Preface

The ever-increasing pervasiveness of edge computing is creating challenges for users' privacy. Given this state-of-affairs, we decided to pursuit an overview and future directions for novel approaches for privacy-preserving computation. In this process, we highlight of some most important privacy concepts and their application to both Fog Computing and IoT.

While we do not offer a definitive solution for privacy, our work explores several ideas that might lead to significant advances in the area. For this purpose, we explored current literature and discuss the integration of several different approaches.

We start by first exploring three major concepts, namely, blockchain, IoT/fog computing and Multi-Party Computation (MPC). These concepts provide the necessary context and background for developing possible research paths and ideas. For blockchain, we describe some practical frameworks and applications and then describe which ones can have an impact in IoT. We then move to investigate and describe current approaches that combine blockchain, IoT and fog computing. Lastly, we explore MPC. Since it is a concept that promotes privacy without a third party, we explore its use in conjunction with the aforementioned concepts.

Furthermore, we offer an overview on some potential frameworks for MPC and assess the feasibility of their integration with other privacy concepts. We conclude by discussing current unsolved problems and possible future research directions.

Department of Computer Science,
December 2016

Patrícia R. Sousa
Luís Antunes
Rolando Martins

Acknowledgements

This work was supported by Project "NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016", financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

Contents

List of Contributors

Patrícia R. Sousa
Department of Computer Science, Rua do Campo Alegre, e-mail: psousa@dcc.fc.up.pt,

Luís Antunes
Department of Computer Science, Rua do Campo Alegre e-mail: lfa@dcc.fc.up.pt,

Rolando Martins
Department of Computer Science, Rua do Campo Alegre, e-mail: rmartins@dcc.fc.up.pt

Acronyms

GC	Garbled Circuits
HE	Homomorphic Encryption
IoT	Internet of Things
MPC	Multi-Party Computation
SCAPI	Secure Computation API
SFE	Secure Function Evaluation
SMPC	Secure Multi-Party Computation
TPC	Two-Party Computation
VIFF	Virtual Ideal Functionality Framework
VSS	Verifiable Secret Sharing

Part I
The Present and Future of Privacy-Preserving Computation in Fog Computing

Chapter 1
The Present and Future of Privacy-Preserving Computation in Fog Computing

Patrícia R. Sousa, Luís Antunes and Rolando Martins

Abstract Edge computing is becoming a prevalent alternative to the classical cloud paradigm. Instead of relying on a centralized infrastructure, hyper local clouds which are used in fog computing and edge clouds, focus on performing computation and storing data locally. This increase of locality allows an enhancement of privacy and interactivity with end users. In particular, this allows computation to be performed near the users and thus shielding them from directed tracking. However, current computational frameworks are not suitable to implement privacy preserving computation on the edge. Multi-party computation (MPC) poses itself as a suitable option to offer the basic building block for building decentralized privacy preserving computational frameworks. In MPC, each party has to share their own data (inputs) with the other parties over an public function while ensuring that no private information is leaked. One of the recent approaches in this field is Enigmas computation model based on an optimized version of secure multi-party computation which removes the need for a trusted third party. This model works in parallel with blockchain technology that controls the network, manage access control, identities and serves as a tamper proof log of events. In this work, we follow this path of privacy based on blockchain with secure multi-party computation. We start describing the related work, then the current state of the art in terms of security and privacy and finally new directions in the field with special focus in security and privacy.

Patrícia R. Sousa
Department of Computer Science, Rua do Campo Alegre, e-mail: psousa@dcc.fc.up.pt

Luís Antunes
Department of Computer Science, Rua do Campo Alegre e-mail: lfa@dcc.fc.up.pt

Rolando Martins
Department of Computer Science, Rua do Campo Alegre, e-mail: rmartins@dcc.fc.up.pt

1.1 Introduction

The increasing attacks on users' privacy reveals the economical importance of sensitive data for both companies and criminals. The need to reduce time-to-market has lead companies to deploy edge computing systems without security and privacy by design. The exponential growth of the data generated by this type of systems is chronically exacerbating the problem.

Fog computing is one of the most predominant example of edge systems, and is one facet of the overarching concept that is the Internet-of-Things (IoT). It can be best described as the relocation of computing, preferably closer to devices near the end user. Current security and privacy approaches used in cloud computing infrastructures are not appropriate for the underlying requirements of fog computing, namely their intrinsic decentralized nature. Furthermore, privacy preserving frameworks are still not available in public cloud computing, and are an active field of research. Current approaches use pseudo-anonymization techniques that can be deanonymized with the aggregation of multiple sources of information.

Decentralized in nature, fog computing offers a pathway that we argue can be used to create novel privacy preserving techniques. By keeping the data near the end user it provides a natural barrier to large scale data collection performed by big-data/analytics based companies. By itself, edge systems do not provide the necessary mechanisms to meet this goal. To fill this gap, there are some privacy primitives that we want to explore and potentially combine.

The first relies on Secure Multi-Party Computing (MPC), that can be used to compute responses based on confidential data, so that when the protocol is completed users know only their own input and the answer. Concurrently, blockchains offer public ledgers that reduce and potentially eliminate the presence of centralized trusted entities. As an example, within the Bitcoin virtual currency, a blockchain is the data structure that represents a financial accounting entry or a record of a transaction. Each transaction is digitally signed with the purpose of guaranteeing its authenticity and ensuring that no one adulterated it, so that the record itself and the transactions within it are considered of high integrity.

In this chapter, we focus on these two different approaches as a means to create solutions to enhance privacy preserving approaches for IoT. We also explore the combination of blockchain and Multi-Party Computation techniques on the edge.

The next sections of this chapter are organized as it follows: Section 1.2 presents the blockchain and their limitations in the integration with IoT. Also, the same section presents the IoT, fog computing and blockchain concepts together with their respective applications. In Section 1.3, we describe the Multi-Party Computations, the description and comparison between MPC frameworks and also, some future research directions of this field. Section 1.4 presents both Multi-Party Computation and blockchain and the applications of the concepts together. Also, we describe the future research directions of the combination of the two technologies. Lastly, we have a summary of this chapter in the Section 1.5.

1.2 Blockchain

Blockchain is a decentralized ledger of all Bitcoin [1] transactions across a peer-to-peer network, growing as miners add new blocks to it in order to record new transactions. The nodes within the network validate the transaction and the user's status by using known algorithms to ensure that the same Bitcoins were not spent previously, thus eliminating the double expense problem. A verified transaction can involve cryptocurrency, contracts, records, or other information. Once the transaction is verified it gets combined with other transactions in order to create a new data block within the public ledge. The new block is then added to the existing blockchain, in a way that is permanent and immutable. Only at this point is the transaction considered completed [49] [50].

Using this technology, users can confirm transactions without the need for a central certifying authority, normally enforced by central banks. Other possible applications include fund transfers, settling trades and voting.

Blockchain limitations for IoT

The integration of blockchain in IoT is not straightforward, since the majority of IoT devices are resource restricted and low latency is desirable. Also, there are a large number of nodes in IoT networks and devices are bandwidth-limited. In order to successfully integrate blockchain with IoT several critical challenges will have to be addressed, namely:

- Mining is computationally intensive;
- Mining of blocks is time consuming;
- Blockchain scales poorly as the number of nodes in the network increases;
- The underlying BC protocols create significant overhead traffic. [52]

1.2.1 The Internet of Things, Fog Computing and Blockchain

Despite the fast paced development of new architecture frameworks for IoT, privacy and security remains a second class citizen, leading to several open privacy and security challenges. There are several advantages of using blockchain technology that, in theory, can potentially help to improve privacy and security, majorly through decentralization. As described in [52], user identities must be kept private and this can be accomplished when using a blockchain. As they are decentralized by design, blockchains offer scalability and robustness by using the resources from all participating nodes, and in the case of IoT, from all devices. In the process it

[1] Bitcoin is a digital currency and online payment system, also called digital cash. It works in a decentralized way, that uses peer-to-peer to enable payments between parties without the need of mutual trust. The payments are made in Bitcoins that are digital coins issued and transferred by the Bitcoin network [48].

also eliminates many-to-one traffic flows, which reduces delays, and overcomes the problems associated with the presence of single point-of-failure [22].

However, there are problems that need to be solved in order to allow practical usage of blockchain. For example, IoT devices normally have limited resources that are not enough to properly support cryptocurrency mining due to is computational cost. It is desirable for IoT applications to have low latency and low traffic in the network. Mining of blocks is time consuming and creates significant overhead traffic, which is undesirable. Moreover, blockchain does not properly scale with the ever-increasing introduction of nodes in the network [52].

In order to solve the current limitations within IoT that would provably compromise a seamless integration with blockchains, a new paradigm must be used. We argue that fog computing is a prime candidate be employed in this scenario. One of its main goals is to deal with the current limitations of public cloud computing when dealing with services and applications that requires very low latency, "Local awareness" and mobility (including vehicular mobility).

Follows the presentation of approaches that explore the integration of between IoT, blockchain and fog computing. We will describe their applications and how they address the aforementioned challenges.

IOTA

IOTA is designed to be as lightweight as possible, unlike the complex and heavy duty blockchain operations of Bitcoin. IOTA is a novel transactional settlement and data transfer layer for IoT. The first part in 'IOTA' emphases the importance that is conceded to the IoT. It is based on a new distributed ledger, the Tangle, which overcomes the inefficiencies of current blockchain designs and introduces a new way of reaching consensus in a decentralized peer-to-peer system [70].

With the growth of the number of devices that exist in the IoT, that can reach tens of billions of connected devices in the next decade, one of the main needs are the interoperability and resource sharing. For this, IOTA enables companies to explore new business-to-business (B2B) models by making every technological resource a potential service to be traded on an open market in real time and without fees.

Recently, new approaches for IoT have been proposed with the introduction of Fog and Mist [2] [37]. The main goal of these new paradigms is to decrease the network latency to cloud servers that can located far away from end-devices. Hence, it is crucial for the industry to rely on a free real-time, low-latency and decentralized settlement system [37].

IOTA combines both Fog and Mist into a new distributed computing solution. This can be seen as a combination of smart sensors with built-in computational capabilities (mist computing) with nearby processing stations (fog computing). IOTA micro-transactions enable party A's sensor data to be processed by Party B's processors in real time. In return, Party B can use IOTA to use resources from Party A or any other technological resource from other parties [39].

[2] Mist computing decreases latency and increases subsystems' autonomy. This takes fog computing concepts further by pushing some of the computation to the very edge of the network, to the sensor and actuator devices that make up the network. [69]

In this new autonomous Machine Economy, IOTA can be viewed as its backbone. The Tangle [3] ledger is able to settle transactions with zero fees so devices can trade exact amounts of resources on-demand, as well as store data from sensors and data loggers securely and verified on the ledger [35].

IOTA is different from the approaches taken by Bitcoin and Ethereum [4]. One main differentiating factor is that IOTA does not use blockchain, but instead it uses "Tangle", a Directed Acyclic Graph shaping up a tangle. Secondly, blockchain is not suited to support micro-payments. Conversely, Tangle supports micro-payments by enabling IOTA to be efficient, scalable and lightweight. But rather than being isolated paradigms, both approaches can work together. IOTA can communicate to blockchain, that allows a future collaboration between IoT and established digital currencies. In fact, the IOTA project could even be used as an oracle to complete smart contracts [36].

Autonomous Decentralized Peer-to-Peer Telemetry (ADEPT)
The exponential growth that IoT is experiencing is making increasingly important to have decentralized networks as a means to eliminate single point-of-failures that are associated with traditional centralized networks, as a way to increase its robustness and reduce the infrastructure and maintenance costs to manufacturers and vendors. By using the devices themselves as computational, storage and communication nodes, we can build "hybrid" IoT systems where the "edge" complements centralized systems. We argue that edge computing will become a frontier of new economic value, creating an Economy of Things. As a way to spark adoption and change the current *status quo*, IBM and Samsung have developed the Autonomous Decentralized Peer-to-Peer Telemetry (ADEPT) proof-of-concept [40].

With it, they were able to demonstrate a distributed systems capable of sustaining a fully decentralized framework for IoT. As its backbone, ADEPT uses the blockchain to build a decentralized and distributed network of things [41, 42], using a combination of proof-of-work [43] and proof-of-stake [44] to secure transactions. This work was supported by using three distinct protocols:

- **BitTorrent** - BitTorrent is used to the file sharing.
- **Ethereum** - Ethereum is necessary to understand smart contracts and capabilities. At this point, the blockchain comes into the process.
- **TeleHash** - TeleHash is used to make the peer-to-peer messaging, since it is designed to be decentralized and secure, it fits in this system. [45]

As a proof-of-concept for ADEPT, researches have deployed these three protocols that into a commercial washing machine (Samsung W9000) that was pro-

[3] The main innovation behind IOTA is the Tangle, a novel new blockless distributed ledger which is scalable, lightweight and for the first time ever makes it possible to transfer value without any fees. Contrary to todays blockchains, consensus is no-longer decoupled but instead an intrinsic part of the system, leading to decentralized and self-regulating peer-to-peer network [35].

[4] Ethereum is a decentralized platform that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference. [47]

grammed to work with the ADEPT system, making an "Autonomous Washing Machine Orders Detergent" [46]. The goal is to automate the process of the ordering supplies. This process makes use of smart contracts to define the commands to receive a new batch of supplies. This way, the device can order and pay by itself when the capacity of the detergent is low. This payment is made using the blockchain. Later, the retailer receives the notice that the detergent has been paid for and ships it. Moreover, the owner of the washer can also be notified of the purchase details in its smartphone via its home network.

Another use case consists in a decentralized advertising marketplace using Large Format Displays (LFD)s to share and publish content without a centralized controller. The concept consists in an LFD, or more commonly a conventional display, where we can share the screen with anybody.

We have to choose the LFDs where the advertising will be published and also choose the advertisements (video files served by BitTorrent) to be published. Then, the advertiser receives the request through peer-to-peer messaging by TeleHash. After this, the content is shared and published. Finally, the advertiser receives the analytics, confirms the approval and finalizes the payment.

1.3 Multi-Party Computation

After the era of connecting places and connecting people, the Internet of the future will also connect things. These "things" have sensitive information and data that can be shared but requires privacy. Secure multi-party computing is a technique that can be used here, since its purpose is to have multiple parties exchanging secret information privately without the need of a Trusted third party. More formally, MPC consists of two or more parties, where each party has their own secret input. MPC computes some joint function f, that receives as input the secret information of each party.

It can be better explained with one of its well known use cases, commonly referred as the Millionaire's Problem. Assuming that we have three parties: Alice, Bob and Charlie. Each party uses respective inputs x,y and z denoting their salaries. The goal is to find the highest salary of the three, without reveling their respective salaries. Mathematically, this can be achieved by computing:

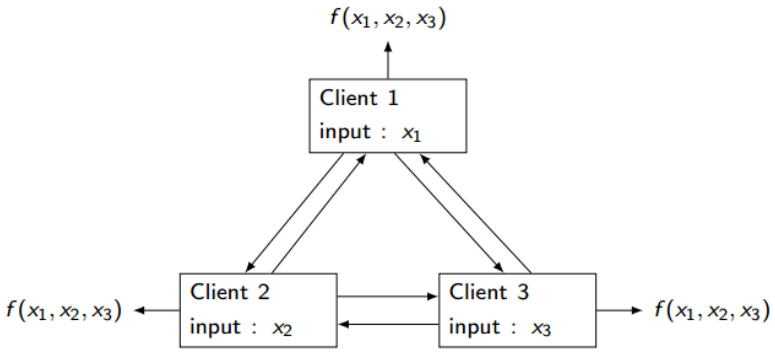
$$f(x,y,z) = max(x,y,z)$$

Each party will share his secret input without reveal. At the end of the protocol, each participant will get only the result of the function f, without getting anything else about the other parties input, i.e., the secret inputs will not be revealed. The security of such protocols is defined with respect to the ideal model where f is computed by a trusted party T_f. During the execution of a protocol, the parties cannot get information about the inputs of the other parties. A third party Tf computes

a function f receiving the inputs from the parties and after this, computes f, and finally sends the output back to the parties.

MPC based on secret-sharing refers to methods for distributing a secret for a group of participants. Each participant has a share of the secret. The secret can be reconstructed only when a sufficient number of shares are combined together, where a threshold cryptosystem *(t+1, n)* where n is the number of parties and $t+1$ is the minimal number of parties to decrypt a secret encrypted with threshold encryption. We can see an image that illustrates MPC in the figure 1.1.

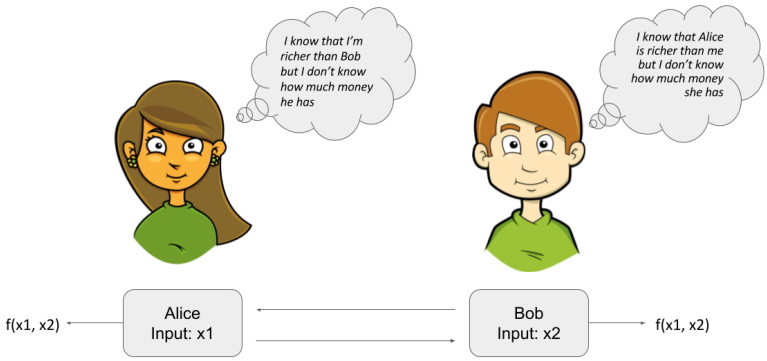
Fig. 1.1 Secure MPC without a trusted third party



A real-world example for MPC's applicability can be one where a patient wants to access his clinical records. He can make use of his private DNA code to make a query to a medical database of DNA related diseases. However, the patient does not want the hospital, and potential others, to know his DNA and health status. At the same time, the hospital does not want to disclose its entire DNA database to the patient. This is a problem, where the privacy must be preserved and can be solved while using MPC.

Extending our previous discussion, the millionaire's problem was first introduced in 1982 by Andrew Yao [38]. Suppose that we have Alice and Bob, and they want to know who is richer without reveal to each other or to a trusted third party the amount of money that they have or any type of additional information. The function $f(x1,x2) : if\ x1 > x2 = Alice$ else $x1 < x2 = Bob$ computes the inputs and returns the name of the richest (we can see the example in the figure 1.2. Alice know that is richer than Bob, but does not know how much money he has, and Bob also, know that Alice is richer, but does not know how much money she has. Therefore, in this protocol the privacy of each data is preserved, since they never reveal their salary.

Fig. 1.2 Millionaire's problem



In [38], it also describes other potential applications of the multi-party computation such as secret voting. It consists in a number of m members having globally to decide on a yes-no action. Each member has to choose an option xi and the result is computed by the function $f(x1,x2,x3,...,xm)$. In turn, this function gives the final result without disclose the opinion of any other members and thus preserving privacy.

Another possible application is related with oblivious negotiation. In this case, we have Alice trying to sell Bob a house, each one having a strategy of negotiation in mind. Alice has possible strategies numbered as $A1,A2,...,At$ and the same for Bob as $B1,B2,...,Bu$. The result (no deal or sell at x dollars,...) will be decided once the actual strategies Ai,Bj used have been determined. The result is wrote as $f(i,j)$. This way, it is possible to carry out the negotiation obliviously, since Alice will not gain any information on Bob's negotiation tactics, expecting that it is consistent with the outcome, and vice-versa.

The last problem presented in [38] focus on privately querying a database. Suppose that Alice wants to compute a function $f(i,j)$ and Bob $g(i,j) = constant$. Bob does not know anything about i in the end. If we assume Bob as a database query system, with j being the state associated with the database, Alice can perform a query with the number i, and then, she can get an answer without getting any other information besides the data stricly required by her query. Conversely, the database system does not know which element was queried by Alice. This allows for users to preserve their privacy while avoiding data leakage from the database system.

1.3.1 Framework Analysis

Frameworks provide a set of solutions to commonly known problems for specific application domains, and are normally supported by a set of libraries. Developers can reuse code provided in these libraries and avoid handling with domain specific

problems or low-level coding techniques. In most cases, frameworks result from a collaborative effort, as such, the burden of maintaining and improving it relies on a community instead of a single individual. A community offers a crowd-sourcing mechanism that enables users and developers to obtain information and resources to overcome issues found.

Despite their intrinsic advantages, there also some disadvantages associated with them. Namely, creating a framework is difficult and time-consuming, i.e. expensive, and the learning curve can be steep. Moreover, they often add up to the size of programs, a phenomenon termed "code bloat" [10]. Over time and depending on the number of features introduced, a framework can become increasingly complex.

This added complexity can surpass the gains obtained from using a framework and the intended reduction in overall development time may not be achieved [56]. However, if the know-how can be further re-used in future projects, then this learning curve can be considered as an investment as it can be amortized across multiple projects [24].

There are some frameworks designed and implemented that enable Secure Multi-Party Computation (SMPC) providing basic MPC functionality that allow algorithm designers to build complex applications. Different flavors can be found for security level, accessibility, software composition, usability, scalability and performance. MPC frameworks allow users to specify a SMPC where a number of parties execute a cryptographic protocol to do some joint computation with an agreed function without leaking any information of their inputs. An example could be an election where the correct tally is computed without revealing any information on the individual votes. Using a framework, the protocol is run without the players revealing anything about their inputs. Follows the set of MPC frameworks that we have tested:

VIFF (Virtual Ideal Functionality Framework)
VIFF is a framework that allows users to specify SMPC. VIFF is implemented in *Python* using *Twisted* [1] Framework to manage communication and GMPY (General Multiprecision PYthon project) [4] more specifically, the GNU Multiple Precision Arithmetic Library for the precision arithmetic. This framework is able to run on any platform where *Python* runs such as, *Linux*, *Windows*, and *Mac OS X*) [3]. Protocols implemented in VIFF can be compositions of basic primitives like addition and multiplication of secret-shared values, or one can implement new primitives. In short, the goal of VIFF is to provide a solid basis where practical applications using MPC can be built [5]. VIFF's features include:

- Arithmetic with shares from Zp or GF(2^8).
- Secret sharing based on Shamir and pseudo-random secret sharing (PRSS).
- Secure addition, multiplication, and exclusive-or of shares.
- Comparison of secret shared Zp inputs, with secret Zp or GF(2^8) output.
- Automatic parallel (asynchronous) execution.
- Secure communication using SSL. [2]

Sharemind
Sharemind is a framework for privacy-preserving computations. It consists in a com-

putation runtime and associated programming library for creating private data processing applications. This enables users to develop and test their custom privacy-preserving algorithms. As a result, one can develop secure multi-party protocols without the explicit knowledge of all implementation details. This also allows developers to test and add their own protocols to the library, as Sharemind is an open-source project [11]. The experimental Sharemind SDK contains the *SecreC 2* programming language that separates public data and secrets on a type system level and an emulator that developers can use to estimate the running time of their applications in a fully secured environment. SecreC programs are fully compatible with the Sharemind Application Server, that provides full cryptographic protection and supports enterprise applications [12].

SPDZ
SPDZ implements a general multiparty computation protocol secure against an active adversary corrupting up to n1 of the n players [9].

The processing model implemented by SPDZ is as follows: a) an offline phase, where some shared randomness is generated, but neither the function to be computed nor the inputs need be known, and; b) an online phase, where the actual secure computation is performed.

In the latter, we have active security [7, 8] with the following associated feature set:

- It uses BDOZ/SPDZ style MACs.
- It uses the n-party variant of the Tiny OT protocol to perform the pre-processing (outlined in some of the papers below).
- Works over any finite field GF(p) for p bigger than 40 bits; which is needed for statistical security. In practice to support floating and fixed point operations p may be 128 bits in size.
- Provides actively secure offline and online phases.
- It provides a python based front end to produce byte-code for execution by the system. [6]

FairplayMP
Fairplay [17] is a full-fledged system that implements generic Secure Function Evaluation (SFE). SFE allows two parties to implement a joint computation, that in real world applications may be implemented using a trusted party, but does digitally without any trusted party. However, the Fairplay system uses Yao's Garbled Circuits (GC) and only supports secure communication between two parties. FairplayMP was created as an extension that appears to counter this limitation and introduce multi parties. The extension to the multi-party case is needed since cryptographic protocols for the multi-party scenario are completely different than protocols for the two-party case [18]. This version implements secure computation using Yao circuits and secret sharing techniques.

SCAPI (Secure Computation API)
SCAPI is an open-source general library tailored for Secure Computation imple-

mentations. This framework provides a flexible and efficient infrastructure for the implementation of secure computation protocols. Moreover, it also provides uniformity by offering a modular code-base to be used as the standard library for Secure Computation. SCAPI is also efficient because is built upon native C/C++ libraries using JNI. SCAPI tries to improve adoption by developers by proving a clean design, streamline source code and detailed documentation [19].

TASTY
TASTY (Tool for Automating efficient Secure Two-Party Computations) is a tool that uses Homomorphic Encryption (HE) or Garbled Circuits (GC) or the combination of both for describing and generating efficient protocols for several privacy-preserving applications [30]. TASTYL, that is the programming language adopted and created by TASTY, is an intuitive high-level language for describing the SFE protocols as sequence of operations on encrypted data (based on GC and HE). Also, TASTY allows to automatically analyse, run, test and benchmark the two-party SFE protocol.

SEPIA
Security through Private Information Aggregation is a Java library for generic SMPC [32, 33]. It is tailored for network security and monitoring applications where the basic operations are optimized for large numbers of parallel invocations. SEPIA's basic primitives are optimized for processing high-volume input data. It uses Shamir's secret sharing scheme and is secure in the honest-but-curious adversary model [34].

1.3.2 Comparison between MPC frameworks

Table 1.1 Comparison between MPC frameworks

Framework	Programming Language	Techniques	Number of participants	Year of creation
VIFF [2] [5]	Python	Secret Sharing	$\geqslant 3$	2007
Sharemind [11] [12]	SecreC (C++)	Secret Sharing	3	2006
SPDZ [6] [7] [8]	Java/C++/Python	Secret Sharing	$\geqslant 2$	2016
SCAPI [19] [20]	Java	GC	$\geqslant 2$	2013
FairPlay [17]	SFDL (Java)	GC	2	2003
FairPlayMP [18]	SFDL (Java)	GC and Secret Sharing	$\geqslant 3$	2006
TASTY	TASTYL (based on Python)	GC and HE	2	2009
SEPIA	Java	Secret Sharing	$\geqslant 3$	2008

In practical terms, a number of frameworks and specialized programming languages have been created to implement and run SMPC protocols. Fairplay, FairplayMP and TASTY were built on top of the idea of "Garbled Circuits (GC)" [5]. Sharemind and SPDZ use additive secret sharing [6] over a ring. In the case of VIFF, FairplayMP and SEPIA, they were built on top of Shamir's secret sharing [7] [27, 29]. Lastly, TASTY uses combinations of GC and HE techniques [30].

The main application for GC is to secure Two-Party Computation. For more than two parties, secret sharing schemes [59] are normally used. All these frameworks support a similar set of primitives, including addition, multiplication, comparisons and equality testing. Programming on these platforms either uses a specialized language, or a standard programming language and library calls, depending on the platform [29].

Some of these frameworks are more accessible for non-experience developers, more specifically, VIFF, Sharemind, SPDZ, FairplayMP, TASTY and SEPIA. SPDZ has the particularity of having the online and offline phases built-in into the framework.

Adding examples to the source code in the frameworks, is something that helps the development stage. For instance, if we start with a new API, sometimes we would not be capable of implement new examples because we don't know the structure of the entire framework. Sometimes, it would be more easier to study an implementation of a known standard protocol to understand the structure. In this case, an example could be the millionaire's problem.

In terms of known programming languages, VIFF, SPDZ and SEPIA can be more easier to adapt. These frameworks use standard programming languages such as Python, Java and C++, making them more accessible. Alternatively, TASTY and FairplayMP have the TASTYL and SFDL specification which may require more adaptation time.

SCAPI is the preferable framework for advanced users, as SCAPI is an open-source general library tailored for Secure Computation implementations. It is best suited for users that already are knowledgeable on how the protocol works and therefore only need a library to implement secure protocols.

---

[5] Yao's GC, utilized for SMPC, allows multiple parties to compute an arbitrary Boolean function on their individual inputs without revealing information about those inputs to any trusted third party, as long as they are semi-honest [58, 26]

[6] "Additive sharing supports efficient addition and multiplication due to the algebraic properties of the scheme. However, floating-point arithmetic is much more sophisticated and contains a composition of different operations, both integer arithmetic as well as bitwise operations." [57]

[7] Shamir's Secret Sharing is a form of secret sharing, where a secret is divided into parts, giving each participant a random part of the secret, where some of the parts or all of them are needed in order to reconstruct the secret. Sometimes, it is used a threshold scheme to define k parts that are sufficient to reconstruct the original secret, since can be impractical to have all participants to combine the secret. [27]

1.3.3 Future Research Directions

In this section, we discuss the main research questions and summarize the identified challenges in the MPC area. For this study, we analyse some papers that present future challenges.

Havron et al [13] describes a problem that can be solved with MPC in the future. Social scientists and researchers are always the need to make data analysis. However, they have to reveal some of the input data to another party to perform the analysis. Often, they can not make the data analysis due to legal restrictions and privacy issues. This can be solved by MPC for scientific analysis of large data. A new research pathway lays on the improvement of MPC implementations to enable novel scientific data methods through the creation of new tools that will make these techniques accessible to social scientists. This points to the need for a closer examination of automatic data-matching between separate datasets with private set intersection, improving fixed-point integer conversion for decimal data values used in computation, and other privacy-preserving applications. To summarize, the ultimate goal is to achieve this without disclosing private information, i.e., the inputs of each party.

Since the number of devices in IoT is growing, the data that is being exchanged is also increasing. For this, it would be important to have a filter in order to identify non-sensitive data, making tools that help us detect sensitive versus non-sensitive data.

The authors of the paper [15] propose an interesting research direction for ”MPCs on Bitcoin” where Alice and Bob can determine who is the wealthiest one based on who has more coins. However, this is only possible if each party is interested in proving that it is the wealthiest one, because every participant can easily pretend to be poorer than it really is and ”hide” its true wealth by transferring it to some other address under its control.

The authors of the paper suggests that analyzing what functionality can be computed this way, i.e., talking into account the problem of the participants may pretend to be poorer than they are, may be an interesting research direction. In our opinion, this can be a possible research direction not only in the ”millionaires problem” but in other problems that are isomorphic in nature, just with varying underlying contexts. There are still open problems in the MPC such as constructing protocols that are secure against ”malleability” and ”eavesdropping” attacks [15].

Another issue is related with the information leakage from memory access patterns, that is solved by cryptographic techniques like oblivious RAM and Private Information Retrieval (PIR). Both techniques can be used in a black-box manner to resolve the aforementioned issues but the actual selection of a scheme and associated security and complexity analysis is still subject for future research. Arguably a more difficult challenge is the protection against misuse of branching instructions where proper code obfuscation seems a possible solution, in particular by hiding control flow operators such as ”then” and ”else”. In this case, the code obscurity is a necessity to thwart chosen instruction attacks by stripping the semantics from

the instructions (much like encryption removes the meaning from a plain-text by casting it into a cipher-text)" [16].

1.4 Multi-Party Computation and Blockchain

With the ever-increasing pervasiveness of IoT, there is a necessity to create platforms that are both decentralized and private. The combination of SMPC with blockchain technology can be an important advance in this area, as it may be possible to create platforms that enable privacy-preserving (data remains encrypted even in-use) and are resilient. Although some issues remain unanswered, namely, is it possible to design a decentralized platform without completely relying on a trusted third party, or can one construct a fully decentralized protocol to the sell secret information without allowing sellers and buyer to cheat?

1.4.1 Applications

Enigma [22] combines the use of SMPC and blockchain technologies. In the next sections, we describe Enigma and associated use cases with real-world applications.

Enigma
Enigma is a peer-to-peer network that enables different parties to jointly store and run computations on data while keeping the data completely private. This model works in parallel with an external blockchain technology. Similar to Bitcoin, Enigma removes the need for a trusted third party.

The main motivation for this work lays in the avoidance of centralized architectures that might lead to catastrophic leakage of data that would result in the loss of privacy. Their approach is designed to connect to an existing blockchain and offload private and intensive computations to an off-chain network. Code is executed both on the blockchain (public tasks) and on Enigma (private and computationally intensive tasks). Opposing blockchain, which only ensures correctness in execution, Enigma is able to simultaneously provide privacy and correctness. One of its main features is its privacy-enforcing computation, as Enigma can execute code without data leakage while still ensuring correctness. Since heavy duty computations are a known issue for blockchains, Enigma avoids it by only allowing running computations to be broadcasted throughout the blockchain. While blockchains are not mean to be general-purpose databases they can be used to strategically store information. Enigma has a decentralized off-chain distributed hash-table that makes use of blockchains to store data references (not to the actual data). Nevertheless, private data must be encrypted on the client-side before storage and access-control protocols should be programmed into the blockchain, that will act as a public proof for

the authorization schema.

Enigma Application Cases

SMPC can be applied in some fields where privacy is a concern. In this section we describe some of the most relevant domains where we envision it can be applied.

Applicability in IoT seems rather straightforward, since we can store, manage and use highly sensitive data collected by IoT devices in a decentralized, trust-less cloud. The Crypto Bank is also a field where the internal details have to be anonymous, so, we can run a full-service crypto bank without exposing information about its internal design and implementation. The autonomous control of the blockchain allow users to take loans, deposit cryptocurrencies or buy investment products without publicly revealing their financial situation.

In line with the millionaire's problem, where n-parties want to know if they are more wealthier than the others, without exposing their financial status to each one, there is blind e-voting. In the latter case, not only the privacy of each voter is maintained but also the actual vote count can potentially remain private.

Another application of Enigma is on the N-factor Authentication, where voice, face and fingerprint recognition are all stored and computed on Enigma. As the access-control is supported by private contracts, only the user is able to access its own data.

Furthermore, private contracts are used when we want to securely share some data with a third party. We can define some policies in the contracts which restrict the access to data, maintaining and enforcing control and ownership. The shared data on MPC is always reversible, since third parties do not have access to actual raw data, and are restricted to only running secure computation over it. The identity management is also supported by private contracts, since when an user wants to log in, an authenticating private contract is executed in order to validate the user and to link to his real real identity with a public pseudo identity making this process completely trust-less and privacy-preserving. This way, the authentication and store identities is fully anonymous, and the user on Enigma only has to secret-share her personal information required for authentication.

For data protection, privacy preserving approaches should be paramount to companies, since they hold large volumes of potential sensitive user data that is a potential target for criminals. With Enigma, companies can use data to provide personalized services and match individual preferences without storing or processing the data on their servers, and thus removing the security and privacy risks. By doing so, companies are also protected against corporate espionage and rogue employees. It should be noted that employees can still use and analyse data for the benefit of the user while enforcing agreed consents. With these solutions in place, companies can potentially provide access to the data while preserving security and privacy.

A potential interesting case can be found in the the data marketplace, for example, a pharmaceutical company looking for patients for clinical trials can scan genomic databases for candidates. In this process, consumers can sell access to their data with guaranteed privacy, autonomous control and increased security. The marketplace would eliminate tremendous amounts of friction between companies and

individuals, lower costs for customer acquisition and offer a new income stream for consumers.

1.4.2 Future Research Directions

Enigma has yet to release any source code, as so, its off chain-network performance is still unknown. However, historically, computation on encrypted data has been slow in practice, and it remains an active research path [66].

Additionally, Enigma entails techniques for processing transactions without knowing their contents that might provide an alternative way to achieve similar accountability benefits while supporting transactions. However, this approach does not preclude the possibility of a validator favoring transactions based on a bias, because it can identify the transactions with help of colluding peers, even if the transactions are encrypted [67].

Enigma's novel combination of three paradigms, secret-sharing, MPC and P2P, opens new possibilities to address current open issues on data privacy and the growing liabilities faced by organizations that store or work on large amounts of personal data. However, until the official launch of Enigma's source code it is not possible to guess what problems will be solved by it [68].

1.5 Summary

This chapter has provided a summary of the multiple concepts of the secure computation in different approaches. The first sections (1.2 and 1.3) presented some concepts of secure computation such as secure Multi-Party Computation that consists in exchange data anonymously without a trusted third party, or blockchain that is a secured way of online transaction. We described the concepts and its applications, as well as the combination of both (if any) and/or with Internet of Things. In the secure Multi-Party Computation section, we describe the frameworks that we tested, as a way of analyzing the functionality of each one.

With this analysis, we discovered some interesting applications which shows that there are some attempts developments based on the combination of some of these concepts. The main finding applications were:

- Blockchain with IoT: IOTA and ADEPT (Section 1.2.1);
- Blockchain with secure Multi-Party Computation: Enigma MIT's (Section 1.4.1).

As we describe in the Section 1.2, there are several limitations in order to integrate the blockchain technology with IoT. Section 1.2.1 describes a solution that consists of using fog computing as a way of decreasing latency, having "local awareness" and mobility (including vehicular mobility). As the limitations of cloud com-

puting are undesirable for IoT, the integration between IoT and blockchain should be made using the fog computing with IoT.

However, there are unsolved problems and open issues yet, that we described as future research directions in the sections 1.3.3 and 1.4.2.

References

1. Twisted Matrix Labs: Building the engine of your Internet. (2016) http://twistedmatrix.com/trac/. Cited 25 October 2016
2. VIFF, the Virtual Ideal Functionality Framework. (2007) http://viff.dk/. Cited 25 October 2016
3. Network Flow Problems with Secure Multiparty Computation. PhD Thesis. Abdelrahaman Aly.
4. The General Multiprecision PYthon project (GMPY) (2008) https://wiki.python.org/moin/GmPy. Cited 25 October 2016
5. Damgrd, Ivan, et al. "Asynchronous multiparty computation: Theory and implementation." International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, 2009.
6. SPDZ Software. (2016) https://www.cs.bris.ac.uk/Research/CryptographySecurity/SPDZ/. Cited 25 October 2016
7. Lindell, Yehuda, et al. "Efficient constant round multi-party computation combining BMR and SPDZ." Annual Cryptology Conference. Springer Berlin Heidelberg, 2015.
8. Damgrd, Ivan, et al. "Practical covertly secure mpc for dishonest majorityor: Breaking the spdz limits." European Symposium on Research in Computer Security. Springer Berlin Heidelberg, 2013.
9. Damgrd, Ivan, et al. "Multiparty computation from somewhat homomorphic encryption." Advances in CryptologyCRYPTO 2012. Springer Berlin Heidelberg, 2012. 643-662.
10. Edwin, Njeru Mwendi. "Software frameworks, architectural and design patterns." Journal of Software Engineering and Applications 7.8 (2014): 670.
11. Bogdanov, Dan, Sven Laur, and Jan Willemson. "Sharemind: A framework for fast privacy-preserving computations." European Symposium on Research in Computer Security. Springer Berlin Heidelberg, 2008.
12. Sharemind SDK Beta. (2015) https://sharemind-sdk.github.io/. Cited 25 October 2016
13. Havron, Samuel. "Poster: Secure Multi-Party Computation as a Tool for Privacy-Preserving Data Analysis."
14. Nielsen, Jesper Buus. "Secure Multiparty Computation - Basic Technology, Past, Present, Future"
15. Andrychowicz, Marcin, et al. "Secure multiparty computations on bitcoin." 2014 IEEE Symposium on Security and Privacy. IEEE, 2014.
16. Rass, Stefan, Peter Schartner, and Monika Brodbeck. "Private function evaluation by local two-party computation." EURASIP Journal on Information Security 2015.1 (2015): 1-11.
17. Malkhi, Dahlia, et al. "Fairplay-Secure Two-Party Computation System." USENIX Security Symposium. Vol. 4. 2004.
18. Ben-David, Assaf, Noam Nisan, and Benny Pinkas. "FairplayMP: a system for secure multi-party computation." Proceedings of the 15th ACM conference on Computer and communications security. ACM, 2008.
19. SCAPI Documentation. (2014) https://scapi.readthedocs.io/en/latest/intro.html. Cited 30 October 2016
20. Ejgenberg, Yael, et al. "SCAPI: The Secure Computation Application Programming Interface." IACR Cryptology ePrint Archive 2012 (2012): 629.

21. Tapscott, Don, and Alex Tapscott. Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World. Penguin, 2016.
22. Zyskind, Guy, Oz Nathan, and Alex Pentland. "Enigma: Decentralized computation platform with guaranteed privacy." arXiv preprint arXiv:1506.03471 (2015).
23. Acquisti, Alessandro, et al., eds. Digital privacy: theory, technologies, and practices. CRC Press, 2007.
24. Vuksanovic, Irena Petrijevcanin, and Bojan Sudarevic. "Use of web application frameworks in the development of small applications." MIPRO, 2011 Proceedings of the 34th International Convention. IEEE, 2011.
25. Malka, Lior. "Vmcrypt: modular software architecture for scalable secure computation." Proceedings of the 18th ACM conference on Computer and communications security. ACM, 2011.
26. Yao, Andrew Chi-Chih. "How to generate and exchange secrets." Foundations of Computer Science, 1986., 27th Annual Symposium on. IEEE, 1986.
27. Shamir, Adi. "How to share a secret." Communications of the ACM 22.11 (1979): 612-613.
28. Bit Global Pay. (2016)
https://bitglobalpay.com/ Cited 9 November 2016
29. Jnsson, Kristjn Valur, Gunnar Kreitz, and Misbah Uddin. "Secure Multi-Party Sorting and Applications." IACR Cryptology ePrint Archive 2011 (2011): 122.
30. Henecka, Wilko, et al. "TASTY: tool for automating secure two-party computations." Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010.
31. Wyld, David C., Jan Zizka, and Dhinaharan Nagamalai. "Advances in computer science, engineering and applications." (2012).
32. Burkhart, Martin, et al. "Sepia: Security through private information aggregation." arXiv preprint arXiv:0903.4258 (2009).
33. Burkhart, M., Strasser, M., Many, D., Dimitropoulos, X.A.: SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics. In: USENIX Security Symposium, USENIX Association (2010) 223240
34. SEPIA - Security through Private Information Aggregation (2011)
http://sepia.ee.ethz.ch/ . Cited 10 November 2016
35. IOTA (201)
http://www.iotatoken.com/ . Cited 10 November 2016
36. IOTA: Internet of Things Without the Blockchain? (2016)
http://bitcoinist.net/iota-internet-things-without-blockchain/ . Cited 10 November 2016
37. Atzori, Marcella. "Blockchain-Based Architectures for the Internet of Things: A Survey." Browser Download This Paper (2016).
38. Yao, Andrew C. "Protocols for secure computations." Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. IEEE, 1982.
39. IOTA: Economy of Internet-of-Things (2016)
https://medium.com/DavidSonstebo/iota-97592581f985#.rhosuii7l . Cited 10 November 2016
40. Veena, Pureswaran, et al. "Empowering the Edge-Practical Insights on a Decentralized Internet of Things." Empowering the Edge-Practical Insights on a Decentralized Internet of Things.
41. Autonomous Decentralized Peer-to-Peer Telemetry (2015)
http://wiki.p2pfoundation.net/Autonomous_Decentralized_Peer-to-Peer_Telemetry . Cited 11 November 2016
42. Signorini, Matteo. "Towards an internet of trust: issues and solutions for identification and authentication in the internet of things." (2015).
43. Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
44. King, Sunny, and Scott Nadal. "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake." self-published paper, August 19 (2012).
45. TeleHash - Encrypted Mesh Protocol.
http://telehash.org/ . Cited 11 November 2016

46. IBM & Samsung live demo of ADEPT — TheProtocol.TV https://www.youtube.com/watch?v=U1XOPIqyP7A . Cited 11 November 2016
47. Ethereum - Homestead Release Blockchain App Platform https://www.ethereum.org/ . Cited 11 November 2016
48. Ron, Dorit, and Adi Shamir. "Quantitative analysis of the full bitcoin transaction graph." International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2013.
49. Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.
50. Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
51. Szabo, Nick. "The idea of smart contracts." Nick Szabos Papers and Concise Tutorials (1997).
52. Dorri, Ali, Salil S. Kanhere, and Raja Jurdak. "Blockchain in internet of things: Challenges and Solutions." arXiv preprint arXiv:1608.05187 (2016).
53. Blockchain Applications - What are Blockchain Technology Applications and Use Cases? http://www.blockchaintechnologies.com/blockchain-applications . Cited 14 November 2016
54. Perera, Charith, Chi Harold Liu, and Srimal Jayawardena. "The emerging internet of things marketplace from an industrial perspective: A survey." IEEE Transactions on Emerging Topics in Computing 3.4 (2015): 585-598.
55. Madisetti, Vijay, and Arshdeep Bahga. "Internet of things." (2014).
56. Edwin, Njeru Mwendi. "Software frameworks, architectural and design patterns." Journal of Software Engineering and Applications 7.8 (2014): 670.
57. Pullonen, Pille, and Sander Siim. "Combining secret sharing and garbled circuits for efficient private IEEE 754 floating-point computations." International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2015.
58. Chen, Peinan, Shruthi Narayanan, and Jeffrey Shen. "Using Secure MPC to Play Games."
59. Huang, Yan, et al. "Faster Secure Two-Party Computation Using Garbled Circuits." USENIX Security Symposium. Vol. 201. No. 1. 2011.
60. Dastjerdi, Amir Vahid, and Rajkumar Buyya. "Fog Computing: Helping the Internet of Things Realize Its Potential." Computer 49.8 (2016): 112-116.
61. Abdelshkour, Maher. IoT, from Cloud to Fog Computing. (2015) http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing . Cited 17 November 2016
62. Mrs. R. Waheetha, Mrs.Sowmya Fernandez. Fog Computing And Its Applications. (2016)
63. Peter, Nisha. "FOG Computing and Its Real Time Applications."
64. Privacy on the Blockchain. (2016) https://blog.ethereum.org/2016/01/15/privacy-on-the-blockchain/ . Cited 18 November 2016
65. Signorini, Matteo. "Towards an internet of trust: issues and solutions for identification and authentication in the internet of things." (2015).
66. Yuan, Ben, Wendy Lin, and Colin McDonnell. "Blockchains and electronic health records." Mcdonnell. mit. edu.
67. Herlihy, Maurice, and Mark Moir. "Enhancing accountability and trust in distributed ledgers." arXiv preprint arXiv:1606.07490 (2016).
68. Blockchain and Health IT: Algorithms, Privacy, and Data (2016) White Paper.
69. Preden, Jrgo S., et al. "The benefits of self-awareness and attention in fog and mist computing." IEEE Computer Magazine (2015).
70. What is IOTA? https://iota.readme.io/v1.1.0/docs . Cited 23 January 2017