in such documents, among other important information the document might entail. These documents aren't restricted to a physical format however, technological advancements in the last century have enabled and created a need for digital formats. Thus, it has become the norm for physical documents to accommodate chips capable of holding that individual's information in digital format, commonly known as smart cards.

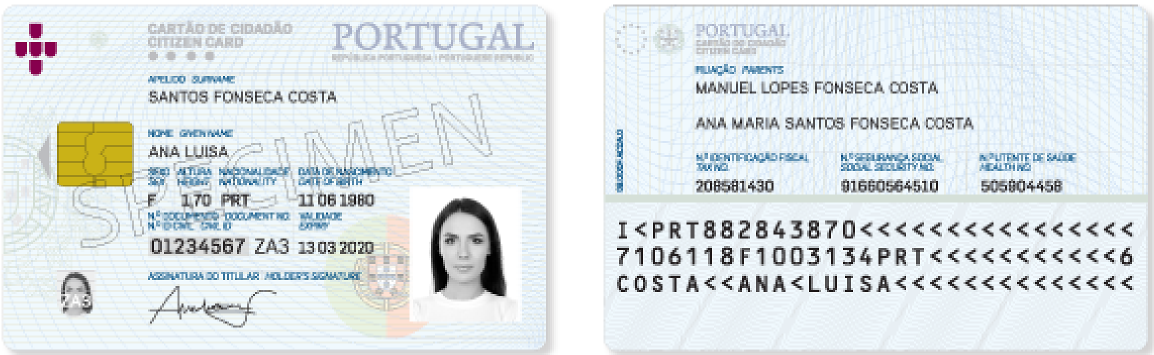
Physical documents, which are susceptible to forgery and consequently identity falsification, employ measures of ensuring they are legitimate. This can be done through a plethora of ways, but ultimately they allow for verification procedures that capture illegitimacies and authenticate the documents. Digital documents are no exception to this risk, and so, measures are also employed to allow the capture of illegitimacies and authentication.

These measures are based on Public-Key Infrastructure (PKI). PKI employs cryptographic and mathematical algorithms which are established as norms to provide private and public key pairs to users. The former is held in secret by the user and used to sign documents, and the latter is publicly available and is used to verify said signature. In combination with trusted third-party entities called Certifying Authorities (CA), the authenticity of such signatures is guaranteed through signed certificates. The individuals are then able to emit digital signatures that behave the same way as traditional signatures or even to a greater extent [25, 39].

2.3.1 Cartão de Cidadão

Cartão de Cidadão is Portugal's identity document assigned to its citizens. In actuality, it's a combination of many documents, such as the ones used for social security and taxes, to name a few.

It displays general information about the individual it belongs to, such as name, date of birth, and face picture. More importantly, it's accompanied by a smart card chip holding this information and more, such as an address, as can be seen in figure 2.3. The smart card holds numerous certificates, of which, and to our interest, including authenticity and qualified digital signature certificates, both provided by a CA named Multicert [40].



CARTÃO DE CIDADÃO
CITIZEN CARD
PORTUGAL
REPÚBLICA PORTUGUESA | PORTUGUESE REPUBLIC

APELIDO SURNAME
SANTOS FONSECA COSTA

NOME GIVEN NAME
ANA LUISA

SEXO SEX | ALTURA HEIGHT | NACIONALIDADE NATIONALITY | DATA DE NASCIMENTO DATE OF BIRTH
F 1,70 PRT 11 06 1980

Nº DOCUMENTO DOCUMENT NO. Nº ID CIVIL CIVIL ID | VALIDADE EXPIRY
01234567 ZA3 13 03 2020

ASSINATURA DO TITULAR HOLDER'S SIGNATURE

SPECIMEN

PORTUGAL
CARTÃO DE CIDADÃO
CITIZEN CARD

FILIAÇÃO PARENTS
MANUEL LOPES FONSECA COSTA

ANA MARIA SANTOS FONSECA COSTA

Nº IDENTIFICAÇÃO FISCAL TAX NO. | Nº SEGURANÇA SOCIAL SOCIAL SECURITY NO. | Nº UTENTE DE SAÚDE HEALTH NO.
208581430 | 91660564510 | 505804458

```
I<PRT882843870<<<<<<<<<<<<<<<<
7106118F1003134PRT<<<<<<<<<<<6
COSTA<<ANA<LUISA<<<<<<<<<<<<<<
```

Figure 2.3: Specimen of Cartão de Cidadão [31]

These certificates are built using X.509 v3, which links a non-repudiation public key (to protect against individuals who falsely deny having performed a certain action) to its holder (in this case, the individual). They present other fields that identify the certificate by a unique serial number, the subject the certificate was assigned to, the issuing CA and its signature, and many more. Additionally, these certificates are built following policy guidelines identified in the certificate itself, which include the System of Electronic Certification of the Portuguese State (SCEE)[30, 23, 7].

The smart card is a Qualified Signature Seal Creation Device (QSCD) [41] which follows strict requirements that ensure with high guarantee the protection of private keys from replication or forgery methods, thus ensuring high legal certainty regarding digital signatures. The certificate of qualified digital signature has a public key associated with a private key held by the QSCD. By using a PIN code known by the individual, QSCD enables this private key to be used to sign digital documents which in turn can be verified by anyone using the public key of that certificate [23].

2.4 Zero Knowledge Proofs

ZKP were initially introduced by Goldwasser, Micali, and Rackoff [22] as probabilistic theorem-proving procedures. They make it possible for a prover to convince a verifier of the validity of a claim without revealing any information besides the statement itself, by providing proofs.

Provers can assert knowledge of a certain property while ensuring that the verifier cannot gain any additional information beyond what it already knows and the proof itself.

The probabilistic nature of these proofs provides the verifier a way to conclude with high certainty that the prover does in fact possess knowledge about said property [21]. As noted by Goldreich et al., one can compare a ZKP to being *"computationally equivalent to an answer of a trusted Oracle"* [20].

These proofs are done over Nondeterministic Polynomial (NP) problems, which as [28] put it, are hard computational problems with no efficient solution algorithm in polynomial time, but whose proposed solutions can be verified efficiently. A solution, commonly known as a witness, to the problem, is known by a prover who doesn't want to disclose it.

As presented by [21], ZKP can be constructed for every NP problem. This is important as it gives credit to how general and widely applicable ZKP are in numerous fields, and not just mathematics or cryptography. For example, by conducting a series of rounds wherein a color-blind individual alternates the positions of two distinct-colored balls hidden behind their back, and consistently providing correct answers to the question "Have I switched the balls?", the prover can convince the color-blind person of their ability to differentiate colors without explicitly revealing the specific colors involved.

Three properties define ZKP within the bounds of a negligible soundness error, which is the probability of a false proof convincing a verifier [27]:

- **Completeness**: A verifier will be convinced when given proof of a true statement;
- **Soundness**: A verifier can't be convinced given proof of a false statement;
- **Zero-Knowledge**: The only information the verifier learns, given proof of a true statement, is that the statement itself is true.

That being said, Zero-Knowledge protocols can either be interactive or non-interactive [6, 15]:

- **Interactive**: The proof has multiple rounds of interaction between the prover and verifier. It's useful for a single verifier;
- **Non-Interactive**: The prover sends a single proof to the verifier. It's useful when there are multiple verifiers;

2.4.1 ZK-Snarks and ZK-Starks

zk-SNARK, first introduced by [5] and zk-STARK, introduced by [3], are a variation of non-interactive ZKP that have been widely adopted, as they are short and easily verifiable. Furthermore, they are capable of being used without communication between prover and verifier, which proves to be great when no communication can be established between them[1] or for multiple verifiers.

zk-SNARK and zk-STARK differ in some ways. While zk-SNARK requires a trusted setup [29], zk-STARK does not. Also, the latter has better scalability even though it has

[1] After a trusted setup has been finalized

larger proof sizes. Verification is faster with zk-SNARK but building proofs proves to be quicker using zk-STARK [11, 38].

The trusted setup is the process in which the prover and the verifier agree on two publicly available keys, a proving key *pk*, and a verification key *vk*. These keys are public parameters that only need to be generated once, by the verifier, using a secret *lambda* for a given program *C*, used for the proof generation and validation. This *lambda* must remain secret, or else a prover can provide proof of a false statement which is validated by the verifier [33, 12].

It is not our focus to highlight the differences between these two approaches, but instead to understand the capability they provide to create non-interactive zero-knowledge proofs, that can be leveraged for scaled applications.

Multiple schemes and software libraries exist that support creation and use of these proofs in various languages and environments, such as arkworks for Rust, DIZK for Java, and ZoKrates, a toolbox for zk-SNARK on Ethereum, to name a few [37].

3

Related Work

In this chapter, the attention is directed toward existing works that possess functionalities akin to the one intended to be developed in this dissertation, or that address the context in which it is part of. A comprehensive overview of several works is provided, and some of them are examined in greater depth. The chapter is divided into 3 sections:

1. The first section shows that there has been considerable study regarding ZKP and identity mechanisms, such as proofs of identity through knowledge and development of schemes for authenticity using various types of documents;

2. The second shows that the need for valid inputs in SMPC has been mentioned and addressed by several studies, arguing for possible use cases that can benefit from this functionality, and hope for improvement in this regard.

3. The third addresses proofs of provenance over TLS, which can be used to assert that some data is provided by a trusted entity without requiring mechanisms such as signatures, allowing for greater accessibility.

3.1 Zero Knowledge Proofs for Anonymous Identity Credentials

Identity credentials are important documents whose main purpose is to identify and authenticate individuals. These can be physical or digital documents and provide entities assurance of who an individual claims to be.

The use of ZKP opens the door to the possibilities of anonymous credentials and the privacy-ensuring properties they bring along with them.

The use of ZKP with identity documents isn't a new proposition, in fact, it has been proposed by Feige et. al [17] in 1987 as a means for parties to prove their identity through knowledge, and like Thomas Beth [4], for authentication protocols in smart cards.

Burmester et al. [9] have also leveraged ZKP to build a cryptographic scheme for passport authentication and secure remote logins, which in their belief, at the time of writing in 1991, could further be enhanced by using smart cards.

More recently, the idea of privacy-preserving identification was also suggested by Rosenberg et al. [34] in response to the increasingly expanding internet services' need for identification, which makes users susceptible to trackers, data exposure risks, and, of course, privacy-invading procedures.

If these mechanisms were to be established as the norm, there would be general-purpose ZKP capable of dealing with the world's complexity regarding identity credentials and specific eligibility protocols.

In addition, through the use of Blockchain technology, decentralization from CA could be achieved, enabling greater control over personal information while ensuring greater security and privacy. In 2022 the World Wide Web Consortium (W3C) officially recommended the use of Decentralized Identifiers (DID), a decentralized solution that provides genuine trust through respectful two-way relationships that safeguard against forgery, prioritize privacy, and elevate usability, supported by many institutions, such as the U.S. Department of Homeland Security [42].

3.2 Valid Inputs in Secure Multi-Party Computation

As mentioned in previous chapters, while SMPC ensures the privacy of inputs, it, unfortunately, suffers from not having validity guarantees over them, which deters mutually suspicious parties from adhering to such protocols.

3.2.1 Validity of sustainability metrics

The need for solutions to this problem has been addressed by McDaniel and Shih [26], with an emphasis on sustainability. McDaniel makes a call for action to reach sustainability goals, by addressing problems with emission metrics shared by big corporations, which may not hold legitimacy. Even though regulators exist, there's the understanding that corporations would want to keep their privacy regarding how that data is collected, thus, some suggestions are made in order to reach the desired goal:

- Verifiable data collection gathered from sensor readings using proofs of construction;
- Sensor disclosure of sustainability metrics that preserve privacy;
- Construction of proofs of computation that reveal compliance to manufacturing processes, whilst not exposing sensitive private information;

Essentially, McDaniel is exposing the need for validation of inputs in SMPC and suggesting solutions that can reach that goal without breaking privacy properties.

3.2.2 Private Set Intersection

Camenisch and Zaverucha [10] also address the issue of input validation, albeit in the context of private set intersection through a solution using certified sets.

Private set intersection is a SMPC technique in which two parties, each holding a private set, compute their intersection while hiding elements not part of the intersection from one another. An example of this technique is the intersection of medical databases, which protects patient information while allowing a better understanding of their illnesses through common symptoms and medical practices.

Through their solution, they protect parties from malicious participants using arbitrary sets. They do this by ensuring inputs are validated by mutually trusted CA which binds them to the participant prior to the computation, ensuring high security and, subsequently, trusted validity.

3.3 Proof of provenance

Establishing proofs regarding the origin of data can give credibility to statements in which such data is used, therefore it is crucial in some cases to assert that such data is provided by trusted entities.

Of course, as we have already talked about, digital signatures and certificates exist for this purpose, but Zhang et al. propose a different solution that leverages ZKP over TLS connections to guarantee to some interested party that the provenance of such data is genuine [44].

In their paper, they make two important points:

- Current solutions not only require server-side modifications but are based on undesirable trust assumptions.
- Users should be free to export their data, without help and permission from data holders, with preserved integrity.

TLS, or Transport Layer Security, is a secure communication protocol widely used to establish secure connections between clients and servers, ensuring confidentiality, integrity, and authenticity of data transmitted over the network. Its big limitation, however, is that it doesn't allow a user to prove to third parties that a piece of authentically accessed data came from a particular source, as it can't be exported securely.

To this effect, they propose DECO - Decentralized Oracle - a secure improvement over existing Oracle schemes capable of providing zero-knowledge proofs of provenance over TLS connections, that enable more accessible data without the need for server-side modifications. It can also provide proofs over substrings of session-data commitments which serve substantial efficiency over general ZKP.

4

Proposed Solution

In this chapter, we present our solution and the direction we want to take to ensure its development meets our goal: trusted inputs for SMPC, by leveraging ZKP over Signed Identity Documents and integrating them with STAR for input legitimacy. Firstly we will focus on the requirements of such a solution, and how that will shape the decisions we take in regards to technologies used. Afterwards, a description of the system architecture and justification of our choices, mostly outlined by the requirements specified previously. Next, the evaluation strategy, and how we chose to test our solution. Finally, a work plan is provided.

4.1 Requirements

Our proposed solution must adhere to the requirements outlined by the remarks in the current section.

SMPC has its foundations set on two properties: **privacy** and **correctness**. As previously mentioned in 2.1, they provide guarantees that no information apart from the result is revealed, and that all participants capture the correct computation result, respectively.

As was noted in 2.2, STAR already ensures these properties and provides proof of their validity, which can be used to evaluate if they still hold after our solution is implemented. Therefore:

Requirement 1. *It is crucial to ensure, throughout our proposed solution, that the properties of privacy and correctness remain true, regardless of ensuring valid inputs.*

As was specified earlier in 2.3.1, Cartão de Cidadão has a smart card that holds, besides general individual information, numerous certificates provided by a CA and a private key stored in a QSCD that requires PIN authentication. This private key is used to sign digital documents, and the certificates enable any third party to validate the signature. Therefore:

Requirement 2. *It is required to collect the card data through a card reader so that proofs can be built with them as witnesses.*

Another important requirement we wish to account for is that STAR won't require any communication with third parties, such as the CA responsible for the certificates present in Cartão de Cidadão. Our solution must remain as simple and as efficient as possible and rely solely on inputs provided by clients, whose task of authentication with these CA might depend on their behalf.

Requirement 3. *STAR is exempt from any communication done with a third party CA which if needed must be strictly done by the client.*

A ZKP of the Cartão de Cidadão must ensure the 3 properties specified in 2.4, **completeness**, **soundness** and **zero-knowledge**, which provide guarantees that proofs of true statements result in acceptance, while false ones don't, and that the only information revealed is that of the statement being proven.

Requirement 4. *The ZKP must ensure the properties of completeness, soundness, and zero knowledge.*

Furthermore, the ZKP must be non-interactive, and a decision has to be made regarding using zk-SNARK or zk-STARK since both have important characteristics that could benefit the protocol's efficiency in different ways, like faster verification in the case of zk-SNARK, against zk-STARK's lack of a trusted setup.

Requirement 5. *The ZKP must be of the non-interactive type.*

4.2 System Architecture

4.2.1 Zero Knowledge Proof

As denoted by requirement 5, the ZKP will be of the non-interactive type, such as zk-SNARK or zk-STARK, since the protocol has to be as efficient as possible, but also account for multiple provers and verifiers. Both of these differ regarding proof size, and performance on proof generation or verification, but ultimately the differentiating factor is the need for a trusted setup, which zk-STARK doesn't require. We must still decide which is the best approach, so if both types are developed and integrated for use in the protocol, we can evaluate which meets our requirements in a more efficient manner.

For now, we define the architecture of our ZKP, first with known entities such as the CA, Individual's Certificate, the Individual itself, and the Message sent to the protocol, declared in figure 4.1.

These entities possess parameters required to create the ZKP:

- The CA has a Public Key used to validate certificates they have issued, which have been signed using their Secret Key.

- The Individual Certificate is the certificate issued by the CA, and has present two important parameters: The CA's signature of the certificate and the certificate's Public Key used to validate the individual's signed documents.
- The Individual Card has all data regarding the Individual and the Certificate assigned by the CA, along with a private Signing Key.
- The Signed Message that has the statement the individual wants to prove, the card's data, the Signature on that message, and the Individual's Certificate once again.

```
CA {
  secretKey,
  publicKey
}

IndividualCertificate {
  "issuer":"CA",
  "issuedTo":"Individual"
  caSignature,
  certificatePublicKey
}

IndividualCard {
  data: {
      "name": "name"
      "age": 10,
      ...
  },
  IndividualCertificate,
  individualSigningKey
}

SignedMessage {
  statement,
  cardData,
  individualSignature,
  IndividualCertificate
}
```

Figure 4.1: ZKP Architecture Entities

The algorithms used to produce and validate a certificate, **Issue_Certificate** and **Validate_Certificate**, respectively, along with the algorithms for signature and verification of documents, **Sign** and **Verify_Sig**, in that order, are declared in 4.2, but without any logic implemented, just the expected behavior. Note, however, that the algorithm used to produce a certificate is only present as a means to demonstrate the relation between itself and the validation algorithm, which will be required to guarantee the certificate is legitimate.

```
//CA Certificate algorithm:
Issue_Certificate(CA.secretKey, unsignedCertificate)
    -> IndividualCertificate

//Validate Certificate algorithm:
Validate_Certificate(IndividualCertificate,
    IndividualCertificate.caSignature, CA.publicKey)
    -> 1/0

//Individual Signature algorithm:
Sign(statement, IndividualCard.individualSigningKey,
    IndividualCard.IndividualCertificate, IndividualCard)
    -> SignedMessage

//Individual Signature Verification algorithm:
Verify_Sig(Message, Message.individualSignature,
    Message.IndividualCertificate.certificatePublicKey)
    -> 1/0
```

Figure 4.2: ZKP Architecture Algorithms

The algorithms used by ZKP for proof generation and proof verification, **Generate_ZKP** and **Verify_ZKP**, are declared in figure 4.3, and although they also lack any logical implementation, they are expected to hold some embedded relations. These relations are denoted by 1, 2 and 3, which are required to hold, so that **Verify_ZKP**'s outputs 1.

```
//KP generate:
Generate_ZKP(secret=SignedMessage.cardData,
    statement=SignedMessage.statement,
    SignedMessage.individualSignature,
    SignedMessage.IndividualCertificate, CA.publicKey)
    -> proof

//ZKP verify:
Verify_ZKP(SignedMessage.statement,
    SignedMessage.individualSignature,
    SignedMessage.IndividualCertificate, CA.publicKey, proof)
    -> 1/0
```

Figure 4.3: ZKP Architecture Proof Algorithms

Relation 1. *SignedMessage.statement == 1*

Relation 2. ***Verify_Sig****(SignedMessage, SignedMessage.individualSignature, SignedMessage.IndividualCertificate.certificatePublicKey) == 1*

Relation 3. ***Validate_Certificate****(SignedMessage.IndividualCertificate.caSignature, CA.publicKey) == 1*

As an example, suppose Alex is an honest individual that holds authenticated credentials. He could choose to provide a statement from the likes of **SignedMessage.cardData.name == "Alex"**, claiming his name to be "Alex". Using his card data as a witness to the proof, all relations should hold whenever the verifying algorithm ran, thus providing confidence in Alex's statement.

If Alex were to be dishonest however, or hold invalid credentials, at least one of the relations wouldn't hold, thus allowing the protocol to reject Alex's message.

4.2.2 Messages

The messages in STAR hold measurements ciphered with a key (common among equal measurements), a share of that same key, and a label that refers to that specific measurement value. Once k measurements with the same label are collected, the aggregator is able, through the collected k shares of the ciphering key, to decipher the measurement, and learn its value.

Due to the simplicity of the message, and because of our desire to be as little invasive as possible, all that is required to include a proof of validity is adding an extra field, separated from the main fields mentioned above. So, the structure of the message would resemble something of the likes of figure 4.4.

```
STAR {
    cipheredMeasurement,
    share,
    label
}
ZKP {
    proof
}
```

Figure 4.4: Proposed Message Structure

4.2.3 Protocol

To preserve STAR's existing functionality and performance, we wanted to avoid as many as possible changes to its existing protocol architecture, shown in figure 2.2. As mentioned in 4.2.2, we opted to add to the existing message structure the minimum necessary to satisfy both STAR's requirements and our solution's.

Even though the proof generator and the proof verifier aren't as straightforward, we kept the same line of taught regarding a minimal approach. We wanted a solution that was as less invasive as possible, therefore, we decided on using a modular approach.

To do so, we decided to have the proof generation phase be done outside the scope of the protocol itself, being generated by the client and sent along with their measurement in the message.

Thus, our only requirement is to effectively design a protocol solution capable of intersecting those messages before they reach the Aggregation Server, in order to validate their proofs. We call this the Validation Phase, which acts as a middle-man between the Message Phase and the Aggregation Phase, filtering out invalid messages and letting through valid ones. The architecture of this solution would look like what is presented in figure 4.5, where the Proof Generation phase isn't included but has its output included in the message.

Currently, the Aggregator receives client messages holding ciphered measurements and only reveals them once at least k equal measurements have been collected, indicated by the common labels that identify them, using the shares they include for deciphering. The behavior of the protocol using our solution would be just about the same, except that the aggregator would now only receive client messages if they hold valid proofs, which are verified by the Validation Phase.

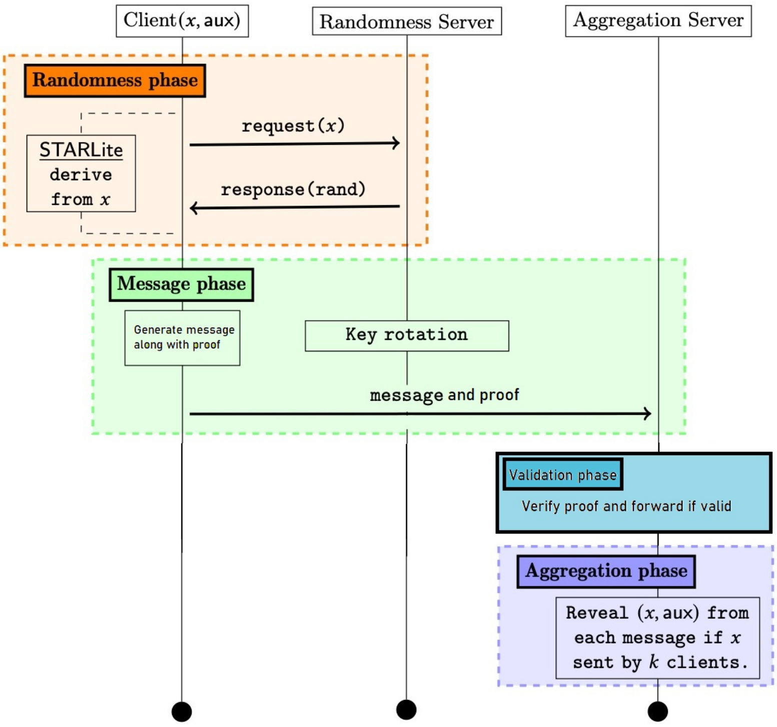


Figure 4.5: STAR solution architecture adapted from [14]

4.3 Evaluation and Testing

The evaluation and testing of our proposed solution is of utmost importance to ensure all properties are met, and in order to gather metrics that validate its efficiency. By doing it in a periodic way, it serves as a way to detect flaws in our solution and lead to improvements throughout development.

4.3.1 Cryptographic Guarantees

To establish cryptographic guarantees, we will define a Formal Security Model that outlines the security properties and assumptions of the protocol. Within this model, an Ideal Functionality will be specified, serving as a benchmark against which the protocol's behavior is evaluated. The Ideal Functionality represents an idealized scenario where the protocol operates flawlessly, allowing us to assess whether the real protocol behaves as intended, even under sub-optimal conditions, by assessing their indistinguishably.

We will be guaranteeing the properties of **correctness**, **privacy**, **completeness**, **soundness**, and **zero-knowledge**. Furthermore, by leveraging the known STAR Ideal Functionality, we may assume SMPC properties remain valid, thus, there's the possibility that we may only have to verify the ZKP ones in addition.

We will be providing proofs of security for this new version of STAR with respect to the aforementioned Security Model.

Proofs regarding the verification of these properties will be required to support the claims of our Formal Security Model.

4.3.2 Performance

We aim to perform performance evaluation using thousands of clients, preferably using a cluster such as the NOVA School of Science and Technology (FCT) Informatics Department's Cluster, using real-world data sets or, if not realistically achievable, due to the nature of the data set itself, use generated pretend cards, which will be used by clients in the STAR protocol.

We will be evaluating the efficiency according to the time it takes to process and compute the results compared to the costs. We will vary the number of clients and message sizes, mimicking usual network usage.

A final comparison with STAR will be done to determine the impact of our solution on the existing protocol's efficiency, and if it's acceptable in real-world scenarios.

4.4 Work Plan

This section outlines the tasks required to develop the solution and elaborate the dissertation.

4.4.1 Tasks

1. **Planning and Design**: Planning is necessary to estimate the amount of work needed and what resources might be required. Besides, design is important as it helps set short-term objectives to focus on during the development phase, in order to reach the long-term ones.

2. **Study of Cartão de Cidadão's Architecture**: Learning the specifics of the card, how to connect it to a device in order to extract information, and possibly contact individuals who have experience or have worked on these cards, will help us determine what are the possibilities the card provides for the next phases, and what barriers we may need to overcome.

3. **Study of Programming Language and Library**: Since STAR is developed with Rust, learning the programming language will be necessary to eventually integrate it with our solution. Besides, even though this doesn't implicate we won't choose another language, many libraries for building ZKP use Rust, which should facilitate integration with the protocol.

4. **Protocol Development**

    a) **Zero Knowledge Proof Development**: Firstly we'll have to come up with a ZKP over the card, to be used in the protocol.

    b) **Integration with STAR**: After developing the ZKP, we'll have to integrate with STAR which implicates the development of a protocol that should be able to communicate with it and use the proof algorithms.

    c) **Intermediate Testing**: We will periodically test our solution to identify issues we can improve on.

5. **Evaluation** Like testing, an evaluation will be done periodically to evaluate efficiency as mentioned in 4.3.2, with a full evaluation in the end to provide results used in the final document.

6. **Writing of the Document**: Over the next months, drafts will be written and improved on as the development advances, culminating in a final deliverable.

4.4.2 Schedule

The following schedule shown in figure 4.6 specifies the timeline of the tasks mentioned in the prior subsection.

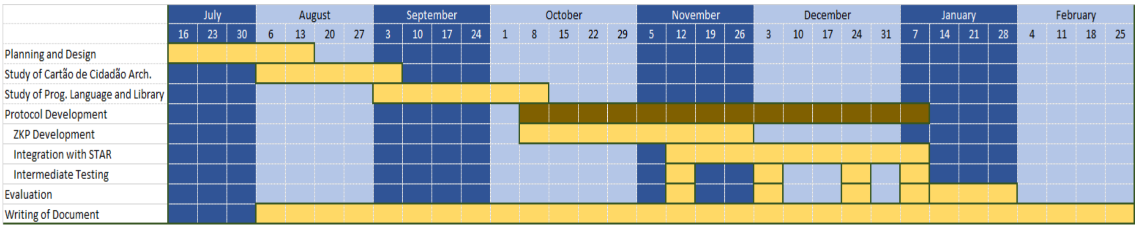


Figure 4.6: Gantt chart of the task schedule

5 Preliminary Work

The field of cryptography isn't explored much in our degree, and so, a lot of study work [32, 35] had to be done to cover all the different concepts explored in the dissertation. To achieve this, many books and papers had to be read to get accustomed to, and understand, the language used to explain problems and propose solutions. This was a necessary effort as the materials aren't simple to understand without a base knowledge to make sense of what is written.

While we wish to make improvements on SMPC, before taking any steps towards the solution, we had to get an understanding of the already existing properties it assures. Without this understanding, there's room for mistakes that would potentially violate existing properties while trying to ensure new ones, and that can't be the case.

Next, to support our solution hypothesis, we had to understand ZKP and the existing types to make a conscious decision on which would be the best type to use, in order to achieve our goal, and how this decision would consequently shape the infrastructure of our solution. By deciding on a specific type, the next thing to do was establish a core set of tools that would enable to development of the solution using these technologies. Such tools include libraries, like arkworks.

Although we address the general problem with SMPC, to maintain our goals achievable in the time given for the dissertation, we had to establish a specific case study to experiment with. Therefore, we considered using inputs such as identification documents to validate and collect data from. Although the starting proposal was using passports, due to a lack of sufficient information regarding Portuguese passport specifications, we agreed to focus on Cartão de Cidadão, which not only has specifications available online but also serves many more purposes than the passport alone, allowing for greater use cases.

Understanding which technologies are involved outside the scope of our solution allowed us to detect dependencies, interesting data points to use, and limitations to our solution. Such is the case of the certificates present in identification documents, for example, which enable data points for validation but also, due to their nature, require communication with a CA, although it's something we aim our solution to be detached from.

Apart from this, special considerations will need to be done to develop a protocol that can effortlessly be integrated with STAR and maintain its efficiency. Our ultimate goal is to restructure messages to include zero-knowledge proof arguments and develop a modular protocol solution placed between the client and the aggregator. This way, by intercepting these messages in their new structure, the presence of our protocol as a middle-man would ensure the aggregator's perception of them, and corresponding behavior, to remain the same. Due to them being processed and validated prior to their forwarding, the aggregator would still view them as they were originally established in STAR.

There still is, of course, room for improvement which ultimately, and with high probability, will come to the surface during the development of the solution in the following months to come until the delivery.

Bibliography

[1] W. Agahari, H. Ofe, and M. de Reuver. "It is not (only) about privacy: How multiparty computation redefines control, trust, and risk in data sharing". In: *Electronic Markets* 32 (2022), pp. 1577–1602. DOI: 10.1007/s1252502200572w. URL: https://doi.org/10.1007/s1252502200572w (cit. on p. 2).

[2] M. Naor, ed. *A general composition theorem for secure reactive systems*. Springer Berlin Heidelberg, 2004, pp. 336–354 (cit. on p. 5).

[3] E. Ben-Sasson et al. *Scalable, transparent, and post-quantum secure computational integrity*. 2018. URL: https://starkware.co/wp-content/uploads/2022/05/STARK-paper.pdf (cit. on p. 10).

[4] T. Beth. "Efficient zero-knowledge identification scheme for smart cards". In: *Advances in Cryptology—EUROCRYPT'88: Workshop on the Theory and Application of Cryptographic Techniques Davos, Switzerland, May 25–27, 1988 Proceedings 7*. Springer. 1988, pp. 77–84 (cit. on p. 12).

[5] N. Bitansky et al. "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*. ITCS '12. Cambridge, Massachusetts: Association for Computing Machinery, 2012, pp. 326–349. ISBN: 9781450311151. DOI: 10.1145/2090236.2090263. URL: https://doi.org/10.1145/2090236.2090263 (cit. on p. 10).

[6] M. Blum, P. Feldman, and S. Micali. "Non-Interactive Zero-Knowledge and Its Applications". In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. STOC '88. Chicago, Illinois, USA: Association for Computing Machinery, 1988, pp. 103–112. ISBN: 0897912640. DOI: 10.1145/62212.62222. URL: https://doi.org/10.1145/62212.62222 (cit. on p. 10).

[7] S. Boeyen et al. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF, 2008-05. URL: https://datatracker.ietf.org/doc/html/rfc5280 (visited on 2023-07-02) (cit. on p. 9).

[8] D. Boneh et al. *Lightweight Techniques for Private Heavy Hitters*. 2023. arXiv: 2012.14884 [cs.CR] (cit. on p. 8).

[9] M. Burmester, Y. Desmedt, and T. Beth. "Efficient Zero-Knowledge Identification Schemes for Smart Cards". In: *The Computer Journal* 35.1 (1992-02), pp. 21–29. ISSN: 0010-4620. DOI: 10.1093/comjnl/35.1.21. eprint: https://academic.oup.com/comjnl/article-pdf/35/1/21/1116430/35-1-21.pdf. URL: https://doi.org/10.1093/comjnl/35.1.21 (cit. on p. 12).

[10] J. Camenisch and G. M. Zaverucha. "Private Intersection of Certified Sets". In: *Financial Cryptography and Data Security*. Ed. by R. Dingledine and P. Golle. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 108–127. ISBN: 978-3-642-03549-4 (cit. on p. 13).

[11] Chainlink. *zk-SNARK vs zkSTARK - Explained Simple | Chainlink*. Chainlink Blog, 2023-02. URL: https://blog.chain.link/zk-snarks-vs-zk-starks/ (visited on 2023-07-03) (cit. on p. 11).

[12] ConsenSys. *Introduction to zk-SNARKS*. ConsenSys, 2017-03. URL: https://consensys.net/blog/developers/introduction-to-zk-snarks/ (visited on 2023-07-08) (cit. on p. 11).

[13] R. Cramer, I. B. Damgård, and J. B. Nielsen (aut). *Secure Multiparty Computation*. Cambridge University Press, 2015-07, pp. 16–17. URL: https://books.google.pt/books?hl=pt-PT&lr=&id=HpsZCgAAQBAJ&oi=fnd&pg=PR9&dq=Secure+Multi+Party+computation&ots=aPKfwiJ5uP&sig=4osn3H6ivOOXsrU9ri4bBxEvkIE&redir_esc=y#v=onepage&q=Secure%20Multi%20Party%20computation&f=false (visited on 2023-06-06) (cit. on p. 6).

[14] A. Davidson et al. "STAR". In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2022-11. DOI: 10.1145/3548606.3560631. URL: https://doi.org/10.1145%2F3548606.3560631 (cit. on pp. 2, 7, 20).

[15] A. De Santis, S. Micali, and G. Persiano. "Non-Interactive Zero-Knowledge Proof Systems". In: *Advances in Cryptology — CRYPTO '87*. Ed. by C. Pomerance. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 52–72. ISBN: 978-3-540-48184-3 (cit. on p. 10).

[16] W. Du and M. J. Atallah. "Secure multi-party computation problems and their applications". In: *Proceedings of the 2001 workshop on New security paradigms - NSPW '01* (2001). DOI: 10.1145/508171.508174. (Visited on 2020-04-30) (cit. on p. 1).

[17] U. Fiege, A. Fiat, and A. Shamir. "Zero Knowledge Proofs of Identity". In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. STOC '87. New York, New York, USA: Association for Computing Machinery, 1987, pp. 210–217. ISBN: 0897912217. DOI: 10.1145/28395.28419. URL: https://doi.org/10.1145/28395.28419 (cit. on p. 12).

[18] O. Goldreich, S. Micali, and A. Wigderson. "How to Play ANY Mental Game". In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. STOC '87. New York, New York, USA: Association for Computing Machinery, 1987, pp. 218–229. ISBN: 0897912217. DOI: 10.1145/28395.28420. URL: https://doi.org/10.1145/28395.28420 (cit. on p. 6).

[19] O. Goldreich. "Secure Multi-Party Computation". In: *Manuscript. Preliminary Version* (1999-03) (cit. on p. 5).

[20] O. Goldreich and H. Krawczyk. "On the Composition of Zero-Knowledge Proof Systems". In: *SIAM Journal on Computing* 25 (1996-02), pp. 169–192. DOI: 10.1137/s0097539791220688. (Visited on 2021-03-04) (cit. on p. 10).

[21] O. Goldreich, S. Micali, and A. Wigderson. "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems". In: *Journal of the ACM* 38 (1991-07), pp. 690–728. DOI: 10.1145/116825.116852. URL: https://people.csail.mit.edu/silvio/Selected%5C%20Scientific%5C%20Papers/Zero%5C%20Knowledge/Proofs_That_Yield_Nothing_But_Their_Validity_or_All_Languages_in_NP_Have_Zero-Knowledge_Proof_Systems.pdf (visited on 2023-06-25) (cit. on p. 10).

[22] *The knowledge complexity of interactive proof-systems*. Association for Computing Machinery, 1985, pp. 291–304. DOI: 10.1145/22145.22178. URL: https://doi.org/10.1145/22145.22178 (cit. on p. 9).

[23] IRN. *Política de Certificados da EC de Assinatura Digital Qualificada do Cartão de Cidadão*. 2022-01. URL: https://pki.cartaodecidadao.pt/publico/politicas/POL23_PC.ECAsC_1.0_Jan.2022_sig_INCM.pdf (visited on 2023-07-02) (cit. on p. 9).

[24] Y. Ishai et al. "Zero-Knowledge Proofs from Secure Multiparty Computation". In: *SIAM Journal on Computing* 39 (2009-01), pp. 1121–1152. DOI: 10.1137/080725398 (cit. on p. 6).

[25] G. Lax, F. Buccafurri, and G. Caminiti. "Digital Document Signing: Vulnerabilities and Solutions". In: *Information Security Journal: A Global Perspective* 24 (2015-02), pp. 1–14. DOI: 10.1080/19393555.2014.998843 (cit. on p. 8).

[26] P. McDaniel and T.-M. Shih. *Sustainability is a Security Problem*. Zoom, 2022-11. URL: https://acm-org.zoom.us/rec/play/Qq2cWNOs7adzTqdHwlGac7WNK1Q5HRcd87_RuG_nPPxdSGM9ZqX2hmAVAjD6Q-I_UKeCz2RrTa7HRu_o.yzXRQo0DJkJE646v?startTime=1667925686000&_x_zm_rtaid=JdFqTUFwRe2aQtobT3IC_w.1675347503231.e3aebedc42d134a4fe7f8cd3e51c209d&_x_zm_rhtaid=200 (visited on 2023-07-04) (cit. on pp. 2, 13).

[27] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. Fifth Printing (August 2001). CRC Press, 1996. URL: https://cacr.uwaterloo.ca/hac/ (visited on 2023-06-28) (cit. on p. 10).

[28] *NP-complete problem | mathematics*. Encyclopedia Britannica, 2023-06. URL: https://www.britannica.com/science/NP-complete-problem (visited on 2023-06-28) (cit. on p. 10).

[29] M. Petkus. "Why and How zk-SNARK Works". In: (2019-06). DOI: 10.48550/arxiv.1906.07221. (Visited on 2023-07-03) (cit. on p. 10).

[30] *Política de Certificado de Autenticação*. 2019-01. URL: https://pki.cartaodecidadao.pt/publico/politicas/PJ.CC_24.1.2_0011_pt_AuC.pdf (visited on 2023-07-02) (cit. on p. 9).

[31] G. Português. *Specimen of Cartão de Cidadão*. URL: https://www.autenticacao.gov.pt/o-cartao-de-cidadao (visited on 2023-07-05) (cit. on p. 9).

[32] M. M. Prabhakaran and A. Sahai. *Secure Multi-Party Computation*. IOS Press, 2013-01. URL: https://books.google.pt/books?hl=pt-PT&lr=&id=1h7vAgAAQBAJ&oi=fnd&pg=PR1&dq=secure+multi-party+computation+&ots=umBhiGl0ru&sig=SbssxD-4jgefdx9sygCP5DwNAr8&redir_esc=y#v=onepage&q&f=false (visited on 2023-05-31) (cit. on p. 24).

[33] U. B. C. RDI. *Lecture 10.3: What is a zk-SNARK?* Youtube, 2021-10. URL: https://www.youtube.com/watch?v=gcKCW7CNu_M&t=1006s (visited on 2023-06-09) (cit. on p. 11).

[34] M. Rosenberg et al. *zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure*. Cryptology ePrint Archive, Paper 2022/878. https://eprint.iacr.org/2022/878. 2022. URL: https://eprint.iacr.org/2022/878 (cit. on p. 13).

[35] M. Rosulek. *The Joy of Cryptography*. URL: https://joyofcryptography.com/pdf/book.pdf (visited on 2023-06-06) (cit. on p. 24).

[36] D. Rotaru. *awesome-mpc*. GitHub, 2022-10. URL: https://github.com/rdragos/awesome-mpc (cit. on p. 7).

[37] V. Tan. *Awesome Zero Knowledge*. GitHub, 2023-07. URL: https://github.com/ventali/awesome-zk (visited on 2023-07-03) (cit. on p. 11).

[38] P. Team. *zk-SNARKs vs zk-STARKs — Comparing Zero-knowledge Proofs*. Panther Protocol Blog, 2022-09. URL: https://blog.pantherprotocol.io/zk-snarks-vs-zk-starks-differences-in-zero-knowledge-technologies/#what-are-zk-snarks (visited on 2023-07-03) (cit. on p. 11).

[39] E. Tonkin and J. Allinson. *Signed metadata: Method and application*. International Conference on Dublin Core and Metadata Applications, 2006. URL: https://dcpapers.dublincore.org/pubs/article/view/861 (cit. on p. 8).

[40] C. Veloso and M. Almeida. *A Segurança da Informação no Cartão do Cidadão Português*. 2009-12 (cit. on p. 9).

[41] W. Vercruysse. *What is a qualified signature seal creation device QSCD - eSignature Knowledge Base -*. ec.europa.eu, 2019-12. URL: https://ec.europa.eu/digital-building-blocks/wikis/display/ESIGKB/What+is+a+qualified+signature+seal+creation+device+QSCD (visited on 2023-07-02) (cit. on p. 9).

[42] W3C. "Decentralized Identifiers (DIDs) v1.0 becomes a W3C Recommendation". In: *W3C* (2022-07). URL: https://www.w3.org/press-releases/2022/did-rec/ (visited on 2023-07-05) (cit. on p. 13).

[43] A. Yao. *Protocols for Secure Computations*. 1982. URL: https://crysp.uwaterloo.ca/courses/pet/W11/cache/www.cs.wisc.edu/areas/sec/yao1982-ocr.pdf (visited on 2023-06-07) (cit. on p. 5).

[44] F. Zhang et al. *DECO: Liberating Web Data Using Decentralized Oracles for TLS*. 2020. URL: https://arxiv.org/pdf/1909.00938.pdf (cit. on p. 14).