Project N°: **FP7-284731**

Project Acronym: **UaESMC**

Project Title: **Usable and Efficient Secure Multiparty Computation**

Instrument: **Specific Targeted Research Project**

Scheme: **Information & Communication Technologies**
Future and Emerging Technologies (FET-Open)

Deliverable D3.4
Design of Protocols with a Mixture of Cryptographic and Game-Theoretic Assumptions

Due date of deliverable: 31st July 2015

Actual submission date: 31st July 2015

Start date of the project: **1st February 2012** Duration: **42 months**

Organisation name of lead contractor for this deliverable: **UoA**

Specific Targeted Research Project supported by the 7th Framework Programme of the EC		
Dissemination level		
PU	Public	✓
PP	Restricted to other programme participants (including Commission Services)	
RE	Restricted to a group specified by the consortium (including Commission Services)	
CO	Confidential, only for members of the consortium (including Commission Services)	

Executive Summary:
Design of Protocols with a Mixture of Cryptographic and Game-Theoretic Assumptions

This document summarizes deliverable D3.4 of project FP7-284731 (UaESMC), a Specific Targeted Research Project supported by the 7th Framework Programme of the EC within the FET-Open (Future and Emerging Technologies) scheme. Full information on this project, including the contents of this deliverable, is available online at `http://www.usable-security.eu`.

This report contains a high-level discussion and overview of the work done by the research team from the University of Athens during the last year of the project. This work is included in [26] and [25]. The scope of the deliverable is to study the effect of using game-theoretic tools in the SMC setting. Our final goal is to construct fair and robust secure multi-party protocols, i.e. protocols in which all parties learn the output or nobody does, and moreover, they guarantee output delivery and prevent denial of service attacks.

List of Authors

Yiannis Giannakopoulos (UoA)
Yiannis Tselekounis (UoA)

Contents

Chapter 1

The Bitcoin Protocol and SMC

1.1 Bitcoin

Bitcoin is a decentralized, peer-to-peer, payment system, introduced by Satoshi Nakamoto ([31]) in 2008, in which the users may perform transactions without the need of a central entity. The system is based on a public transaction ledger, called blockchain, implemented in a distributed way. The blockchain is maintained and extended by the participants, called miners. In order to generate new blocks for the blockchain, the miners need to solve a proof-of-work which requires to brute-force a hash inequality based on SHA-256. Concretely, the miners need to find a number, called nonce, such that when the block content is hashed along with the nonce, the output is numerically smaller than a target value defined by the network. The proof is easy to verify but not easily generated. This proof-of-work system, combined with the chaining of blocks, prevents attackers from modifying the block chain, since in order to modify a single block in the chain the attacker needs to modify all subsequent blocks. Each block in the blockchain consists of transactions that are generated by owners of bitcoins who issue transactions for entities of their choice that accept payments in bitcoin. Payers broadcast the set of transactions and miners incorporate the transactions they receive into their newly generated blocks. After block generation, the miner that generated the new block receives a predetermined amount of bitcoins. This is how bitcoins are created and distributed among miners.

As the Bitcoin protocol relies on the blockchain, a major concern is preventing attackers from manipulating, or reorganizing, the transaction ledger. As we discuss in the following sections, Bitcoin has many applications in SMC, and therefore, any attack that compromises the security of the blockchain, renders the application built upon it, insecure. In [31], Nakamoto provides a set of system parameters that guarantee security against attackers that try to manipulate the blockchain and his main argument is the following: assuming we have a transaction that transfers a specific amount of bitcoins to a payee, then, if the payee waits for the transaction to advance into the blockchain a number of k blocks, then the probability that an attacker succeeds in producing an alternative blockchain that reorganizes the public one, drops exponentially with k. This approach may be oversimplified, and moreover, it is completely out of the game-theoretic context, which seems to be mandatory for any construction (e.g., [26]) that models and builds SMC over Bitcoin, using game-theoretic tools. [25] is an ongoing work that aims to give a game-theoretic analysis of the Bitcoin protocol. One of the major concerns of [25] is to investigate the circumstances under which following the Bitcoin protocol is a Nash equilibrium.

1.2 The Effects of Selfish Behaviour

In this section we argue about the necessity of providing a thorough game-theoretic analysis of the Bitcoin protocol. In [12] and [28], it is stated that Bitcoin miners tend to behave strategically and form mining pools. This comes from the fact that rewards in Bitcoin are distributed at infrequent, random intervals, and miners form mining pools so as to receive their rewards in a more predictable rate. All pool participants contribute to the solution of the proof-of-work, and if the pool solves the current crypto-puzzle, the participants receive rewards proportionally to their contributions. Besides that, mining pools may receive slightly higher rewards since they achieve better coordination between participants due to lower network latency. In this setting, analyzing participants' incentives looks mandatory: are there any strategies that give higher payoff that the prescribed one? What if the attacker that controls a significant amount of participants, or a number of pools, tries to introduce disagreement between the honest miners so that it splits their hashing power on different proof-of-work instances?

[12] is the first work addressing such concerns. They show that the Bitcoin protocol is not incentive compatible, meaning that participants that do not disclose information related to their actions, as it is dictated by the protocol, they achieve higher payoff than following the protocol. The main argument of [12] is based on a strategy, called "Selfish Mining", that when followed by a minority pool it gives revenue which is higher than the one which is proportional to the total mining power of the pool. When a pool follows "Selfish Mining" it prefers to keep its newly discovered blocks private, and therefore, forking the chain in two chains. The public one, followed by the participants that are unaware of the new blocks (call that group of parties A), and the private one, followed by the pool. Whenever the pool creates a new block it keeps it private incrementing the distance from the public chain. On the other hand, when the public chain approaches the private one, selfish miners publicize blocks from the private chain obtaining payments for those blocks. As it is proved in [12], this strategy makes the parties in group A to waste resources on mining blocks that will never be part of the public block chain, and even if selfish miners waste resources, the parties in group A waste proportionally more. Therefore, the expected reward of selfish mining exceeds its share of the total network power and this fact gives incentives to follow the "Selfish Mining" strategy. The analysis made in [12] takes into account the size of the pool and the network propagation speed. For a mining pool with almost perfect connectivity the size of the pool may be close to zero, meaning that Bitcoin is safe only if 100% of the miners belong to group A. As the network latency increases, the hashing power of the pool needs to be bigger, reaching the lower bound of 1/3 over the total power, for the lowest level of connectivity. This is significantly lower than the 50% currently assumed in practice.

In order to simplify things, [12] assumes that miners split into two groups. The parties of group A, assumed to be the honest majority following the protocol, and the minority mining pool that follows the selfish mining strategy. This is the motivation for [25], which aims to provide a game-theoretic analysis of the Bitcoin protocol while assuming that we have one group of parties, which are all rational, i.e., they try to maximize their utility functions, and we assume that during initialization there is only a single block, the genesis block, being the target block for all parties. In other words, in [12] there is no assumption on the network state when the game starts. In this setting, parties define their mining and announcing strategies, meaning that they choose which block to mine, and moreover, they decide to publicize, or not, their newly created blocks. The strategy of every party depends on its private view (local view) in case there are blocks not published by the party, and the public view of the network which is the common view that all parties have. For a high level description of [12] we refer the reader to Chapter 4.

1.3 Circumventing Limitations in SMC

Secure multi-party computation (SMC), initiated by the seminal works of Yao [43] and Goldreich et al. [19], enables a set of parties to evaluate any function on their private inputs, while guaranteeing privacy on those inputs, i.e., no party learns anything about other parties' inputs besides what is being inferred by the output of the computation. Besides privacy, an SMC protocol is desirable to be fair, meaning that all parties learn the output or nobody does, and robust, meaning that it guarantees output delivery and prevents denial of service attacks.

From the mid-80s it was made clear that fairness is hard to achieve. The impossibility result of Cleve [10] showed that secure two-party computation for general functionalities is impossible for any protocol without assuming honest majority. Since then, there are many works studying various relaxations of fairness, like "partial fairness", both for the case of specific functionalities like coin tossing ([10, 11, 29]), as well as for general functionalities ([5, 7, 14, 15, 20, 21, 36]), and furthermore, may not require honest majority. The major drawback for those protocols is that they are highly inefficient.

In order to circumvent the impossibility results in fairness in the setting of dishonest majority, [1, 2, 6, 27, 26], introduced a new direction which is based on Bitcoin (and other cryptocurrencies), where fairness is guaranteed by imposing penalties to the parties in case of detection while cheating. By choosing the appropriate value for the penalty, the attacker is enforced to follow the protocol, and moreover, honest parties are guaranteed to learn the output, or receive compensation by the cheating party. The high level idea of [26] is given in Chapter 3.

Chapter 2

The Game-Theoretic Lens

Game Theory [42] can be described as the field of mathematics that tries to formally study, by analyzing and predicting, the behaviour of rational, selfish entities, especially under conflict of their preferences. This is a very general description, and through the years this field has evolved into on of the richest and more diverse contemporary areas of research, spanning many diverse disciplines such as Economics, Computer Science, Mathematics, Biology, Sociology, etc.. As such, various models of such interactions between competing agents have been developed, trying to capture different settings that arise from many different applications. In this chapter we will very briefly present only a few of the most important game-theoretic notions that inspired some of the results of the current deliverable D3.4.

2.1 (Non)cooperative Games

The "standard" notion of a *game* $\mathcal{G}$ [13, 35] consists of a finite set of *players* $N = \{1, 2, \dots, n\}$, and for each player $i \in N$ a set of *strategies* (or *actions*) S_i available to her together with a *utility* (or *payoff*) function $u_i : \prod_{i=1}^{n} S_i \longrightarrow \mathbb{R}$ over them. The intuitive interpretation is that, every possible *strategy profile* (or *outcome*) $\mathbf{s} = (s_1, s_2, \dots, s_n) \in S = \prod_i S_i$ of $\mathcal{G}$ results in an amount of "happiness" measured by $u_i(\mathbf{s})$ for each player i, and thus, being *fully rational and selfish*, each one of them will try to choose a strategy s_i that will maximize *her own* payoff. Usually we generalize that model by allowing the agents to play *mixed* strategies, i.e. assign probabilities on their pure strategies S_i instead of just deterministically committing to a single action $s_i \in S_i$. Denote the set of all such possible probability distributions over S_i by Σ_i. Then, naturally enough, the utility maximization is done under expectation over the randomization of the players.

The most celebrated and widely studied and *solution* concept (or *equilibrium*) of such games is that of *Nash equilibrium (NE)* [32]: a profile of (mixed) strategies $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n) \in \prod_{i=1}^{n} \Sigma_i$, one for each player, is a NE if no player has an incentive to *unilaterally deviate* from it. That is, if all other players $N \setminus \{i\}$ commit to their strategies σ_{-i}, player i maximizes his expected payoff by also committing to his strategy σ_i:

$$\mathbb{E}_{s \sim \sigma}\left[u_i(s_i, s_{-i})\right] \geq \mathbb{E}_{s_{-i} \sim \sigma_{-i}}\left[u_i(s_i', s_{-i})\right],$$

for all players $i \in N$ and pure strategy $s_i \in S_i$. The seminal result of Nash demonstrates how all games possess at least one NE, establishing it a natural and meaningful solution concept. More on traditional solution concepts of games can be found in our Deliverable D3.1 [41].

Notice how, in this standard approach to Game Theory, each player's behaviour is totally *individualistic*: deviations and utility-maximization are performed by the perspective of each agent separately, and no *coordination* of actions is assumed. However, in many settings this is not the most suitable or realistic assumption: we would like to capture and predict the outcome in models where the participants are willing to coordinate and deviate in *groups* in an effort to drive their utilities up. This is the main idea behind the important branch of Game Theory known as *Cooperative Game Theory* [4], which is essentially the theory of coalition formation.

The traditional notion of a NE can be then generalized in a natural way to that of *strong* NE, where no *group* of players have an incentive to collectively deviate and improve each one of the participants' utilities:

$$\mathbb{E}_{s\sim\sigma}\left[u_i(s_A, s_{-A})\right] \geq \mathbb{E}_{s_{-A}\sim\sigma_{-A}}\left[u_i(s'_A, s_{-A})\right],$$

for all possible coalitions $A \subseteq N$, every player $i \in A$ and pure strategies $s_A \in \prod_{j\in A} S_j$. However appealing that game solution notion may seem, it is actually very strong and restrictive and as such, very few games actually have such equilibria.

2.2 Fairness

Perhaps a more natural way of approaching the study of cooperative games is through the assumption of *transferable utilities*. In such models, every group of players $A \subseteq N$ is associated with some joint value $v(A)$; think of that as a combined value that this group A can generate for its members. Then, the critical question is how this value is going to be split among its members in a *"fair"* way. For example, a natural way to capture deviations would be that of associating *share* x_i to each player $i \in N$ such that the system is *balanced*, i.e. $\sum_{i=1}^{n} x_i = c(N)$ and then require our solutions v to be such that no coalition can gain by deviating, that is

$$\sum_{i\in A} x_i \geq v(A),$$

for all groups $A \subseteq N$. This is the important notion of a *core* [40, 38] of a cooperative game.

A different but well studied notion of fairness is that of the *Shapley value* [39]: fix an ordering of the set of players $N = (1, 2, \dots, n)$ and define $N_i = (1, 2, \dots, i)$ for all i. Then, the *marginal* value of the i'th player is $v(N_i) - v(N_{i-1})$. Notice how this of course depends on the particular initial ordering of N which we have fixed. Then the Shapley value sets the value-share x_i of a player i to be the *expected* value of her marginal value, where the expectation is taken uniformly over the set of possible orderings of N. An important, characteristic property of the Shapley value is that it can be formally proved to be the only value-share function $x = (x_1, x_2, \dots, x_n)$ that satisfies the following desirable properties: *anonymity* (it does not depend on the actual labels/names of the players), *dummy* (if a player does not contribute to the joint value v then she should not be given any share) and *additivity* (x is an additive function of the joint-value v). Finally, let us briefly mention here that an other, widely studied solution concept of cooperative games in the Economics literature is that of the *Nash Bargaining* [33]. For an extensive discussion on these matters of fairness we suggest [23].

These notions of group-deviation resistance, fairness and cooperative games have inspired our work in the current deliverable, and we tried to shed new light in standard, cryptographic views to Bitcoin computation and functionality by approaching it in a novel, game-theoretic way (see Chapter 4). Also, the use of modern cryptocurrencies allow for a more flexible platform for implementing value-shares, monetary compensations and payoffs for the players, and we expect that to evolve into a fruitful area of research in the coming years (see Chapter 3)

2.3 Payments and Mechanism Design

Although games seem to capture the fundamental notion of strategic interaction between players, the models so far are totally *passive* from the perspective of the designer: the system somehow reaches a stable state, an equilibrium point, which is an internal property of the game and we can have no control over it at all. But for many problems we are interested in studying, there is some external designer who wants to create protocols that implement some desired objective, for example minimization of the total social cost or the aggregation of the preferences of the members of the society in a single outcome, e.g. voting. This means that we would like to have a way of designing *rules* for the games in a way that the previous, passive convergence to a stable state will coincide with an outcome that satisfies our design needs. This is essentially the subject matter of the entire area of Mechanism Design [30, 34], which for that reason sometimes is being referred to as *reverse game theory*. More on Mechanism Design can also be found in our Deliverable D3.2 [17].

A key aspect of Mechanism Design is the requirement that our protocols should be *truthful*: they should give no incentive to the players to lie about their true preferences and manipulate the mechanism. Monetary transfers and payments play an essential part in the implementation of such mechanisms, such as auctions, scheduling rules and bandwidth allocation in networks. This is formally supported by the celebrated impossibility results by Arrow [3] and Gibbard-Satterthwaite [18, 37], that essentially state that without payments the only meaningful social choice rules that can be implemented are *dictatorships*: there is always going to be an agent that can manipulate the outcome just by switching her own preference [16]. These results laid the way to the foundations of a formal and powerful theory of implementing collective preferences by the use of monetary transfers. Under this light, our work in the context of the current deliverable (see Chapter 3) tries to provide a formal framework for monetary transfers in SMC settings, deploying cryptocurrencies, that bypasses analogous seminal impossibility results in cryptography [10].

Chapter 3

SMC Through Bitcoin Using Compensations

Beyond standard privacy features, a secure multi-party computation (SMC) protocol is highly desirable to be also *fair* (either all parties learn output or none) and *robust* (the delivery of the output is guaranteed and the adversary cannot mount a "denial of service" against the protocol). Achieving fairness and robustness in a setting where there is an arbitrary number of possible corruptions, as ideal as it may appear, is prohibited by strong impossibility results stemming from the work of Cleve [10] who showed that coin-flipping is actually infeasible in any setting where there is no honest majority among parties that execute the protocol. These impossibility results, combined with the importance of the properties that they prevent, strongly motivate the exploration of alternative —yet still realistic— models that would enable fair and robust SMC protocols.

With the advent of Bitcoin [31] and other decentralized cryptocurrencies, a new direction was developed [2, 1, 6, 27] for circumvention of the impossibility results regarding the fairness property: enforcing fairness could be achieved through imposing *monetary penalties*. In this setting a breach of fairness by the adversary is still possible but it results in the honest parties collecting a compensation in a way that is determined by the protocol execution. At the same time, in case fairness is not breached, it is guaranteed that no party loses any money (despite the fact that currency transfers may have taken place between the parties). The rationale here is that a suitable monetary penalty suffices in most practical scenarios to force the adversary to operate in the protocol fairly.

The above constitute an important step for the design of SMC protocols with properties that circumvent the classical impossibility results. However, several critical open questions remain to be tackled and that was the goal of the authors of [26] in the context of the current deliverable D3.4. The key elements of this work are listed below:

- A new formal model that utilizes two ideal functionalities and expresses the ledger of transactions and a clock in the sense of [24] that is connected to the ledger and enables parties to synchronize their protocol interactions. Our ledger functionality enable us to abstract all the necessary features of the underlying cryptocurrency. Contrary to the only previous formalization approach [6, 27] our modelling allows the entities that participate in a SMC execution to be regular interactive Turing machines (ITM) and there is no need to equip them with additional physical features such as "safe" and "locks". Furthermore, the explicit inclusion of the clock functionality (which is only suggested in [6, 27]) reveals the exact dependencies between these two functionalities that are necessary in order for SMC with compensation protocols to be properly described. We express our model within a general framework that we call Q-*fairness* and may be of independent interest as it can express meaningful relaxations of fairness in the presence of a global ideal functionality.

- A composition theorem that shows that our protocol realizes SMC in a universally composable fashion. Our composition proof treats the clock and ledger functionalities as global setups in the sense of [8, 9]. We emphasize that this is a critical design choice: the fact that the ledger is a global functionality ensures that any penalties that are incurred to the adversary that result to credits towards the honest parties will be globally recognized. This should be contrasted to an approach that utilizes regular ideal functionalities which may be only accessible within the scope of a single protocol instance and hence any penalty bookkeeping they account may vanish with the completion of the protocol. Providing a composition theorem for SMC protocols with compensation was left as an open question in [6].

- A new protocol for fair and robust SMC with compensation. The robustness property we prove guarantees that once the protocol passes an initial round of deposits, parties are guaranteed to obtain output or be compensated. This is in contrast to fair SMC with compensation [2, 1, 6, 27] where the guarantee is that compensation takes place only in case the adversary obtains output while an honest party does not. To put it differently, it is feasible for the adversary to lead the protocol to a deadlock where no party receives output however the honest parties have wasted resources by introducing transactions in the ledger (as posting transactions in the ledger requires a fee that is collected by the miners). We remark that it is in principle possible to upgrade the protocols of [2, 1, 6, 27] to the robust SMC setting by having them perform an SMC with identifiable abort [22] (in such protocol the party that causes the abort can be identified and excluded from future executions). However even using such protocol the resulting robust SMC with compensation will need in the worst case a linear number of deposit/communication rounds in the number of malicious parties. Contrary to that our robust protocol requires a constant number of deposit/communication rounds independently of the number of parties that are running the protocol. Our construction uses time-locked transactions in a novel way to ensure that parties do progress in the SMC protocol or otherwise transactions are suitably revertible to a compensation for the remaining parties. The structure of our transactions is quite more complex than what can be presently supported by bitcoin; we provide a high level overview of how our protocol can be implemented via Ethereum3[1] contracts.

[1] http://www.ethereum.org

Chapter 4

Bitcoin Under Rationality

In this chapter we provide the high level idea behind [25], which aims, as a first step, to provide a game-theoretic abstraction of the Bitcoin protocol. As it was discussed in section 1.2, [25] tries to avoid strong assumptions on the incentives of the participants and its final goal is to investigate the circumstances under which the Bitcoin protocol is a Nash equilibrium. Since Bitcoin is a quite complex protocol, providing the formal game, as well as analysing it, are quite involved and challenging procedures.

The ideas in [25] are preliminary and its structure goes as follows. First we define the Bitcoin as a game using a graph based approach (Definition 1), and then, using a grid based approach (Definition 7). The two definitions are almost equivalent, but the later one seems to be more intuitive. Informally, we consider an n-party, lottery-based, game Γ_{T,k^*}, which is defined concretely in Definition 1 and is parametrized by T, $k^* \in \mathbb{N}$. The game proceeds in rounds and each round corresponds to a specific time interval such that for every $i \in [n]$, there is a set, V_i, of active lotteries which are visible to player P_i. In the first round there is only one active lottery and for all rounds there is a fixed set of players $\{P_1, \ldots, P_n\}$, $n \in \mathbb{N}$. In each round of the game the parties decide about their betting and announcing strategies. During the betting stage, each party P_i, $i \in [n]$, may allocate x_i^j tickets from lottery L_j, where $\sum_{\{j \mid L_j \in V_i\}} x_i^j \leq x_i$, and x_i is defined right before the game starts and denotes the budget of player i. The total number of tickets for each lottery is $T \in \mathbb{N}$, and we assume that $\sum_{i=1}^{n} x_i^j << T$, i.e., for each lottery L_j there is a number of tickets which remains unallocated and there might be no winner with probability $(T - \sum_{i=1}^{n} x_i^j)/T$. The winning probability of player P_i in lottery L_j is x_i^j/T, and if P_i wins in L_j, a new lottery $L_{j'}$ is created and P_i is being informed about his win, while no other party does. It is P_i who decides to inform, or not, the rest of the participants about the newly created lottery. This mechanism creates a set of directed trees $\{G_1, \ldots, G_n\}$, where each $G_i = (V_i, E_i)$ consists of lotteries visible to party P_i; if P_i wins in L_j, a new lottery (vertex) $L_{j'}$ is created and the edge $(L_j, L_{j'})$ is added to E_i. In addition, if P_i announces its win on L_j, then the node $L_{j'}$, as well as the edge $(L_j, L_{j'})$, are being incorporated into the private graphs of all parties; the graph $G = (V, E)$ that consists of the lotteries that have been publicly announced is being updated accordingly. The game terminates when the depth of G becomes k^*, i.e., when we have a publicly known directed path of length k^*. Let $\bar{V} = \{L_1, \ldots, L_k\}$ be the set of lotteries, public or not, created during the game, and let $\{L_{w_1}, \ldots, L_{w_{k^*}}\}$ be the publicly known set of lotteries that forms a directed path of length k^*. Assuming the game terminates, each party P_i receives payoff $c \cdot \sum_{j=1}^{k^*} W_{P_i}^{w_j} - \sum_{j=1}^{k} \bar{x}_i^j$, where c is a constant reward given to the winner, the predicate W_P^j is equal to 1 if and only if player p wins lottery L_j and $L_j \in V$, and $\bar{x}_i^j$ denotes the total investment of party P_i in lottery L_j.

In the description given above, lotteries correspond to Bitcoin blocks, the parties need to decide which blocks (lotteries) to mine, and in case of wining, they need to decide if they publicizing the new block, or not. We prove (Lemma 9) that if all parties follow the strategy that corresponds to the strategy indicated by the Bitcoin protocol, i.e., they always announce their wins and they bet on the last lottery of the longest path of the public lottery graph, they receive payoff which is proportional to their investments (or hashing power). This property is mandatory for any model that aims to capture the properties of the Bitcoin protocol. Lemma 10 is the first attempt towards analyzing parties' strategies. We prove that for finite games in which all parties have the same budget $x << T$, and the graph is of depth two, choosing not to announce a winning lottery yields lower payoff than the strategy that does. Currently, [25] is in the process of analyzing games of bounded depth, say k^*, bigger than 2, and the analysis seems to be quite complex, since it needs to consider all different strategies for announcing and betting, for all graphs of depth k^*.

Bibliography

[1] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek. Secure multiparty computations on bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 443–458, May 2014.

[2] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. Fair two-party computations via bitcoin deposits. In Rainer Böhme, Michael Brenner, Tyler Moore, and Matthew Smith, editors, *Financial Cryptography and Data Security*, volume 8438 of *Lecture Notes in Computer Science*, pages 105–121. Springer Berlin Heidelberg, 2014.

[3] K.J. Arrow. *Social Choice and Individual Values.* Yale University Press, 1951.

[4] Robert J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67 – 96, 1974.

[5] Donald Beaver and Shaft Goldwasser. Multiparty computation with faulty majority. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO 89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 589–590. Springer New York, 1990.

[6] Iddo Bentov and Ranjit Kumaresan. How to use bitcoin to design fair protocols. In JuanA. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014*, volume 8617 of *Lecture Notes in Computer Science*, pages 421–439. Springer Berlin Heidelberg, 2014.

[7] Dan Boneh and Moni Naor. Timed commitments. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 236–254. Springer Berlin Heidelberg, 2000.

[8] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In *Theory of Cryptography*, pages 61–85. Springer, 2007.

[9] Ran Canetti, Abhishek Jain, and Alessandra Scafuro. Practical uc security with a global random oracle. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 597–608, New York, NY, USA, 2014. ACM.

[10] R Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, pages 364–369, New York, NY, USA, 1986. ACM.

[11] Richard Cleve. Controlled gradual disclosure schemes for random bits and their applications. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO 89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 573–588. Springer New York, 1990.

[12] Ittay Eyal and EminGün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, volume 8437 of *Lecture Notes in Computer Science*, pages 436–454. Springer Berlin Heidelberg, 2014.

[13] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, Cambridge, MA, 1991.

[14] Zvi Galil, Stuart Haber, and Moti Yung. Cryptographic computation: Secure fault-tolerant protocols and the public-key model (extended abstract). In Carl Pomerance, editor, *Advances in Cryptology - CRYPTO 87*, volume 293 of *Lecture Notes in Computer Science*, pages 135–155. Springer Berlin Heidelberg, 1988.

[15] Juan Garay, Philip MacKenzie, Manoj Prabhakaran, and Ke Yang. Resource fairness and composability of cryptographic protocols. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 404–428. Springer Berlin Heidelberg, 2006.

[16] J. Geanakoplos. Three brief proofs of arrow's impossibility theorem. *Economic Theory*, 26(1):211–215, 2005.

[17] Yiannis Giannakopoulos. Strongly Truthful and Composable Mechanism Design, January 2013. UaESMC Deliverable 3.2.

[18] A. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41(4):587–601, 1973.

[19] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all np statements in zero-knowledge and a methodology of cryptographic protocol design (extended abstract). In AndrewM. Odlyzko, editor, *Advances in Cryptology - CRYPTO 86*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185. Springer Berlin Heidelberg, 1987.

[20] Shafi Goldwasser and Leonid Levin. Fair computation of general functions in presence of immoral majority. In AlfredJ. Menezes and ScottA. Vanstone, editors, *Advances in Cryptology-CRYPT0 90*, volume 537 of *Lecture Notes in Computer Science*, pages 77–93. Springer Berlin Heidelberg, 1991.

[21] S.Dov Gordon and Jonathan Katz. Partial fairness in secure two-party computation. *Journal of Cryptology*, 25(1):14–40, 2012.

[22] Yuval Ishai, Rafail Ostrovsky, and Vassilis Zikas. Secure multi-party computation with identifiable abort. In *Advances in Cryptology–CRYPTO 2014*, pages 369–386. Springer, 2014.

[23] Kamal Jain and Mohammad Mahdian. Cost sharing. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*, chapter 15. Cambridge University Press, 2007.

[24] Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Universally composable synchronous computation. In Amit Sahai, editor, *Theory of Cryptography*, volume 7785 of *Lecture Notes in Computer Science*, pages 477–498. Springer Berlin Heidelberg, 2013.

[25] Aggelos Kiayias, Elias Koutsoupias, and Yiannis Tselekounis. Bitcoin: a game-theoretic approach. Unublished report, 2015.

[26] Aggelos Kiayias, Hong-Sheng Zhou, and Vassilis Zikas. Fair and robust multi-party computation using a global transaction ledger. Cryptology ePrint Archive, Report 2015/574, 2015. `http://eprint.iacr.org/`.

[27] Ranjit Kumaresan and Iddo Bentov. How to use bitcoin to incentivize correct computations. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 30–41, New York, NY, USA, 2014. ACM.

[28] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S. Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 919–927, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.

[29] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. In Omer Reingold, editor, *Theory of Cryptography*, volume 5444 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2009.

[30] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[31] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[32] J.F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[33] Jr. Nash, John F. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[34] Noam Nisan. Introduction to mechanism design (for computer scientists). In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*, chapter 9. Cambridge University Press, 2007.

[35] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[36] Benny Pinkas. Fair secure two-party computation. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 87–105. Springer Berlin Heidelberg, 2003.

[37] M.A. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[38] Herbert E. Scarf. The core of an n person game. *Econometrica*, 35(1):pp. 50–69, 1967.

[39] Lloyd S Shapley. A value for n-person games. 1952.

[40] Lloyd S. Shapley. On balanced sets and cores. *Naval Research Logistics Quarterly*, 14(4):453–460, 1967.

[41] Yiannis Tselekounis and Yiannis Giannakopoulos. Potential Uses of SMC in Game Playing and Mechanism Design, January 2013. UaESMC Deliverable 3.1.

[42] John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 3rd edition, 1953.

[43] Andrew C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.