

Equitable Cake Cutting without Mediator

Sophie Mawet, Olivier Pereira and Christophe Petit

Université catholique de Louvain
ICTEAM – Crypto Group
B-1348 Louvain-la-Neuve, Belgium

Abstract. We consider enhancing solutions to the fair division problem of cake cutting by proposing a cryptographic protocol preserving the privacy of the players’ preferences in each step. This addresses an important shortcoming of traditional division procedures, that disclose some of the players’ preferences and offer other players the possibility to dynamically adjust their behaviour in order to obtain unfair advantages. By contrast, the only thing that players learn when applying our procedure is strictly minimal, that is, players only learn the cutting point that would be determined by a fair mediator. Our approach relies on the description of the players’ utility function through step functions, a choice that enables us to obtain an efficient procedure for the secure evaluation of cutting points in the two-player case. We also explore a procedure that could be applied for more players.

Keywords: Secure multiparty computation, Game theory

1 Introduction

1.1 The cake cutting problem

Cake cutting is a game-theoretic problem of fair division. It consists in dividing a heterogeneous good so that all parties believe they have received a fair share. Beyond the metaphor of cutting a cake, fair divisions are advised in a wide variety of situations: to determine the borders in an international dispute, to divide goods in an inheritance or to split the costs of building a shared road among the different users. The problem is interesting when players value the parts of the cake differently. This potentially allows the k players to receive more than $1/k$ of the value of the cake according to their own preferences.

Different criteria of fairness can be used to divide a cake. A *proportional* division guarantees each of the k players to receive at least $1/k$ of the cake according to his valuation. A division is *envy-free* if the resource is divided in such a way that no one will prefer another player’s share. If all the players get the same proportion according to their valuation, the division is called *equitable*.

Game theory provides fair divisions in a lot of interesting cases. The basic one with two players is “I cut, you choose”. Alice cuts the cake into two parts she thinks equal and Bob chooses the part with the greatest utility according to his preferences. This procedure is not equitable and, like all the game theoretic procedures, suffers from a lack of secrecy: Bob gets a lot of information about Alice’s preferences, since Alice cuts the cake into two halves she considers exactly equal. However, Alice does not know how far Bob prefers the share he has chosen to the other one.

More complex game theoretic solutions exist for an infinitely divisible cake as well as for indivisible goods [BT96]. There are two kinds of procedures: discrete and continuous moving-knife procedures. Brams and Taylor [BT95] give an unbounded discrete procedure for an envy-free division between k players. Brams, Taylor and Zwicker [BTZ97] describe a moving-knife procedure for an envy-free division between four players. Like “I cut, you choose”, these procedures reveal a lot of information about the players’ preferences but this disclosure of information is even more annoying, as it enables the players to adapt their strategy in order to obtain a bigger piece of cake.

The presence of a mediator improves and simplifies fair divisions. The players can privately communicate their preferences to him and he can then suggest a fair division satisfying all parties. A mediated division reveals nothing about the players’ preferences except for the information that can be deduced from the cutting points. Furthermore, such a division potentially leads to much higher payoffs, that is, the players receive a bigger piece of cake according to their own valuation. A trusted mediator is, however, not very easy to find and his existence is unsafe. A person who knows the preferences of each party will focus all attacks on himself.

1.2 Goal of introducing secure multiparty computation techniques

In cryptography, players can emulate a mediator by running a protocol themselves. They can interact with each other, under proper assumptions, in such a way they learn the cutting point a mediator would have computed and nothing more. The cryptographic protocols rely on secure multiparty computation. SMC is the problem of k players who want to compute an agreed function of their inputs in a secure way. Security implies correctness of the outputs and privacy of the inputs, even when some parties cheat. In concrete terms, there are $k \geq 2$ players P_1, \dots, P_k , where P_i knows his input x_i . The players want to compute $f(x_1, \dots, x_k) = y_i$ so that P_i learns the output y_i and nothing more, except for information that can be deduced from (x_i, y_i) .

The adversary \mathcal{A} stands for the set of cheating players. In other words, an adversary may corrupt a subset of the players. Once corrupted, a player gives the adversary his entire history, that is, the complete information on all actions and messages he has received so far. There are different kinds of corruptions. A *honest-but-curious* adversary can read all the data of the corrupted players but he cannot modify their behaviour: a corrupted player still executes the protocol correctly. A *malicious* adversary takes full control of the corrupted players who may deviate arbitrarily from the protocol. Between these two extremes, a *covert* adversary may behave maliciously but a sufficiently large probability of being caught cheating will prevent him acting in such a way. We use the honest-but-curious model, nevertheless generic techniques such as those proposed by Damgård, Geisler and Nielsen [DGN10] enable to turn our protocol into a secure protocol against covert attacks.

We work in the *information theoretic model* of communication introduced by Ben-Or, Goldwasser and Wigderson [BGW88] and Chaum, Crépeau and Damgård [CCD88]. In the information theoretic (I.T.) model, communication channels are supposed to be pairwise secure, that is, the adversary gets no information at all about messages exchanged between honest players. Security can then be guaranteed even when the adversary has unbounded computing power. We assume that communication is synchronous in the I.T. model. A protocol proceeds in rounds: in every round, each player may send a message to each other player, and all messages are delivered before the next round begins.

As explained in [CD06], a protocol cannot be secure if any subset of the k players can be corrupted. Limitations must be specified on the subsets the adversary can corrupt. If the adversary is allowed to corrupt all subsets of the parties of size at most t for some $t < k$, then it is called a threshold adversary and t is called the threshold. In the information theoretic model and with a honest-but-curious adversary, unconditional security is possible if at most $t < k/2$ players are corrupted [BGW88, CCD88].

Secure multiparty computation can typically be divided into three phases. First, each player P_i secretly shares his input x_i between the k players (including himself). Now, all players have a secret share of the input x_i of each other player. Second, each player runs a protocol with, as inputs, the k shares he has received. The goal is to compute the function $y = f(x_1, \dots, x_k)$ securely without a trusted party. Each player's output is a share of the secret value y . Third, the shares can be used for further computation or revealed to the players, who can then reconstruct y .

A secure multiparty protocol leaks no information during its execution. The players can thus not adapt their preferences and the protocol leads to a more equitable division. We work under the settings of a continuous cake. The cutting points have therefore non integer values. This is a difficulty because SMC deals with discrete values.

1.3 Contribution

We give a secure multiparty protocol for an equitable cake cutting without a mediator. Contrarily to the game theoretic procedures, this solution preserves the privacy of the preferences and leads to the high payoffs of a mediated division. This protocol is more efficient than generic techniques and, therefore, usable in practice. There is no known procedure in game theory to achieve, without a mediator, an equitable division of a heterogeneous cake with uncut pieces. This new result is possible by using SMC to emulate a mediator.

First, we introduce step functions to model the players' preferences because they are well appropriate to the restrictions inherent to the secure multiparty protocols. Furthermore, step functions appear to approximate arbitrarily closely any reasonable utility function. This class of functions enables to write an equitable cake allocation procedure for SMC.

Second, the cake allocation procedure is used in a secure multiparty protocol. The cake cutting protocol relies on a secure protocol for comparison [DFK06] and on a secure protocol for inversion of non integer values [ACS02]. The properties of the cake cutting protocol depend on the properties of these building blocks.

1.4 Related work

The SMC techniques are useful for many applications, in particular for those involving different participants with conflicting interests who want to keep their data confidential. This is the case for elections and for auctions. A medical survey is another example where people are often reluctant to share their sensitive data.

In their paper, Bogetoft *et al.* [BDJ⁺06] give an implementation of secure auctions for practical real-world problems. They address the problem of double auctions of a single divisible commodity with multiple sellers and buyers. The participants face the same confidentiality issue as in the cake cutting problem. The disclosure of an individual bid may be of great interest for the other buyers, who can use the revealed information to adapt their strategy, not only for the current auction but also for upcoming ones. Miltersen, Nielsen and Triandopoulos [MNT09] propose a rational cryptographic protocol for a single item auction.

Branching programs are commonly used to model diagnostic and classification algorithms with applications in areas such as health care, fault diagnostic or benchmarking. Barni *et al.* [BFK⁺09] present efficient privacy-protecting protocols for remote evaluation of such algorithms. They apply their protocols to the secure classification of medical ElectroCardioGram (ECG) signals.

Automatic recognition of human faces is another topic that raises important privacy issues, for example when a client privately searches for a specific face image in the database of a server. Sadeghi, Schneider and Wehrenberg [SSW09] propose a privacy-preserving face recognition scheme and give implementation results that show the practicality of their solution even for large databases.

2 Modelling the cake cutting problem

2.1 Modelling the players' preferences

To achieve a mediated equitable cake division, the players have to express their preferences in a digital way and to agree on a function that, given their preferences as input, computes the cutting points as output. Each player explicitly tells how much he values the different parts of the cake through a utility function defined on the “cake interval”. We represent the cake by a given interval of \mathbb{R} . To get a simple analytic expression of the cutting points, utility functions cannot be general.

First, utility functions were modelled by polynomials. Linear functions made it possible to obtain an envy-free and equitable division for two players. However, as soon as the degrees of the polynomials increased, it became quite difficult to obtain the analytic expression of the cutting points. Cake cutting being more interesting when players have more degrees of liberty to express their preferences, we propose another modelling of the players' preferences.

We express the players' preferences for the different pieces of cake by a step function. This is a convenient choice: it is very easy to calculate the integral of a step function and to approximate more general functions by step functions. To get a simple analytic expression, we add other assumptions. All the steps of the function should have the same width and an integer value. The cake should be valued on $[0, N]$ with N equal to the number of steps of the step function (the steps have a width of 1). Adapting the size of the cake to the number of steps of the function makes it possible to deal with integer values for the integration: the integral of a utility function on a unitary interval is an integer value. This simplifies the integration and since secure comparison is possible, it is not a problem to proceed by parts.

No one is favoured: all players have the same total utility for the cake. Let $U \in \mathbb{R}^+$ be the utility of a player for the whole cake. Moreover, all the utilities for each interval are supposed to be strictly positive integers.

2.2 Equitable division for two players

In this section, we focus on determining the functions that two players have to evaluate securely in order to find the cutting point. The inputs of a player are the values of his N -step function on each interval. The output corresponds to the cutting point. Equitability is required for the sharing. Every player gets exactly the same proportion of the cake according to his own valuation.

Let α be the cutting point. The utility function of the first player (P_1) is defined by $f_1(x) = u_1$ on $[0, 1]$, $f_1(x) = u_2$ on $]1, 2]$, \dots , $f_1(x) = u_N$ on $]N - 1, N]$. In the same way, the utility function of the second player (P_2) is defined by $f_2(x) = v_1$ on $[0, 1]$, $f_2(x) = v_2$ on $]1, 2]$, \dots , $f_2(x) = v_N$ on $]N - 1, N]$. Equitability implies that

$$\int_0^\alpha f_1(x) dx = \int_\alpha^N f_2(x) dx.$$

The left piece of cake is arbitrarily allocated to P_1 and vice-versa for P_2 . This is not a problem since the division is equitable and there are only two players. Either both players are happy with their shares, or they both receive less than a half of the cake (according to their own valuation) and they envy each other. In this case, they exchange their parts. The equitable division for two players is therefore also proportional and envy-free.

Proposition 1. *There is a unique cutting point for two players.*

Proof. There are two different orderings of the two players. Let us take the ordering $P_1 - P_2$. Define the functions $g_1(\alpha) := \int_0^\alpha f_1(x) dx$ and $g_2(\alpha) := \int_\alpha^N f_2(x) dx$. Since f_1 and f_2 are bounded, g_1 and g_2 must be continuous. Now, since $g_1(0) = 0$ and $g_1(N) > 0$ while $g_2(0) > 0$ and $g_2(N) = 0$, there must be a value $\alpha \in [0, N]$ such that $g_1(\alpha) = g_2(\alpha)$. Moreover, since f_1 and f_2 are strictly positive functions, g_1 and g_2 must be strictly monotonic, and the point α satisfying $g_1(x) = g_2(x)$ must be unique.

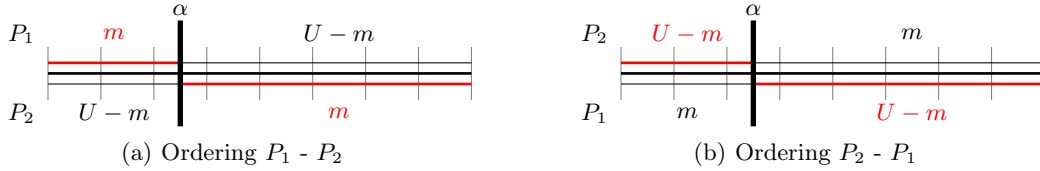


Fig. 1. Equitable divisions for the two different players' orderings

Since $g_1(N) = g_2(0)$, the cutting point α also gives an equitable division for the ordering $P_2 - P_1$ (with a utility $U - m$ for each player) and since we know that the equitable division is unique, we also find the equitable division corresponding to the ordering $P_2 - P_1$. \square

Proposition 2. *There always exist an equitable, envy-free and proportional division for two players.*

Proof. There are two orderings of the players leading to two different divisions. One division gives a utility of m and the other of $U - m$. At least one division leads therefore to a utility of at least $U/2$, giving a proportional and envy-free division. \square

Where is the cutting point? The players have to evaluate securely the following equation for $\alpha \in]i - 1, i]$:

$$u_1 + \dots + u_{i-1} + (\alpha_i - (i - 1))u_i = (i - \alpha_i)v_i + v_{i+1} + \dots + v_N \quad (1)$$

Equation 1 checks whether $\alpha \in]i - 1, i]$ with $i \in \mathbb{Z}$ and $1 \leq i \leq N$. If $\alpha_i \in]i - 1, i]$, it is the exact cutting point ($\alpha = \alpha_i$). The solution of the equation is

$$\alpha_i = \frac{-(u_1 + \dots + u_{i-1} - (i - 1)u_i) + (i v_i + v_{i+1} + \dots + v_N)}{(u_i + v_i)}.$$

In concrete terms, the function the players have to evaluate securely for $]i - 1, i]$ is

$$F_i(u_1, \dots, u_N, v_1, \dots, v_N) = b_i a_i^{-1}, \text{ with}$$

$$a_i = u_i + v_i$$

$$b_i = - \sum_{m=1}^i u_m + \sum_{m=i+1}^N v_m + i (u_i + v_i).$$

Dichotomic search through the intervals. The function $F_i(u_1, \dots, u_N, v_1, \dots, v_N)$ is computed for each of the N intervals until the cutting point is found. In the worst case, N equations must be solved. We are going to solve partially this efficiency issue. If $\alpha_i \notin]i - 1, i]$, it is not the exact cutting point but an estimation of α , which is all the more precise that $]i - 1, i]$ is close to α , thus $\alpha_i > i$ implies $\alpha > i$ and $\alpha_i < i - 1$ implies $\alpha < i - 1$. This makes it possible to reduce the number of iterations through a dichotomic search algorithm. In the worst case $\log_2 N$ equations are solved. Let us show that $\alpha_i > i$ implies $\alpha > i$.

Proof. Suppose that $\alpha_i > i$ ($\alpha_i = i + \epsilon$ with $\epsilon > 0$), then the following equalities follow from Equation 1:

$$u_1 + \dots + u_{i-1} + ((i + \epsilon) - (i - 1))u_i = (i - (i + \epsilon))v_i + v_{i+1} + \dots + v_N$$

$$u_1 + \dots + u_{i-1} + (\epsilon + 1)u_i = (-\epsilon)v_i + v_{i+1} + \dots + v_N$$

$$\underbrace{u_1 + \dots + u_i}_{\int_0^i f_1(x)dx} + \epsilon(u_i + v_i) = \underbrace{v_{i+1} + \dots + v_N}_{\int_i^N f_2(x)dx}.$$

The last equality implies that

$$\int_0^i f_1(x)dx < \int_i^N f_2(x)dx,$$

which directly implies that $\alpha > i$ since $\int_0^\alpha f_1(x)dx = \int_\alpha^N f_2(x)dx$ with $f_1(x) > 0$ and $f_2(x) > 0$, for all x .

With the dichotomic search algorithm, the computation efficiency is improved by a factor of $\log_2 N$. However, contrarily to the exhaustive search that enables to check all intervals in parallel, the dichotomic search requires to do the computation sequentially.

2.3 Equitable division for k players

With three or more players, the equitability criterion is no longer sufficient to guarantee an envy-free or even proportional cutting. Indeed, it is not always possible to get such a division by letting the players simply exchange their respective pieces. Further, the more the number of players increases, the more difficult it is to solve a system for each combination of all cutting points in all intervals. As illustrated in Figure 2, there are $k - 1$ cutting points $(\alpha, \beta, \gamma, \dots)$.

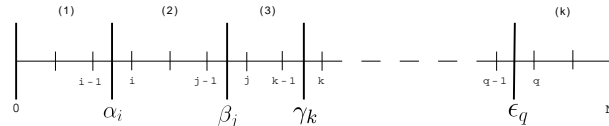


Fig. 2. The $k - 1$ cutting points

The linear equation for two players (Equation 1) can be generalized by a linear system of equations for three or more players (System 2). This system expresses that the value the player P_1 gives to the first part of the cake has to be equal to the value the player P_2 gives to the second part, that this value has to be equal to the value the player P_3 gives to the third part and so on. Other systems are possible to achieve an equitable division. This choice was made to obtain a system with a tridiagonal matrix that is easily factorized.

The matrix A of the initial system is invertible. This can be deduced from the determinant of the matrix L and U . The determinant of L is equal to 1 and the determinant of U is equal to $d'_1 \cdot d'_2 \cdot \dots \cdot d'_n > 0$. Let us show that the determinant of U is strictly positive.

Proof. The matrix of the initial system is tridiagonal with strictly positive elements on its main diagonal and strictly negative elements on the secondary diagonals thus $d_i > 0$ for $i = 1 : n$, $c_i < 0$ for $i = 2 : n$ and $e_i < 0$ for $i = 1 : n - 1$.

$$d'_1 = d_1 > 0 \tag{3}$$

$$d'_i = \frac{d_i d'_{i-1} - c_i e_{i-1}}{d'_{i-1}} \tag{4}$$

Equation 4 shows that if $d'_{i-1} > 0$, then $d'_i > 0$. By recurrence, we have $d'_i > 0$ for $i = 1 : n$ and thus $d'_1 \cdot d'_2 \cdot \dots \cdot d'_n > 0$.

The determinant of L and U are both strictly positive, which directly implies that the determinant of A is strictly positive, that the matrix is invertible and finally that the system has a unique solution.

2.4 Practical application

In this section we give an example for which the used definition of cake cutting makes sense. Suppose the case of two doctors who have to share a day on duty. Each of them gives his preferences for each hour (let's say between 6 am and 9 pm). Since this situation is likely to be repeated, they wish to keep their preferences private.

As each doctor wants to work as less as possible, this situation is actually the inverse of the classical cake cutting problem. Each doctor distributes a total utility of 100 over the 15 intervals (hours) and gives a small utility for an interval he is interested in and a large utility for one he is not interested in. At the end each doctor takes the shift with the smallest utility. The work shift has to be continuous. In Section 4, we give some efficiency results.

	6 am	7 am	8 am	9 am	10 am	11 am	12 am	1 pm	2 pm	3 pm	4 pm	5 pm	6 pm	7 pm	8 pm
Doctor 1	15	7	4	1	1	1	15	4	4	4	7	7	10	10	10
Doctor 2	20	10	10	7	7	4	4	4	1	1	4	7	7	7	7

3 Secure equitable cake cutting

3.1 Definitions and building blocks

Let q be a prime and let $l \in \mathbb{Z}$ be a complexity parameter with $l := \lceil \log_2 q \rceil$. We assume that we have a linear secret sharing scheme like that proposed by Shamir [Sha79] with a multiplication protocol (MUL). The notation $[x]_q$ represents a share of x over \mathbb{Z}_q and the notation $[u_m]_q$ stands for $[u_1]_q, \dots, [u_N]_q$.

The information theoretic model of communication is used. However, a secure channel is useless with only two players. Any protocol will leak some information in the information-theoretic sense [BGW88]. For example, the multiplication protocol of Gennaro, Rabin and Rabin [GRR98] requires the presence of at least three players. A third partly-trusted player is required. He is not a mediator in the strict sense because he does not know the sensible data: the utilities u_m and v_m . He gets only shares of these values. So, the two players need to rely less extensively on his integrity. The third partly-trusted player has to follow the protocol like any other player. We work in the honest-but-curious model. The adversary can control up to $k_c = \lfloor (k-1)/2 \rfloor = 1$ player. The corrupted player follows the protocol but tries to learn as much as possible about the inputs of the other parties.

The two building blocks of the cake cutting protocol are the bit decomposition protocol and the approximate inversion protocol. The bit decomposition protocol [DFK⁺06] is the key tool to compare two shared secrets securely. It computes the bit-decomposition $a_0, \dots, a_{l-1} \in \{0, 1\}$ of $a = \sum_{i=0}^{l-1} a_i 2^i$ with $a \in \mathbb{F}_q$. The total complexity is 114 rounds and $110l \log_2 l + 118l$ invocations of the multiplication protocol. The approximate inversion protocol [ACS02] distributively computes a

floating point approximation of $1/p$ using the Newton iteration. Inputs are polynomial shares of p and outputs are polynomial shares of an integer \tilde{p} so that $\tilde{p}/2^{t+l} = 1/p + \epsilon$, where $0 < \tilde{p} < 2^{t+2}$ and $|\epsilon| < (k+1)2^{-l-t+4}$. To have the r most significant bits of $1/p$ and $\tilde{p}/2^{t+l}$ equal, the parameter t must be chosen bigger than $r + 5 + \log_2(k+1)$ [ACS02]. The total complexity is $O(\log_2 t)$ rounds and $O(\log_2 t)$ invocations of the multiplication protocol.

3.2 Protocol for two players

We present a secure protocol to compute an equitable division for two players that relies on the procedure described in Section 2.2. The protocol `INTERVAL` determines the interval in which the cutting point is included. Then the protocol `CUTTING-POINT` computes the cutting point α .

Protocol <code>INTERVAL</code> ($[u_m]_q, [v_m]_q, c, d$)
Each party executes the following steps:
1. $i = \lceil \frac{c+d}{2} \rceil$
2. $([a]_q, [b]_q) \leftarrow \text{COEFFICIENTS}([u_m]_q, [v_m]_q, i)$
3. $[g]_q \leftarrow \text{LOCMUL}(i-1, [a]_q)$
4. $[h]_q \leftarrow \text{LOCMUL}(i, [a]_q)$
5. $[g]_B \leftarrow \text{BITS}([g]_q)$
6. $[h]_B \leftarrow \text{BITS}([h]_q)$
7. $[b]_B \leftarrow \text{BITS}([b]_q)$
8. $[x]_q \leftarrow \text{BIT-LT}([b]_B, [h]_B)$
9. $[y]_q \leftarrow \text{BIT-LT}([g]_B, [b]_B)$
10. $x \leftarrow \text{REVEAL}([x]_q)$
11. $y \leftarrow \text{REVEAL}([y]_q)$
12. If $x + y = 2$ then <code>CUTTING-POINT</code> ($[a]_q, [b]_q$). Stop.
13. If $x = 1$ then <code>INTERVAL</code> ($[u_m]_q, [v_m]_q, c, i-1$)
14. If $y = 1$ then <code>INTERVAL</code> ($[u_m]_q, [v_m]_q, i, d$)

The protocol `INTERVAL` is recursively called up to $\log_2(N)$ times. Step 2 determines the coefficients $a_i = u_i + v_i$ and $b_i = -\sum_{m=1}^i u_m + \sum_{m=i+1}^N v_m + i(u_i + v_i)$ for the interval $]i-1, i]$. To check if the cutting point is in this interval, we test whether $b_i \in](i-1)a_i, ia_i]$, which is equivalent to test whether $b_i/a_i \in]i-1, i]$ and does not require a secure division. Step 3 to 9 perform a secure comparison of the shared coefficient b_i with the bound of the interval multiplied by a_i . Protocols from [DFK⁺06] enable unconditionally secure constant round multiparty computation for comparison. The comparison function is $\overset{?}{<}: \mathbb{Z}_q \times \mathbb{Z}_q \rightarrow \mathbb{Z}_q$, where $(x \overset{?}{<} y) \in \{0, 1\}$ and $(x \overset{?}{<} y) = 1$ iff $x < y$.

The complexities of the protocols `BIT-LT` and `BITS` are respectively 19 rounds and $22l$ invocations of the multiplication protocol and 114 rounds and $110l \log_2 l + 118l$ invocations of the multiplication protocol [DFK⁺06]. The protocol `INTERVAL` uses 3 invocations of `BITS` and 2 invocations of `BIT-LT` giving a total complexity of 133 rounds and $330l \log_2 l + 398l$ invocations of `MUL`.

Let us consider the case of 2 players and a third partly-trusted player ($k=3$) with a utility function of $N=10$ steps of length $\ell_x = 4$, where we only want to find the correct interval, that is, we do not use the `CUTTING-POINT` protocol. In the worst case, the protocol `INTERVAL` is called $\lceil \log_2 N \rceil = 4$ times giving a complexity of 532 rounds and 16928 invocations of the multiplication protocol.

Protocol COEFFICIENTS($[u_m]_q, [v_m]_q, i$)

Each player executes the following steps locally:

1. $[a]_q \leftarrow \text{ADD}([u_i]_q, [v_i]_q)$
2. $[p]_q \leftarrow \text{LOCMUL}(i, [a]_q)$
3. For $m = 1$ to i run $[-u_m]_q \leftarrow \text{LOCMUL}(-1, [u_m]_q)$
4. $[b]_q \leftarrow \text{ADD}([-u_1]_q, \dots, [-u_i]_q, [v_{i+1}]_q, \dots, [v_N]_q, [p]_q)$
5. Output $([a]_q, [b]_q)$

The protocol **COEFFICIENTS**($[u_m]_q, [v_m]_q, i$) computes the coefficients a_i and b_i securely. This is straightforward since u_m and v_m are polynomial shared for all m . The protocol does not require any communication between the players. Multiplication mod q of a shared element and a known element of \mathbb{Z}_q (**LOCMUL**) is achieved by having all parties locally multiply (mod q) the known element by their shares.

Protocol CUTTING-POINT($[a]_q, [b]_q$)

The two players and the third partly-trusted player execute the following steps:

1. $[\tilde{a}]_q \leftarrow \text{APPINV}([a]_q)$
2. $[\tilde{\alpha}]_q \leftarrow \text{APPMUL}([\tilde{a}]_q, [b]_q)$
3. $\tilde{\alpha} \leftarrow \text{REVEAL}([\tilde{\alpha}]_q)$

Once the correct interval found, the protocol **CUTTING-POINT** computes the function $\alpha_i = F_i(u_m, v_m) = b_i a_i^{-1}$. The protocol **APPINV**($[a]_q$) [ACS02] inverts the polynomial shared coefficient a . It is iterated $O(\log_2 t)$ times and one iteration requires 12 rounds and 2 invocations of **MUL**. Finally, **APPMUL**($[\tilde{a}]_q, [b]_q$) computes shares of an approximation to α_i . It requires 6 rounds and 1 invocation of **MUL**. The protocol **CUTTING-POINT** deals only with integers. The “tilde” variables approximate the following values:

Variable	Approximated value
\tilde{a}	$2^{t+l}/a$
\tilde{b}	$b/2^r$
$\tilde{\alpha}_i = \tilde{a} \cdot \tilde{b}$	$\alpha_i \cdot 2^{t+l-r}$

Let us go back to the example of 2 players and a 10-step utility function. We would like to compute the complexity of the **CUTTING-POINT** protocol. To have the $r = 4$ most significant bits of $1/a$ and $\tilde{a}/2^{t+l}$ equal, the parameter t must be chosen bigger than $r + 5 + \log_2(k + 1)$. The protocol **APPINV** is thus iterated at least $\lceil \log_2 7 \rceil = 3$ times and its total complexity is 37 rounds and 6 invocations of **MUL**. The total complexity of the **CUTTING-POINT** protocol is 43 rounds and 7 invocations of **MUL**. The total complexity of the **INTERVAL** protocol (with the computation of the cutting point) is approximately 700 rounds and 17000 invocations of the multiplication protocol.

3.3 Protocol for k players

While there was a unique cutting point with two players, there are $k!$ distinct vectors of cutting points leading to a different equitable division with k players. Indeed, the $k!$ possible orderings of the players conduct each to a different system of equations that has a unique solution since the matrix A is always invertible (Proof is in Section 2.3).

The protocol to compute an equitable division for a chosen players ordering is quite similar to the protocol for two players. Nevertheless, the issue is that an equitable division achieved through an arbitrary ordering has no reason to be proportional. In other words, the k players have no guarantee to receive at least $1/k$ of the cake according to their valuation.

Computing securely the most efficient division among all equitable divisions for k players would then require to compare the outcome of the $k!$ divisions resulting from the players' permutations, which quickly becomes very expensive. Moreover, it is not sure whether the most efficient division is proportional.

4 Implementation results

The protocol for two players is implemented thanks to the Virtual Ideal Functionality Framework (<http://viff.dk>) developed by Martin Geisler [Gei10] at the Aarhus University. Viff is an efficient and high-level framework to define practical secure multiparty computation applications. It is written in Python which provides a good library for asynchronous communications. While Viff can be used for applications in the passive as well as in the active model, we only implemented our protocol with security against a passive adversary.

Let us go back to the application of two doctors on duty who wish to determine their respective shift (Section 2.4). A straightforward implementation runs in about 20 seconds (with 7 seconds dedicated to the division needed for computing the final cutting point). The result is $\alpha = 6.684$, which means that the two shifts are respectively from 6 am to 12:40 am and from 12:40 am to 9 pm. The first doctor has a utility of 39,3 for the first shift and a utility of 60,7 for the second one and vice-versa for the other doctor, so the first doctor will take the morning shift while the second one will take the afternoon shift.

5 Open problems

The cutting points of an equitable division reveal a precise information about the players' preferences. Equitability requires that all the players receive exactly the same proportion of the cake according to their own valuation. An envy-free division does not reveal such an exact information. Envy-freeness means that any player prefers his own share to all the other ones. It could be interesting to explore this criterion, which has not been developed here. An interesting perspective would be to use Su's algorithm [Su99], which relies on a combinatorial result known as Sperner's lemma. This infinite algorithm constructs an envy-free division by successive approximations [Str07].

There are two directions to bridge cryptography and game theory. We developed only one direction: designing a new cake allocation procedure to solve the game theoretic problem of fair division under the cryptographic settings. The other direction is to apply game theory to cryptography. In the game theoretic settings, all players are supposed to be rational, that is, acting in their own and best interest. In this case, Shamir scheme is no longer relevant. An interesting perspective would be to use the concept of rational secret sharing as developed in [HT04] and [DGK06].

6 Acknowledgements

Sophie Mawet is funded by the Belgian Interuniversity Attraction Pole P6/26 BCRYPT. Olivier Pereira is a Research Associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). Christophe Petit is a Postdoctoral Researcher of the Belgian Fund for Scientific Research (F.R.S.-FNRS).

References

- ACS02. J. Algesheimer, J. Camenisch and V. Shoup. Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 417-432, Springer, 2002.
- BDJ⁺06. P. Bogetoft, I. Damgård, T. Jakobsen, K. Nielsen, J. Pagter and T. Toft. A practical implementation of secure auctions based on multiparty integer computation. In *Financial Cryptography and Data Security*, volume 4107 of *LNCS*, pages 142-147, Springer, 2006.
- BFK⁺09. M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A.-R. Sadeghi and T. Schneider. Secure evaluation of private linear branching programs with medical applications. In *14th European Symposium on Research in Computer Security - ESORICS 2009* of *LNCS*, Springer, 2009.
- BGW88. M. Ben-Or, S. Goldwasser and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *STOC 88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 1-10, ACM Press, 1988.
- BIB89. J. Bar-Ilan and D. Beaver. Non-cryptographic fault-tolerant computing in a constant number of rounds interaction. In *PODC 89: Proceedings of the eighth annual ACM Symposium on Principles of distributed computing*, pages 201-209, ACM Press, 1989.

- BT95. S.J. Brams and A. Taylor. An envy-free cake division protocol. In *American Mathematical Monthly*, volume 102, pages 9-18, Mathematical Association of America, 1995.
- BT96. S.J. Brams and A.D. Taylor. Fair division, from cake-cutting to dispute resolution. Cambridge University Press, 1996.
- BTZ97. S. Brams, A.D. Taylor and W. Zwicker. A moving-knife solution to the four-person envy-free cake-division problem. In *Proceedings of the american mathematical society*, volume 125, pages 547-554, 1997.
- Cat06. D. Catalano. Efficient distributed computation modulo a shared secret. In *Contemporary Cryptology of Advanced Courses in Mathematics - CRM Barcelona*, pages 1-39, Birkhäuser, 2006.
- CCD88. D. Chaum, C. Crépeau and I. Damgård. Multi-party unconditionally secure protocols. In *STOC 88 : Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 11-19, ACM Press, 1988
- CD01. R. Cramer and I. Damgård. Secure distributed linear algebra in a constant number of rounds. In *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *LNCS*, pages 119-136, Springer, 2001.
- CD06. R. Cramer and I. Damgård. Multiparty computation, an introduction. In *Contemporary Cryptology of Advanced Courses in Mathematics - CRM Barcelona*, pages 41-87, Birkhäuser, 2006.
- DFK⁺06. I. Damgård, M. Fitzi, E. Kiltz, J.B. Nielsen and T. Toft. Unconditionally secure constant round multi-party computation for equality, comparison, bits and exponentiation. In *Theory of Cryptography*, volume 3876 of *LNCS*, pages 285-304, Springer, 2006.
- DFN⁺05. I. Damgård, M. Fitzi, J.B. Nielsen and T. Toft. How to split a shared secret into shared bits in constant-round. In *Cryptology ePrint Archive*, Report 2005/140.
- DGN10. I. Damgård, M. Geisler and J.B. Nielsen. From passive to covert security at low cost. In *Theory of Cryptography*, volume 5978 of *LNCS*, pages 128-145, Springer, 2010.
- DHR00. Y. Dodis, S. Halevi and T. Rabin. A cryptographic solution to a game theoretic problem. In *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *LNCS*, pages 112-130, Springer, 2000.
- DR07. Y. Dodis and T. Rabin. Cryptography and game theory. In *Algorithmic Game Theory*, Cambridge University Press, 2007
- DGK06. S. Dov Gordon and J. Katz. Rational secret sharing, revisited. In *Security and Cryptography for Networks*, volume 4116 of *LNCS*, pages 229-241, Springer, 2006.
- Gei10. M. Geisler. Cryptographic protocols: theory and implementation. PhD thesis, Aarhus University Denmark, Department of Computer Science, February 2010.
- GMW87. O. Goldreich, S. Micali and A. Wigderson. How to play any mental game. In *STOC 87 : Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218-229, ACM Press, 1987.
- GRR98. R. Gennaro, M.O. Rabin and T. Rabin. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *PODC 98 : Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 101-111, ACM Press, 1998.
- HT04. J. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In *STOC 04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 623-632, ACM Press, 2004.
- Kat08. J. Katz. Bridging game theory and cryptography : recent results and future directions. In *Theory of Cryptography*, volume 4948 of *LNCS*, pages 251-272, Springer, 2008.
- MNT09. P.B. Miltersen, J.B. Nielsen and N. Triandopoulos. Privacy-enhancing first-price auctions using rational cryptography. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 541-558, Springer, 2009.
- Sha79. A. Shamir. How to share a secret. In *Communications of the ACM*, volume 22, pages 612-613, 1979.
- Str07. W. Stromquist. Envy-free cake divisions cannot be found by finite protocols. In *Fair Division*, Dagstuhl Seminar Proceedings, 2007.
- Su99. F.E. Su. Rental harmony: Sperner's lemma in fair division. In *The American mathematical monthly*, volume 106, pages 930-942, Mathematical Association of America, 1999.
- SSW09. A.-R. Sadeghi, T. Schneider and I. Wehrenberg. Efficient privacy-preserving face recognition. In *12th International Conference on Information Security and Cryptology - ICISC 2009*, volume 5984 of *LNCS*, pages 229-244, Springer, 2009.
- Yao86. A.C. Yao. How to generate and exchange secrets. In *SFCS 86 : Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 162-167, ACM Press, 1986.