Ann Oper Res (2018) 260:217–231
https://doi.org/10.1007/s10479-016-2139-y

S.I.: ADVANCES OF OR IN COMMODITIES AND FINANCIAL MODELLING

A game-theoretical and cryptographical approach to crypto-cloud computing and its economical and financial aspects

Barış Bülent Kırlar[1,2] · **Serap Ergün**[3] · **Sırma Zeynep Alparslan Gök**[1,2] · **Gerhard-Wilhelm Weber**[2]

Published online: 16 February 2016

Abstract Recently, crypto-cloud computing has become an interesting research area with many technical, security, commercial and financial aspects, goals and consequences. The original intention of the cloud is to link computing stations (services) for collaboration. Sharing and coordination of computational resources is important because of the activities between service providers and service requesters. Considering the cooperative functionality of crypto-cloud computing, the use of game theory in that area has became very beneficial. In the sequel, we mathematically associate that area with game theory, i.e., the bargaining and compromising of interests of various "players", by using a game-theoretical approach which arises from networks, servers, operating systems, storage devices, etc. Further, we propose a novel efficient encryption system by using XTR (effective and compact subgroup trace representation) which has the property of semantic security. Those interactions has been constructed in the direction of how cryptographic tools can be used to address a natural optimization problem in the fields of game theory and financial economics. It is believed that game theory and its optimization is going to provide a suitable framework for the design of a crypto-cloud computing system that will be perceived as a strong technique and satisfy the needs of many participants and users of the cloud. The paper ends with a conclusion and an outlook to future studies.

✉ Sırma Zeynep Alparslan Gök
zeynepalparslan@yahoo.com

Barış Bülent Kırlar
bariskirlar@sdu.edu.tr; barisbkirlar@gmail.com

Serap Ergün
serapbakioglu@sdu.edu.tr

Gerhard-Wilhelm Weber
gweber@metu.edu.tr

1 Department of Mathematics, Süleyman Demirel University, Isparta, Turkey

2 Institute of Applied Mathematics, METU, Ankara, Turkey

3 Department of Electronic and Communication Engineering, Süleyman Demirel University, Isparta, Turkey

Keywords Game-theoretical models · Optimization · Cloud computing · XTR · Trace-DLP · Cooperation

1 Introduction

Game theory has a long and efficient history in the literature of economics, mathematics, operations research (OR), and decision making applications. In recent years, game theory brought a new lease to computer science. Computer scientists have found that game-theoretical concepts and methods are very relevant to their problems (cf. Roberts 2008). In this investigation, we study on crypto-cloud computing within the framework of cooperative game theory.

Cloud computing presents an increasingly popular model for businesses in the financial services industry. Banks, stock brokerages, money management firms and other financial entities are attracted to the cost savings that cloud infrastructures can provide (cf. Intel IT Center 2013). Cloud computing comes along with its share of challenges, in terms of security, data privacy, compliance, availability, lack of standards, etc. These challenges are highlighted more in regulated and security-sensitive environments, such as financial services. A financial services firm that heavily relies on information-technology enabled services, can benefit from cloud computing (cf. Garg 2011).

In the case of computing security service providers, the question of cooperation is probably more relevant than in many other fields. Indeed, due to the interactions among users, low security provided by a competitor induces a risk for its own customers and, therefore, a lower security level. Coalition formation can thus become efficient for providers, in terms of reputation and revenue. Hence, it is very interesting to model and investigate the incentive to forming such coalitions, and whether or not a full cooperation is the best solution for all providers (cf. Maillé et al. 2011).

The research areas of game theory and cryptography are extensively studied fields with many problems and solutions; cloud computing is getting investigated. Yet, the cross-over between them has been surprisingly small until now. To the best of our knowledge, the use of cryptography in cloud computing and employment of game theory (non-cooperative) in cloud computing can be rarely seen. We believe that our study is a pioneering work in the area. To the best of our knowledge, no research has been done before by using cryptography in cloud computing and employing cooperative game theory.

The interrelation between OR and cooperative game theory is of a more recent date and is summarized under the heading of OR games. An important part of the interplay between cooperative games and OR systems is given by the basic structures of a graph, a network or a system which are underlying various types of combinatorial optimization problems. If one assumes that at least two players are located at or control parts of the system underlying, then a cooperative game can be associated with this type of optimization problem. By working together, the players can possibly create extra gains or save costs, compared to the situation in which everybody optimizes individually. Hence, the question arises how to share the extra revenues or cost savings. One way to analyze this question is to study the general properties of all games arising from that specific type of an OR problem and to apply a suitable game-theoretical solution concept to this class. Another way is to create a context-specific allocation rule. Such a rule can be based on desirable properties in this specific context or on a kind of decentralised mechanism that prescribes an allocation on the basis of an algorithmic process along with a jointly optimal combinatorial structure established (e.g., following an algorithm to create a minimal-cost spanning tree) (Borm et al. 2001).

A related situation takes place in the presence of a group of agents, each of which needs to be connected (directly or via links to other agents) to a source. Suppose that the construction

of the links is costly. Then, the first important question is how to find the cheapest set of links that will connect each agent to the source. This question constitutes one of the most well-known problems of combinatorial optimization: the minimum-cost spanning tree (mcst) problem. OR literature on mcst problems has provided many algorithmic solutions to the problem and has discussed the computational properties of these solutions. We can mention, for example, the two most famous algorithms, the Kruskal algorithm (1956) and the Prim algorithm (1957). A historic overview on these algorithms provided for the mcst problem can be found in Graham and Hell (1985) (for details see Ciftci and Tijs 2007).

These kind of cost allocation problems may arise on many different physical networks such as telephone lines, highways, electric power systems, computer chips, water delivery systems, rail lines, etc. On the other hand, to retrieve the information needed to assess the exact cost of all the links of a real network is a very hard task (Moretti et al. 2011).

Constructing an mcst, however, is only one part of the problem. In addition to minimizing total costs, a cost allocation problem has to be addressed as well. Claus and Kleitman (1973) introduce this cost allocation problem, whereupon Bird (1976) treats this problem with game-theoretical methods and proposes a cost allocation rule, known as the Bird rule (Borm et al. 2001).

On the other hand, introduced and characterized for cooperative games with a finite player set and where coalitions values are real numbers, the Shapley value (1953) has captured much attention and been extended with new game-theoretical models and widely applied for solving reward/cost sharing problems in OR and economic situations, sociology, computer science, etc. (cf. Gök et al. 2010).

In game-theoretical models for computational tasks in which many computers have to communicate and cooperate, we need to be careful about the extent to which we can carry out computations in this kind of distributed setting in a reliable and secure way (cf. Roberts 2008).

From the cryptographical point of view, the studies in cryptography move into the discussion of a mathematical approach with the contribution of Diffie and Hellman from the middle of 1970s. The security of public key cryptographic schemes depends on the hardness of some computational problems in mathematics: integer factorization problem, discrete logarithm problem (DLP) in cyclic groups, etc. DLP means to find x such that $g^x = h$ for given $g, h \in G$, where $(G, \cdot)$ is an Abelian group. By Akyıldız and Ashraf (2014), proposing a novel cryptographic protocols based on DLP and its versions is an open problem. In this sense, a new concept, the so-called trace-based cryptography over finite fields, can be considered. Trace-based cryptographic schemes use a oneway trace map and have a great advantage for the transmission size. Since the trace map is homomorphic with respect to the addition, trace-based cryptographic schemes can be used efficiently in crypto-cloud computing systems.

There are several works in trace-based cryptography surveyed in Akyıldız and Ashraf (2014) and Kırlar (2012). The leading one is XTR (Effective and Compact Subgroup Trace Representation), proposed by Lenstra and Verheul (2000). XTR is based on different representation of elements in the subgroup of order dividing $p^2 - p + 1$ in the finite field $\mathbb{F}^*_{p^6} = \mathbb{F}_{p^6} \backslash \{0\}$. The security of trace-based cryptographic schemes depends on the hardness of the trace discrete logarithm problem (Trace-DLP) and its different versions. For many other studies about trace-based cryptographic protocols, one can refer to Gong and Harn (1999), Stam and Lenstra (2001) and Ashraf and Kırlar (2014).

The organization of the paper is as follows. In Sect. 2, we give some basics from graph theory, minimum-cost spanning tree situations and algorithms that are applied for solving them and some solution concepts of cooperative game theory. The XTR cryptosystem is

introduced in a comprehensive manner and a novel encryption is proposed which is based on the concepts of XTR in Sect. 3. We present our crypto-cloud computing model in Sect. 4. Finally, Sect. 5 concludes this paper with an outlook to future studies.

2 Preliminaries

An (undirected) *graph* is a pair $\langle V, E\rangle$, where V is a set of vertices or nodes and E is a set of edges e of the form $\{i, j\}$ with $i, j \in V, i \neq j$. The *complete graph* on a set V of vertices is the graph $\langle V, E_V\rangle$, where $E_V = \{\{i, j\} | i, j \in V \text{ and } i \neq j\}$. A *path* between i and j in a graph $\langle V, E\rangle$ is a sequence of nodes $i = i_0, i_1, \ldots, i_k = j, k \geq 1$, such that all the edges $\{i_s, i_{s+1}\} \in E$, for $s \in \{0, \ldots, k-1\}$, are distinct. A *cycle* in $\langle V, E\rangle$ is a path from i to i for some $i \in V$. Two nodes $i, j \in V$ are *connected* in $\langle V, E\rangle$ if $i = j$ or if there exists a path between i and j in $\langle V, E\rangle$. A *connected component* of V in a graph $\langle V, E\rangle$ is a maximal subset of V with the property that any two nodes in this subset are connected in $\langle V, E\rangle$ (Evans and Minieka 1992; West 2001).

A *minimum-cost spanning tree* (*mcst*) *situation* is a situation, where $N = \{1, 2, \ldots, n\}$ is a set of agents who are willing to be connected as cheaply as possible to a source (i.e., a supplier of a service) denoted by 0, based on a cost (or weight) function (Moretti et al. 2011).

Algorithm 1 Bird's Rule (1976)

Require: an mcst problem $(N, *, t)$.
Ensure: an edge set $R \subset E_{N*}$ of an mcst and corresponding Bird allocation $\beta^R(\Gamma)$
1: Choose the source $*$ as root.
2: Initialize $R = \emptyset$.
3: Find a minimal cost edge $e = \{i, j\} \in E_{N*} \backslash R$ incident on $*$ or any of the vertices present in one of the edges in R in such a way that joining e to R does not introduce a cycle.
4: One of i and j, say j, was previously connected to the source and the other vertex i is a player who was not yet connected to the source. Assign the cost $\beta_i^R(T) \mid= t(e)$ to agent i.
5: Join e to R.
6: If not all vertices are connected to the root in the graph $\left(N^*, R\right)$, go back to, step 3.

Given an mcst problem, Bird's tree allocation, $\beta^R(T)$, is constructed by assigning to each player $i \in N$ the cost of the first edge on the unique path in from player i to the player source. The computation of this allocation can be integrated into the Prim algorithm which, starting from a fixed root, constructs an mcst by consecutively adding edges with the lowest cost, without introducing cycles (Borm et al. 2001).

The Prim algorithm can be described as follows: in every iteration of the Prim algorithm, a player who is not connected yet with the source constructs an edge between him/herself and either the source or another player which is already connected with the source in the previous iterations of the algorithm. Hence, the Prim algorithm is also vertex oriented. Moreover, similar to our algorithm, every player has the right to construct the cheapest allowed edge. But, the main difference is that the set of edges allowed for construction by the Prim algorithm is restricted to the ones which provide a connection with the source. The Bird rule assigns the cost of an edge constructed in some iteration of the Prim algorithm to the player which constructs that edge and gets a connection with the source in that same iteration (cf. Ciftci and Tijs 2007).

A cooperative cost game in characteristic function form is an ordered pair $\langle N, c\rangle$ consisting of the player set N and the characteristic function $c : 2^N \to \mathbb{R}$ with $c(\emptyset) = 0$. The real number $c(S)$ can be interpreted as the maximal worth or cost savings that the numbers of S can obtain when they cooperate. Often we identify a game $\langle N, c\rangle$ with its characteristic function c. The family of all cooperative games are denoted by G^N.

The Shapley value $\Phi(c)$ of a game $c \in G^N$ is defined the average of the marginal vectors of the game, i.e.,

$$\Phi(c) := \frac{1}{n!} \sum_{\sigma \in \pi(N)} m^\sigma(c).$$

Here, $m^\sigma(c)$ is the vector with

$$\begin{aligned} m^\sigma_{\sigma(1)}(c) &= c(\sigma(1)) - c(\emptyset), \\ m^\sigma_{\sigma(2)}(c) &= c(\sigma(1), \sigma(2)) - c(\sigma(1)), \\ &\vdots \\ m^\sigma_{\sigma(n)}(c) &= c(N) - c(\sigma(1), \sigma(2), \ldots, \sigma(n-1)). \end{aligned}$$

Shapley viewed the value as an index for measuring the power of players in a game and presented the value as an operator that assigns an expected marginal contribution to each player in the game with respect to a uniform distribution over the set of all permutations on the set of players. The Shapley value averages to aggregate the power of players in their various cooperation opportunities. Alternatively, one can think of the Shapley value as a measure of the utility of players in a game (Branzei et al. 2008).

In this study, we also concentrate on a one point solution, τ value, which is introduced by Tijs (1981). The τ value is based on the idea of a compromise between an upper and a lower value for each player in the game.

Let $c \in G^N$ be a cooperative cost game. The vector $M(c) \in \mathbb{R}^N$ with coordinates $M_i(c) := c(N) - c(N\backslash\{i\})$ is called the *upper vector* of c. Here, $M_i(c)$ can be regarded as the maximal payoff that player i can expect to get in the game in the sense that if the player claims more, then it is advantageous for the other players to exclude the player from the grand coalition N and to divide the value $c(N\backslash\{i\})$ among themselves. Moreover, $M_i(c)$ is also called the *utopia payoff* for player i. The vector $m(c) \in \mathbb{R}^N$ with coordinates

$$m_i(c) := \max_{S:\, i \in S} \left(c(S) - \sum_{j \in S\backslash\{i\}} M_j(c) \right)$$

is called the *lower vector* of c. For each player $i \in N$, the value $m_i(c)$ can be regarded as the *minimal right* in the sense that the player can guarantee him/herself this payoff offering the members of a suitable coalition their utopia payoff, which is a good deal to them, and taking the remainder for him/herself. It makes sense to consider a compromise between the lower and the upper vectors if the following two conditions are satisfied:

- $m(c) \le M(c)$,
- $\sum_{i \in N} m_i(c) \le c(N) \le \sum_{i \in N} M_i(c)$.

In the first condition, "$\le$" is understood in the componentwise sense. A game $c \in G^N$ that satisfies these two conditions is said to be *quasi-balanced*. The class of quasi-balanced

games with player set N is denoted by Q^N. For a quasi-balanced game $c \in Q^N$, the τ value of c, denoted by $\tau(c)$, is the unique compromise between the upper and lower vectors of the game that establishes an allocation of the value $c(N)$. Thus,

$$\tau(c) := m(c) + \alpha(M(c) - m(c)),$$

where $\alpha \in [0, 1]$ and $\sum_{i \in N} \tau_i(c) = c(N)$.

We note that the τ value is defined only for quasi-balanced games. The class of quasi-balanced games contains all games that have a non-empty core. For more information on the τ value we refer the reader to Tijs (1981) and Tijs and Otten (1993).

3 XTR

3.1 Introduction to XTR

The XTR (effective and compact subgroup trace representation) cryptosystem is introduced by Lenstra and Verheul (2000). In our study, we propose a novel public key encryption scheme using XTR in our crypto-cloud computing system since XTR uses a oneway trace map and provides a great advantage for the transmission size. Therefore, we recall the XTR system and give some facts to be utilized in construction of novel encryption scheme.

3.2 Description of XTR

The finite field $\mathbb{F}^*_{p^6} = \mathbb{F}_{p^6} \backslash \{0\}$ is a multiplicative group of order $p^6 - 1$. The subgroup of $\mathbb{F}^*_{p^6}$ has to divide the factorization of $p^6 - 1$, which corresponds to the dth cyclotomic polynomial with $d \mid 6$, as follows:

$$\begin{aligned} p^6 - 1 &= \prod_{d \mid 6} \Phi_d(p) \\ &= (p-1)(p+1)(p^2+p+1)(p^2-p+1). \end{aligned}$$

XTR uses the properties of the trace map over $\mathbb{F}_{p^2}$ to define elements of the subgroup of order dividing $\Phi_6(p) = p^2 - p + 1$ in $\mathbb{F}^*_{p^6}$. Since there is no subfield containing this subgroup, the security of extension field $\mathbb{F}_{p^6}$ is ensured. This also provides us to avoid index calculus attack.

Let p be a prime with $p \equiv 2 \mod 3$ and $f(x) = x^3 - ax^2 + bx - 1$ be an irreducible polynomial over the finite field $\mathbb{F}_{p^2}$ with period Q dividing $p^2 - p + 1$ in $\mathbb{F}^*_{p^6}$. Let α be a root of $f(x)$ in the extension field $\mathbb{F}_{p^6}$ of $\mathbb{F}_{p^2}$, then the conjugates of α with respect to $\mathbb{F}_{p^2}$, α^{p^2} and α^{p^4}, are the other two roots of $f(x)$ in $\mathbb{F}_{p^6}$. For $\alpha \in \mathbb{F}^*_{p^6}$, its trace $Tr(\alpha)$ over $\mathbb{F}_{p^2}$ is defined by

$$a = Tr(\alpha) = \alpha + \alpha^{p^2} + \alpha^{p^4} \in \mathbb{F}_{p^2}.$$

Since the period Q of $f(x)$ divides $p^2 - p + 1$, the order of α also divides $p^2 - p + 1$, so that

$$a = Tr(\alpha) = \alpha + \alpha^{p-1} + \alpha^{-p},$$

since $p^2 \equiv p - 1 \mod (p^2 - p + 1)$ and $p^4 \equiv -p \mod (p^2 - p + 1)$. By using these relations, we obtain

$$b = \alpha^{1+p^2} + \alpha^{1+p^4} + \alpha^{p^2+p^4} = Tr(\alpha)^p = a^p.$$

Therefore, the polynomial $f(x)$ over $\mathbb{F}_{p^2}$ can be rewritten as follows:

$$f(x) = x^3 - Tr(\alpha)x^2 + Tr(\alpha)^p x - 1. \tag{3.1}$$

Let $\underline{s} = \{s_i\}$ be the third-order characteristic sequence generated by $f(x)$. It follows from the Newton's formula that $\underline{s}$ can be represented by

$$s_{n+3} = Tr(\alpha)s_{n+2} - Tr(\alpha)^p s_{n+1} + s_n \quad (n \in \mathbb{N}),$$

with the initial conditions $s_0 = 3$, $s_1 = a$, $s_2 = a^2 - 2a^p$. In this case, $\underline{s}$ also has the trace representation, given by

$$s_n = s_n(Tr(\alpha), Tr(\alpha)^p) = Tr(\alpha^n) = \alpha^n + \alpha^{np^2} + \alpha^{np^4} \quad (n \in \mathbb{N}).$$

This implies that the nth powers of the roots of $f(x)$ are the roots of the polynomial $f_n(x)$ over $\mathbb{F}_{p^2}$, i.e., α^n and its conjugates with respect to $\mathbb{F}_{p^2}$ are fully determined by $Tr(\alpha^n)$ with a compression factor 3. It follows from this fact that the field arithmetic is carried out in $\mathbb{F}_{p^2}$, whereas the security lies in $\mathbb{F}_{p^6}$.

We note that $per(f) = Q$, the so-called period of $f(x)$ in $\mathbb{F}_{p^6}$, is equal to the $per(\underline{s})$, since $f(x)$ is irreducible over $\mathbb{F}_{p^2}$. If $\gcd(per(f), n) = 1$, $f(x)$ and $f_n(x)$ have the same period, so that $f_n(x)$ is irreducible over $\mathbb{F}_{p^2}$.

3.3 XTR exponentiation

In XTR, Lenstra and Verheul propose an efficient algorithm (Algorithm 2) to compute $s_n = Tr(\alpha^n)$ for given $Tr(\alpha)$ and $n \in \mathbb{N}\backslash\{0\}$ in Lenstra and Verheul (2000), Algorithm 2.37 by using the following relations, called as XTR addition and XTR doubling, respectively:

$$\begin{aligned} s_{u+v} &= s_u s_v - s_v^p s_{u-v} + s_{u-2v}, \\ s_{2u} &= s_u^2 - 2s_u^p, \end{aligned}$$

for $u, v \in \mathbb{Z}$.

Algorithm 2 XTR single exponentiation

Require: $s_1 = Tr(\alpha)$ and $0 < n = \sum_{j=0}^{t} n_j 2^j < Q$ with $n_t = 1$
Ensure: $(s_{2n}, s_{2n+1}, s_{2n+2})$
1: $(s_{y-1}, s_y, s_{y+1}) \leftarrow (3, s_1, s_1^2 - 2s_1^q)$
2: **for** $j \leftarrow t$ **to** 0 **do**
3: **if** $m_j = 1$ **then**
4: $s_{y-1} \leftarrow s_y^2 - 2s_y^q$
5: $s_y \leftarrow s_{y+1}s_y - s_1 s_y^q + s_{y-1}^q$
6: $s_{y+1} \leftarrow s_{y+1}^2 - 2s_{y+1}^q$
7: **else**
8: $s_{y-1} \leftarrow s_{y-1}^2 - 2s_{y-1}^q$
9: $s_y \leftarrow s_{y-1}s_y - s_1^q s_y^q + s_{y+1}^q$
10: $s_{y+1} \leftarrow s_y^2 - 2s_y^q$
11: **end if**
12: **end for**
13: **return** (s_{y-1}, s_y, s_{y+1})

Theorem 3.1 (Lenstra and Verheul 2000, Theorem 2.38) *Let $Tr(\alpha)$ and $n \in \mathbb{N}\backslash\{0\}$ be given. The computational cost of the XTR single exponentiation $s_n = Tr(\alpha^n)$ takes $8\log_2 n$ multiplications in $\mathbb{F}_p$.*

Lenstra and Verheul also introduce a double exponentiation algorithm to compute the mixed term $Tr(\alpha^{eu+kv})$ using matrices in Lenstra and Verheul (2000), Algorithm 2.48. Later, Stam and Lenstra (2001), Algorithm 3.1 propose more efficient double exponentiation algorithm to compute $Tr(\alpha^{eu+kv})$ without using matrices. They also show how to use the proposed double exponentiation algorithm to speed up the single exponentiation algorithm (Stam and Lenstra 2001, Algorithm 5.1), which is given in Algorithm 3.

Algorithm 3 XTR Single Exponentiation Using Double Exponentiation

Require: $s_1 = Tr(\alpha)$, n with $0 < n < Q$.
Ensure: $s_n = Tr(\alpha^n)$.
1: $k \leftarrow \left\lfloor \frac{3-\sqrt{5}}{2} n \right\rfloor$, $e \leftarrow n-k$, $u \leftarrow 1$, $v \leftarrow 1$.
2: $s_u \leftarrow s_1$, $s_v \leftarrow s_1$, $s_{u-v} \leftarrow s_0 = 3$, $s_{u-2v} \leftarrow s_{-1} = s_1^p$.
3: Apply (Stam and Lenstra 2001, Algorithm 3.1) to e, k, s_u, s_v, s_{u-v} and s_{u-2v}.
4: Return $s_{eu+kv} = Tr(\alpha^{eu+kv})$.

Theorem 3.2 (Stam and Lenstra 2001, Corollary 5.3) *Let $Tr(\alpha)$ and $n \in \mathbb{N}\backslash\{0\}$ be given. The computational cost of the XTR single exponentiation $s_n = Tr(\alpha^n)$ using double exponentiation algorithm takes in about $5.2\log_2 n$ multiplications in $\mathbb{F}_p$.*

Remark 3.3 In Algorithm 3, $\lfloor x \rfloor$ is an integer-valued function, which is closest to x. Note also that speeding up Algorithm 3, the best way to split n into a sum of two positive numbers e and k is choosing k/e closest to the golden ratio $\frac{1+\sqrt{5}}{2}$, which is the asymptotic ratio between two consecutive Fibonacci numbers.

The commutative law of the characteristic sequence generated by the irreducible polynomial of order three over $\mathbb{F}_p$ is introduced by Gong et al. (2001), Lemma 4. We now adapt this law to XTR, which plays an important role to construct trace-based schemes.

Lemma 3.4 (Commutative Law) *Let $f(x) = x^3 - Tr(\alpha)x^2 + Tr(\alpha)^p x - 1$ be an irreducible polynomial over $\mathbb{F}_{p^2}$. Let $\underline{s}$ be the characteristic sequence of $f(x)$. For all $e, k \in \mathbb{N}\backslash\{0\}$,*

$$\begin{aligned} s_{ek}(Tr(\alpha), Tr(\alpha)^p) &= s_e(Tr(\alpha^k),\ Tr(\alpha^k)^p) \\ &= s_k(Tr(\alpha^e),\ Tr(\alpha^e)^p). \end{aligned}$$

3.4 Proposed scheme using XTR

We now propose a novel encryption scheme which is based on the concepts of XTR.

Scheme. Let $f(x) = x^3 - Tr(\alpha)x^2 + Tr(\alpha)^p x - 1$ be an irreducible polynomial over $\mathbb{F}_{p^2}$ with $\mathrm{per}(f) = Q$ dividing $p^2 - p + 1$. Let $\mathcal{A}$ and $\mathcal{B}$ be two parties corresponding to Alice and Bob, respectively. $\mathcal{B}$ randomly selects a static private key $r \in \mathbb{N}\backslash\{0\}$ satisfying $r < Q$ such that $\gcd(r, Q) = 1$, and he computes his static public key $s_r = Tr(\alpha^r) \in \mathbb{F}_{p^2}$.

Public Parameters: $Tr(\alpha)$, $Tr(\alpha^r)$.

Private Parameters: r.

Encryption: $\mathcal{A}$ encrypts a message $m \in F_{p^2}$ as follows:

Table 1 Comparison of the proposed encryption scheme with the similar ones

	XTR (Lenstra and Verheul 2000)	GH (Gong and Harn 1999)	Improved GH (Ashraf and Kırlar 2014)	Proposed
Encryption	$16\log_2 n$ **M**	$18\log_2 n$ **M**	$(1 + 10.4\log_2 n)$ **M**	$10.4\log_2 n$ **M**
Decryption	$8\log_2 n$ **M**	$9\log_2 n$ **M**	1 **I** + $(1 + 5.2\log_2 n)$ **M**	$5.2\log_2 n$ **M**

i. $\mathcal{A}$ randomly selects an ephemeral private key $t \in \mathbb{N}\backslash\{0\}$ satisfying $t < Q$ such that $\gcd(t, Q) = 1$, and she then computes her ephemeral public key $s_t = Tr(\alpha^t) \in \mathbb{F}_{p^2}$.
ii. $\mathcal{A}$ computes $s_{tr} = Tr(\alpha^{tr}) = s_{tr}(Tr(\alpha), Tr(\alpha)^p) = s_t(Tr(\alpha^r), Tr(\alpha^r)^p) \in \mathbb{F}_{p^2}$ using the static public key $s_r = Tr(\alpha^r)$ of $\mathcal{B}$.
iii. $\mathcal{A}$ computes $c = m + (s_{tr} + s_{tr}^p) = m + Tr_{\mathbb{F}_{p^2}/\mathbb{F}_p}(s_{tr}) \in \mathbb{F}_{p^2}$; then she sends the ciphertext $c \in \mathbb{F}_{p^2}$ along with her ephemeral public key $s_t = Tr(\alpha^t)$ to $\mathcal{B}$.

Decryption: $\mathcal{B}$ recovers the message $m \in \mathbb{F}_{p^2}$ as follows:

i. $\mathcal{B}$ computes $s_{rt} = Tr(\alpha^{rt}) = s_{rt}(Tr(\alpha), Tr(\alpha)^p) = s_r(Tr(\alpha^t), Tr(\alpha^t)^p)$ using $\mathcal{A}$'s ephemeral public key $s_t = Tr(\alpha^t)$.
ii. $\mathcal{B}$ computes $m = c - (s_{rt} + s_{rt}^p) = c - Tr_{\mathbb{F}_{p^2}/\mathbb{F}_p}(s_{tr}) \in \mathbb{F}_{p^2}$.

3.5 Computational complexity

The proposed encryption scheme does not have any multiplications in $\mathbb{F}_p$ coming from enciphering and deciphering procedures. The total computational costs only depend on the efficient computation of trace function. Since we have 2 trace computations in the encryption part, and 1 in the decryption part, the proposed encryption scheme totally requires $15.6\log_2 n$ multiplications in $\mathbb{F}_p$ by Theorem 3.2. The detailed computational costs comparing the similar ones are given in Table 1, where we enumerate the cost of field multiplications and field inversions in terms of **M** and **I**, respectively.

3.6 Security analysis

We now recall the definition of Trace-DLP given by Giuliani and Gong (2005). Here, they also show that the Trace-DLP is computationally equivalent to the DLP.

Definition 3.5 Given an element $c \in \mathbb{F}_{p^2}$, the problem of determining an index $k \in \mathbb{N}$ with $k \leq Q$ such that $Tr(\alpha^k) = c$ is called the *Trace Discrete Logarithm Problem* (*Trace-DLP*).

In 1982, the notion of *semantic security* is first proposed by Goldwasser and Micali (1982). They improved their definition by demonstrating that semantic security is equivalent to another definition of security called *ciphertext indistinguishability* (Goldwasser and Micali 1984). There are many different ways to define semantic security; the following particular version helps us to show the proposed encryption scheme having this property.

Definition 3.6 Given any two plaintexts m_1 and m_2, and the ciphertext c, which is the encryption of one of two plaintexts. An encryption scheme is called *semantically secure* if the polynomial-time adversaries, with the knowledge of m_1, m_2, c and the public parameters, cannot guess negligibly with better probability than 1/2 whether the ciphertext c is the encryption of m_1 or m_2.

It is obvious that the security of the proposed encryption scheme depends on the difficulty of trace discrete logarithm problem (Trace-DLP). It also has the property of semantic security because of the following reasons: suppose that $m_1, m_2 \in \mathbb{F}_{p^2}$ are two known messages from the attacker $\mathcal{E}$, and $\mathcal{E}$ gives these two messages to $\mathcal{A}$ for encryption. $\mathcal{A}$ randomly encrypts the message m_1 or m_2 such that $c = m + (s_{tr} + s_{tr}^p) = m + Tr_{\mathbb{F}_{p^2}/\mathbb{F}_p}(s_{tr}) \in \mathbb{F}_{p^2}$ and sends the ciphertext c to the attacker $\mathcal{E}$. Therefore, with the knowledge of m_1, m_2, c and public parameters, $\mathcal{E}$ can substract m_1 (resp. m_2) from c such that $c - m_1 = d_1 + (s_{tr} + s_{tr}^p)$ (resp. $c - m_2 = d_2 + (s_{tr} + s_{tr}^p)$), where $d_1 = m - m_1$ (resp. $d_2 = m - m_2$). It follows from this information that computing d_1 (resp. d_2) is equivalent to computing $s_{tr} + s_{tr}^p$. Even if in one way or another $s_{tr} + s_{tr}^p$ has been found by $\mathcal{E}$, it is impossible for him/her to find the pair (s_{tr}, s_{tr}^p) from this sum because for a given fixed $z \in \mathbb{F}_{p^2}$ and any $x \in \mathbb{F}_{p^2}$, one can split z as $z = x + (w - x)$, and obviously the total number of such pair is equal to $(p^2 - 1)/2$. However, given all $(p^2 - 1)/2$ ways to split $s_{tr} + s_{tr}^p$, there is no obvious way to identify the pair (s_{tr}, s_{tr}^p) among the total number $(p^2 - 1)/2$. In the sequel, we conclude that the proposed scheme has the property of semantic security.

In the following section, we shall bring together all the preparations made in Sects. 2 and 3.

4 Our model

In this study, to calculate the real costs of financial cloud services, Amazon Glacier cost calculator is used. Amazon Glacier is an extremely low-cast storage service that provides secure and durable storage for data archiving and backup. Glacier is part of the Amazon Web Services (AWS) suite of cloud computing services, it is designed for long-term storage of data. Moreover, it is optimized for data that are infrequently accessed and for which retrieval times of several hours are suitable (http://calculator.s3.amazonaws.com/index.html).

The parameters of Amazon Glacier cost calculator are shown in Table 2.

Consider a group of financial cloud services, each of which needs to be connected to some source, either directly or via other financial cloud services. In our model, we have 3 financial cloud services and 1 source as the cryptology system. Every possible connection has some (non-negative) costs associated to it and the problem is how to connect every financial service to the source such that the total joint cost of the created network is minimal. The costs of financial cloud services in the network are calculated with Amazon Glacier cost calculator (cf. Table 2).

Herewith, pricing and cost accounting mechanisms in financial terms are obtained. The cryptology system as a source locates in the public cloud for safety reasons. The financial cloud services are located in a public cloud; in this way, our model runs on a hybrid cloud. The cryptology part of our model refers to *Platform as a Service* (*PaaS*). In fact, PaaS aims to protect data, which is especially important in case of storage as a service. In case of congestion, there is the problem of outage from a cloud environment. Thus, the need for security against outage is important, the data need to be encrypted when hosted on a platform for security reasons. *Infrastructure as a Service* (*IaaS*) refers to the sharing of hardware resources such as storage, bandwidth, server, etc.; for executing services, the cost part of our model refers to this service (Table 3).

Remark 4.1 In our crypto-cloud computing model, the cost ψ of the required proposed encryption algorithm (Algorithm 4) is added to the costs between the financial cloud services and the cryptology system. Thereby, the new costs are $18.20 + \psi$, $13.43 + \psi$, $104.27 + \psi$,

Table 2 The parameters of Amazon Glacier

Parameters	Explanation
Storage	The amount of storage you expect to use in a month
Requests	
UPLOAD and RETRIEVAL	The number of UPLOAD and RETRIEVAL requests made to your Amazon Glacier account in a month
Data retrieved	The amount of data you expect to request from your Amazon Glacier account for each of the periods requested
Retrieval period	Spreading out your data retrievals over longer periods lowers your billable peak hourly retrieval rate which may allow you to reduce your costs significantly
Data transfer	
Inter-region data transfer out	The amount of network data transfer out from this service to another AWS Region(s) or Amazon CloudFront
Data transfer out	The amount of network data transfer you expect to go out if your Amazon Glacier account over the Internet in a month
Data transfer in	The amount of network data transfer you expect into your Amazon Glacier account over the Internet in a month

Table 3 The costs of financial cloud services

Amazon Glacier parameters	FCS1	FCS2	FCS3	FCS1-FCS2	FCS1-FCS3	FCS2-FCS3	FCS1-FCS2-FCS3
Storage (GB)	750	500	10,240	150	10,990	10,740	11,490
UPLOAD and RETRIEVAL (request)	1000	500	10	1500	1010	600	1510
Data retrieved (GB)	20	150	30	170	50	180	200
Retrieved period (Month)	1	1	1	1	1	1	1
Inter-region data transfer out (GB/Month)	150	200	3	0	0	0	353
Data transfer out (GB/Month)	100	50	35	150	135	85	185
Data transfer in (GB/Month)	100	75	20	175	120	95	195
Total cost ($)	18.20	13.43	104.27	15.36	120.76	113.73	135.9

respectively. Thus, the total costs will be calculated for storing the encryption information of other financial cloud services' data store.

An illustration of our model can be seen in Fig. 1.

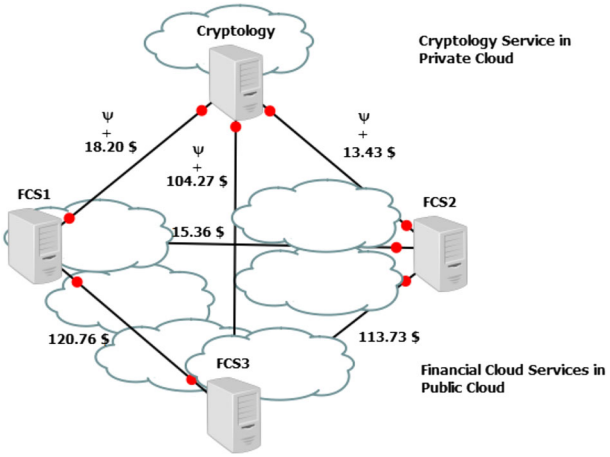

Fig. 1 The crypto-cloud computing model of the study

Table 4 The constructed cooperative cost game

S	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c(S)$	0	$18.20+\psi$	$13.43+\psi$	$104.27+\psi$	$28.79+\psi$	$122.47+2\psi$ or $138.96+\psi$	$117+2\psi$ or $127.16+\psi$	$149.55+\psi$

Algorithm 4 Encryption

Require: $Tr(\alpha), Tr(\alpha^r), m \in \mathbb{F}_{p^2}, t$ with $0 < t < Q$ and $\gcd(t, Q) = 1$

Ensure: $c \in \mathbb{F}_{p^2}$

1: $s_t \leftarrow Tr(\alpha^t)$
2: $s_{tr} \leftarrow Tr(\alpha^{tr}) = s_t(Tr(\alpha^r), Tr(\alpha^r)^p)$
3: $c \leftarrow m + (s_{rt} + s_{rt}^p) = m + Tr_{\mathbb{F}_{p^2}/\mathbb{F}_p}(s_{tr})$
4: Return c

The constructed cooperative cost game of our crypto-cloud computing model can be seen in Table 4. We note that, if $\psi \leq 16.49$ then $c(\{1,3\}) = 122.47+2\psi$ and $c(\{2,3\}) = 117+2\psi$, else $c(\{1,3\}) = 138.96+\psi$ and $c(\{2,3\}) = 127.16+\psi$.

When the financial cloud services are cooperative, decrease or increase in the costs are examined and three different solutions (the Bird rule, the Shapley value and τ value) are evaluated. In the following case, we choose $\psi = 5$. By applying the Bird rule in our problem, which can be seen in Fig. 1. We take the optimal solution with cost 151.55, and this gives the *Bird allocation* $\beta^R(\Gamma)$ as stated below:

$$\beta^R(\Gamma) = (15.36, 15.43, 120.76).$$

Then, our new constructed cooperative cost game of our model can be seen in Table 5.

Table 6 shows the marginal vectors of our crypto-cloud computing model, where $\sigma : N \rightarrow N$ is identified with $(\sigma(1), \sigma(2), \sigma(3))$.

Table 5 The new constructed cooperative cost game

S	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c(S)$	0	23.20	18.43	109.27	33.79	132.47	127	154.55

Table 6 The marginal vectors of our model

σ	$m_1^\sigma(c)$	$m_2^\sigma(c)$	$m_3^\sigma(c)$
$\sigma_1 = (1,2,3)$	23.20	10.59	120.76
$\sigma_2 = (1,3,2)$	23.20	22.08	109.27
$\sigma_3 = (2,1,3)$	15.36	18.43	120.76
$\sigma_4 = (2,3,1)$	27.55	18.43	108.57
$\sigma_5 = (3,1,2)$	23.20	22.08	109.27
$\sigma_6 = (3,2,1)$	27.55	17.73	109.27

The average of the six marginal vectors is the Shapley value of this game which can be calculated as:

$$\Phi(c) = (23.35, 18.22, 112.98)\,.$$

In order to find the τ value, we have a look at our game $c \in G^N$ whether it satisfies the two conditions; then it is said to be *quasi-balanced*. The elements of our utopia payoff vector $M_i(c)$ can be calculated as:

$$\begin{aligned} M_1(c) &= c(N) - c(\{2,3\}) = 27.55,\\ M_2(c) &= c(N) - c(\{1,3\}) = 22.08,\\ M_3(c) &= c(N) - c(\{1,2\}) = 120.77. \end{aligned}$$

In our game, the upper vector is calculated as $M(c) = (27.55, 22.08, 120.77)$ and the lower vector $m(c) = (m_1(c), m_2(c), m_3(c))$ is calculated as:

$$\begin{aligned} m_1(c) &= \max_{1\in S}(c(\{1\}), c(\{1,2\}) - M_2(c), c(\{1,3\}) - M_3(c), c(\{1,2,3\}) - M_2(c) - M_3(c))\\ &= 23.20,\\ m_2(c) &= \max_{2\in S}(c(\{2\}), c(\{1,2\}) - M_1(c), c(\{2,3\}) - M_3(c), c(\{1,2,3\}) - M_1(c) - M_3(c))\\ &= 18.43,\\ m_3(c) &= \max_{3\in S}(c(\{3\}), c(\{1,3\}) - M_1(c), c(\{2,3\}) - M_2(c), c(\{1,2,3\}) - M_1(c) - M_2(c))\\ &= 109.27, \end{aligned}$$

Hence, $m(c) = (23.20, 18.43, 109.27)$.

Since $\tau(c) := m(c) + \alpha(M(c) - m(c))$, where $\alpha \in [0,1]$, is such that $\sum_{i\in N}\tau_i(c) = c(N)$, α can be calculated as $\alpha = 0.1873$. Hence, the τ value of c is given by

$$\tau(c) = (24.01, 19.11, 11.42).$$

Let us choose $\psi = 20$.

The *Bird allocation* $\beta^R(\Gamma)$ is stated as follows:

$$\beta^R(\Gamma) = (15.36, 33.43, 120.76).$$

The Shapley value $\Phi(c)$ of new game is calculated as:

$$\Phi(c) = (28.54, 20.25, 120.76)\,.$$

However, the τ value for $\psi = 20$ can not be calculated since the game is not *quasi-balanced*. The upper vector is calculated as $M(c) = (22.39, 10.59, 120.76)$ and the lower vector is calculated as $m(c) = (38.20, 33.43, 136.57)$. Therefore, the τ value can not be calculated.

In this study, we propose three solution concepts, namely the Bird rule, the Shapley value and the τ value, belonging to our crypto-cloud computing system by using cooperative game theory. It can be seen that the cost for Player 1 and Player 2, with respect to the Bird rule, is lower than the Shapley value. The cost for Player 3, with respect to the τ value, is lower than the Bird rule and the Shapley value. In the example which we take $\psi = 20$, the cost for Player 1, with respect to the Bird rule, is lower than the Shapley value, where $\psi = 5$. The cost for Player 2, with respect to the Shapley value, is lower than the Bird rule. It can be seen that for Player 3, two solutions are the same. Let us note that the choice of the players depends on ψ.

5 Conclusion and outlook

In this paper, we introduce the theory of crypto-cloud computing with an efficient encryption algorithm under XTR by bringing together main topics of cloud computing, cooperative game theory and cryptology.

Using the approach of this study, financial services firms can meet the technical challenges of cloud computing and build a comprehensive, scalable and effective cloud strategy. The most interesting property of our work is the synergy achieved between cryptographic solutions and the cooperative game theory world in financial problems of cloud-computing application areas. We observe that by implementing our cryptographic solution into the cooperative game theory setting, we gain at both the cooperative game-theoretical side and the cryptographic side, but we also benefit in the emerging world of cloud computing.

As a future work, we plan to apply to our model other solution concepts of cooperative game theory and obligation rules from connection situations by including the comparison of efficient encryption algorithms.

Before closing, we note that the research areas of game theory, cryptology and cloud computing are already extensively or just increasingly studied fields, respectively, with many problems and new solution concepts. Yet, the cross-over between them was very small, surprisingly. The importance of this study is given by that it is so for a pioneering and ongoing project and offer to the operational research community in Europe and the world.

Acknowledgements The authors thank the anonymous referees for their detailed and very helpful comments.

References

Akyıldız, E., & Ashraf, M. (2014). An overview of trace based public key cryptography over finite fields. *Journal of Computational and Applied Mathematics*, *259–B*, 599–621.

Amazon Web Services. http://calculator.s3.amazonaws.com/index.html.

Ashraf, M., & Kırlar, B. B. (2014). Message transmission for GH-public key cryptosystem. *Journal of Computational and Applied Mathematics*, *259–B*, 578–585.

Bird, C. G. (1976). On cost allocation for a spanning tree: A game theoretic approach. *Networks*, *6*(4), 335–350.

Borm, P., Hamers, H., & Hendrickx, R. (2001). Operations research games: A survey. *Top*, *9*(2), 139–199.
Branzei, R., Dimitrov, D., & Tijs, S. (2008). *Models in cooperative game theory* (Vol. 556). Berlin: Springer.
Ciftci, B., & Tijs, S. H. (2007). A vertex oriented approach to minimum cost spanning tree problems.
Claus, A., & Kleitman, D. J. (1973). Cost allocation for a spanning tree. *Networks*, *3*(4), 289–304.
Evans, J. R., & Minieka, E. (1992). *Optimization algorithms for networks and graphs*. Berlin: CRC Press.
Garg, A. (2011). *Cloud computing for the financial services industry*. Sapient Global Markets, Sapient Corporation.
Giuliani, K. J., & Gong, G. (2005). New LFSR-based cryptosystems and the trace discrete log problem (Trace-DLP), sequences and their applications-SETA 2004. *Lecture Notes in Computer Science*, *3486*, 298–312.
Goldwasser, S., & Micali, S. (1982). Probabilistic encryption and how to play mental poker keeping secret all partial information. *Proceedings of 14 Annual ACM Symposium on Theory of Computing* (pp. 365–377).
Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. *Journal of Computer and System Sciences*, *28*, 270–299.
Gong, G., & Harn, L. (1999). Public-key cryptosystems based on cubic finite field extensions. *IEEE Transactions on Information Theory*, *45*(7), 2601–2605.
Gong, G., Harn, L., & Wu, H. (2001). The GH public-key cryptosystem, in SAC'01. *Lecture Notes in Computer Science*, *2259*, 284–300.
Gök, S. A., Branzei, R., & Tijs, S. (2010). The interval Shapley value: An axiomatization. *Central European Journal of Operations Research*, *18*(2), 131–140.
Graham, R. L., & Hell, P. (1985). On the history of the minimum spanning tree problem. *Annals of the History of Computing*, *7*(1), 43–57.
Intel IT Center. (2013). Securing the Cloud for Financial Instituations, Industry Brief.
Kırlar, B. B. (2012). The final exponentiation in pairing-based cryptography. *International Journal of Information Security Science*, *1*(1), 1–12.
Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, *7*(1), 48–50.
Lenstra, A. K., & Verheul, E. R. (2000). *The XTR public key system, advances in cryptology-crypto 2000, Lecture Notes in Computer Science* (Vol. 1880, pp. 1–9). New York: Springer.
Maillé, P., Reichl, P., & Tuffin, B. (2011). Of threats and costs: A game-theoretic approach to security risk management. In N. Gülpinar, P. G. Harrison, B. Rustem (Eds.), *Performance models and risk management in communications systems* (pp. 33–53). New York:Springer.
Moretti, S., Gök, S. Z. A., Branzei, R., & Tijs, S. (2011). Connection situations under uncertainty and cost monotonic solutions. *Computers and Operations Research*, *38*(11), 1638–1645.
Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, *36*(6), 1389–1401.
Roberts, F. S. (2008). Computer science and decision theory. *Annals of Operations Research*, *163*(1), 209–253.
Shapley, L. S. (1953). A value for n-person games. *Annals of Mathematics Studies*, *28*, 307–317.
Stam, M., & Lenstra, A. K. (2001). Speeding up XTR, advances in cryptology-Asiacrypt'01. *Lecture Notes in Computer Science*, *2248*, 125–143.
Tijs, S. (1981). Bounds for the core of a game and the t-value. In O. Moeschlin & D. Pallaschke (Eds.), *Game Theory and Mathematical Economics* (pp. 123–132). Amsterdam: North-Holland Publishing Company.
Tijs, S., & Otten, G. J. (1993). Compromise values in cooperative game theory. *Top*, *1*(1), 1–36.
West, D. B. (2001). *Introduction to graph theory* (Vol. 2). Upper Saddle River: Prentice Hall.