Epistemic Games with Conditional Believes for Modelling Security Threats Defence in Cloud Computing Systems

Łukasz Gaża
Cracow University of Technology
31-155 Cracow, Poland

Agnieszka Jakóbik
Cracow University of Technology
31-155 Cracow, Poland

KEYWORDS

Epistemic Games, Cloud, Security.

ABSTRACT

We presented Epistemic Games with Conditional Believes model for automating security decisions in Cloud Computing systems. The model assumes attack-defence scenarios. The game stages model Cloud provider and Cloud attacker rivalry to maximise their payoffs. The paper presents the methodology for including the believes about opponent's rationality. The presented model allows considering the attack on Cloud system in a realistic way. The proposed solution has been tested by the experimental analysis on Cloud Sim simulator. Presented model enables finding strategies for the Cloud provider to protect assets from cybersecurity attacks.

I. INTRODUCTION

The aim of the presented study is to examine the possibility of building the automatic decision system based on Epistemic Games for security decision making in Cloud Computing systems. This topic is important due to the their very rapid development and the fact that the complexity of such systems forces their users to automatise a lot of the decision making process. Game theory supports modelling strategic reasoning, considering rational game players who are benefiting from maximising their profits with interaction with other players. Automatising of security decisions in Cloud systems is the crucial process in securing such complex systems. The presented research is a continuation of our previous development of game theory based modelling of the competition between Cloud providers and Cloud attackers, [21],[15],[14]. The current research is a try to reformulate the assumptions about the Cloud attackers. It incorporates the believes about the utility functions for the Cloud attackers, instead. Epistemic games include into the decision-making process the decision-maker's beliefs about the state of the environment and opponents rationality. It models the situation when each game player is maximising the subjective expected utility assuming not the opponents utility functions, but the defenders belief that such functions will be used.

Our main contributions are:

- to adapt the epistemic approach for modelling the uncertainty of the attacker utility function;
- to introduce the model of belief as the the behaviour scheme of the computer system attacker;
- to propose the payoff functions for both Cloud defender and Cloud attacker.

The paper is organised as follows. Section (II) is describing the game modelling in the context of security decisions. Section (III) is presenting our model incorporating both payoff functions. In section IV we presented the results of the simulation based on Cloud Sim testing environment and the Python coded model. The paper ends with Section (V), which contains conclusions based on the conducted experiments and obtained results. Ideas for future work and potential improvements are also discussed there.

II. RELATED WORK

Game theoretic models were successfully used for modelling security related decisions. In [16], authors used Nash game for securing wireless network system. A zero-sum multi-stage two-player competitive game was used in [11] for securing a network of computers. In [23] Bayesian game was introduced and in [12], authors used stochastic game models. Multiple adversaries were considered in [6] and The bi-level game-theoretic model was used. In [9] authors used Bayesian attacker detection games with incomplete information for modelling the interaction between nodes in wireless networks with channel uncertainty and the concept of Nash equilibrium. Stackelberg games were applied for choosing the security levels of virtual machines [20]. A Cloud system defence was modelled in [15],[14] but we found the game model assumptions to be too strong to model the realistic attacks. For the best of our knowledge epistemic approach was not used so far for modelling the uncertainty of the attacker utility function. A novelty of the proposed approach is to use the concept of belief as the behaviour scheme of the computer system attacker. An additional novelty is the formulation of the payoff functions considering the probabilities of successful attacks in case when considered asset is protected or not. The main differences between the cited solutions and the proposed paper are the usage of Epistemic Game Theory instead of traditional game-based approaches. It enables to model the Cloud Computing related security decisions considering the fact that some information is

Communications of the ECMS, Volume 36, Issue 1,
Proceedings, ©ECMS Ibrahim A. Hameed, Agus Hasan,
Saleh Abdel-Afou Alaliyat (Editors) 2022
ISBN: 978-3-937436-77-7/978-3-937436-76-0(CD) ISSN 2522-2414

uncertain and therefore may be provided as the belief.

III. THEORETICAL MODEL

Let us denote by $i = 1, 2, .., N$ the set of players. For each player let C_i be a set of possible choices. For a particular player number i a choice combinations for his opponents is a list $(c_1, .., c_{i-1}, c_{i+1}, ..., c_N)$ where $c_1 \in C_1, .., c_{i-1} \in C_{i-1}, c_{i+1} \in C_{i+1}, ..., c_N \in C_N$. A belief for a player i about his opponent's choices is a probability distribution b_i over the set $C_1 \otimes C_{i-1} \otimes C_{i+1} \otimes C_N$ that defines for every opponents choice $(c_1, .., c_{i-1}, c_{i+1}, ..., c_N)$ some probability $b_i(c_1, .., c_{i-1}, c_{i+1}, ..., c_N) \geq 0$ such that

$$\sum_{i=1}^{N} b_i(c_1, .., c_{i-1}, c_{i+1}, c_N) = 1 \tag{1}$$

A utility function for a player i assigns a number $u_i(c_1, .., c_N)$ to every combination of $(c_1, .., c_N)$ and represents the outcome that the player i derives.

A dynamic game is defined by [18]:

- non-terminal history x that consists a set of choices that have been made by players in the past and resulted in x;
- the beginning of the game is a non-terminal history denoted by $\oslash$;
- terminal history represents the situation when the game ends. Every terminal history z consists a set of choices that leads to z;
- the set of non-terminal histories is denoted by X and set of terminal histories is denoted by Z;
- for every non-terminal history $x \in X$ let the $I(x)$ denotes the set of players who must choose at x and call it the set of active layers at x. For a player i the set X_i denotes the set of non-terminal histories where player i is active;
- for every player i a collection of sets of information that i has about the opponent's past choices is denoted by H_i. Every information set $h \in H_i$ consist of a set of non-terminal histories $x_1, x_2, .., x_k$ that have been realised without the knowledge which one. If $h = x$ then player is sure that history x has been realised;
- a set of available choices for player i and information set h is denoted by $C_i(h)$ meaning that the player i is able to make any choice from $C_i(h)$ when the game reaches the information set h.

When game reaches the terminal state $z \in Z$ every single player is rewarded by utility $u_i(z)$. Here we are considering only a perfect recall game when every player remembers his choices and his opponent's previous choices.

Complete choice plan is called a strategy. A strategy for a player i is a function s_i that assigns to his information set $h \in H_i$ available choice $s_i(h) \in C_i(h)$ unless h can not be reached due to some choice $s_i(h')$ at earlier information set $h' \in h_i$. In this case no choice needs to be done at h. We denote by S_i a set of all strategies for player i.

The chosen strategy combination $(s_1, .., s_{i-1}, s_{i+1}, s_N)$ leads to h if there is s_i strategy for player i that together with this strategy would lead to h. Strategy s_i leads to information set h if there is some strategy for the opponents that together with this strategy would lead to h. Those assumptions result in the definition of the conditional belief for a player i at h about the opponent's strategies that is a probability distribution $b_i(h)$ over the set of the opponent's strategy combinations assigning a positive probability only to strategy combinations that leads to h.

Among all strategies, we may consider dominant strategies:

- a strictly dominant strategy is that strategy that always provides greater utility to a the player, taking into account all the other player's strategies;
- a weakly dominant strategy is strategy that results at least the same utility for all the other player's strategies, [19].

Lets consider the information set h for player i. In such a case if player holds a conditional belief $b_i(h)$ and given the strategy s_i that leads to h this strategy is optimal if:

$$u_i(s_i, b_i(h)) \geq u_i(s_i', b_i(h)) \tag{2}$$

for every other strategy s_i' that leads to h. Similarly, a conditional belief vector $b_i = [(b_i(h)]_{h \in H_i}$ for player i about his opponent's strategies concatenates at every information set $h \in H_i$ conditional beliefs $b_i(h)$ about the opponent's strategies.

A belief hierarchy in dynamic games defines for every player belief about the opponent's choices and the opponent's belief hierarchies. The hierarchy is formulated as follows:

- First order belief: the belief that player has about the opponent's strategies; Let us denote the belief hierarchy by $t_i^{s_i}$ (type) indication belief of the player i starting at his choice s_i.
- Second order belief: the belief that player has about the belief that the opponents have about their opponent's strategies;
- Third order belief: the belief that player has about the belief that the opponents has about the belief of this player opponent's strategies;
- and so on.

For every player i we denote by T_i the set of types that are considered for this player. Then, an epistemic model specifies for every player i a set T_i of possible types. Additionally, every type t_i for player i specifies for every information set $h \in H_i$ a probability distribution $b_i(t_i, h)$ over the set:

$$(S_1 \otimes T_1) \otimes ... \otimes (S_{i-1} \otimes T_{i-1}) \otimes (S_{i+1} \otimes T_{i+1}) \otimes ... \otimes (S_n \otimes T_N) \tag{3}$$

of his opponent's strategy - type combination. This probability distribution assigns only positive probability to the opponent's strategy combinations that leads to h. b_i represents the conditional belief that type t_i has at h about the opponent's strategies and types.

To define the beliefs about the future strategies, we introduce:

- Two information sets h and h' are simultaneous if there is a history that is present in both h and h';

• Information set h' follows information set h if there is a history x in h and a history x' in h' such that history x' follows history x;
• Information set h' weakly follows information set h either h' follows h or is simultaneous with h;

If we consider a type t_i for player i, and information set h for player i and an information set h' for player j, then we say that type t_i beliefs at h that the opponent j will chose rationally at h' if his conditional belief $b_i(t_i, h)$ at h assigns only positive probability to strategy-type pairs (s_j, t_j) for player j where strategy s_j is optimal for type t_j at information set h'. Analogically, we say that type t_i beliefs at h in opponent's future rationality if t_i beliefs at $h$that j will choose rationally at every information set h' for player j that weakly follows h. Type t_i expresses the common belief in future rationality if t_i express k-fold belief in future rationality for every order belief k.

Now, we can define choosing the rational strategy under belief in future rationality, as follows:
• Player i can rationally choose some strategy s_i under common belief in future rationality if there is some epistemic model and some type t_i for player i in this model such that t_i expresses common belief in future rationality and strategy s_i is optimal for type t_i at every information set $h \in H_i$ that s_i leads to.

Consider an information set $h \in H_i$ for player i. Let $S_i(h)$ be a set of strategies of player i that leads to h and $S_{-i}(h)$ be a set of his opponent's strategy combinations that leads to h. The pair

$$\Gamma^0(h) = (S_i(h), S_{-i}(h)) \tag{4}$$

is called the full decision problem for player i at h. Similarly, a reduced decision problem for player i at h is a pair

$$\Gamma^0(h) = (D_i(h), D_{-i}(h)) \tag{5}$$

where $D_i(h) \subset S_i(h)$ and $D_{-i}(h) \subset S_{-i}(h)$.

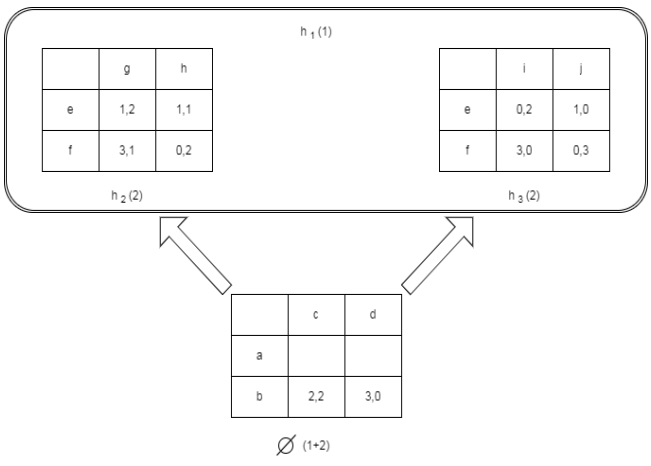

Fig. 1: The graph of game for two players

Algorithm for finding the strategies that can be rationally chosen under belief of future rationality is the backward dominance algorithm, [18]:

1. STEP 1: For every full decision problem $\Gamma^0(h)$ eliminate for every player i strategies that are strictly dominated at some decision problem $\Gamma^0(h')$ that weekly dominates $\Gamma^0(h)$ and at which player i is active. Denote resulted reduced decision problem by $\Gamma^1(h)$ for every information set h. If player i is active at h and the full decision problem is given by $(S_i(h), S_{-i}(h))$ when we are removing strategy from $(S_i(h)$. If player j is active at h but not player i and the full problem is a pair $(S_j(h), S_{-j}(h))$ in such case we are removing from $S_{-j}(h))$ every strategy combination that contains strategy s_i for player i.
2. STEP 2: For every reduced decision problem $\Gamma^1(h)$ eliminate for every player i those strategies that are strictly dominated at some reduced decision problem $\Gamma^1(h')$ that weakly follows $\Gamma^1(h)$ and at which player i is active. Denote resulted reduced decision problem by $\Gamma^2(h)$ for every information set h.
3. STEP 3: continue until no more strategies may be eliminated.

Theorem: For every $k >= 1$ the strategies that can rationally be chosen by a type that expresses up to k-fold belief in future rationality are strategies in Γ^{k+1} that survived the first $k+1$ steps of the backward dominance algorithm presented above. Additionally, the strategies that can rationally be chosen by a type that expresses common belief in future rationality are strategies that survived full backward dominance algorithm. Those strategies are in Γ^k for every k,[18].

The example of the game is presented in the fig 1. In the first stage player 1 may choose a or b, player 2 c or d.In the first case the game has another stage. In second case the game ends with a payoff for player 1 equal to 2 if player 2 chosen c, a payoff for player 1 equal 3 if player 2 chosen d. The relevant payoffs for player two are 2 and 0. If the game is continued the game moves to information set $h_1(1)$ for player 1 and information sets $h_2(2)$ and $h_3(2)$ for player 2. In the second stage player 1 may choose e or f, player 2 g or h if combination (a,c) was chosen in first stage. In the second stage player 1 may choose e or f, player 2 g or h but the player 2 may chose g or h if combination (a,c) was chosen in first stage and he may choose i or j if combination (a,d) was chosen in first stage. The payoffs are given by two tables in side rounded box. $X = \emptyset, (a,c), (a,d)$, $Z = \{(b,c), ((a,c), (f,h)), ..., (b,d)\}$, not-terminal histories $\emptyset, (a,c), (a,d)$, $I(\emptyset) = \{player1, player2\}$, $I((a,c)) = \{player1, player2\}$, $I((a,d)) = \{player1, player2\}$, $H_1 = \{\emptyset, h_1\}$, $H_2 = \{\emptyset, h_2, h_3\}$, $C_1(\emptyset) = \{a,b\}$,$C_1(h_1) = \{e,f\}$,$C_2(\emptyset) = \{c,d\}$, $C_2(h_2) = \{g,h\}$, $C_2(h_3) = \{i,j\}$, $u_1(b,c) = 2$, $u_2(b,d) = 0$,$u_1((a,d),(f,i)) = 3$,$u_2((a,d),(f,i)) = 0$, [18].

IV. NUMERICAL SIMULATION

In the proposed model, the attacker is a player 2 and the Cloud defender is player 1,(see Table 1).

The numerical experiment follows our research presented in [14]. *Assets* is the set of Cloud system components to be protected:

Player 1	Player 2
Cloud provider	Malicious individual, hacker
Defender	Attacker

TABLE 1: Players roles inside the cloud.

$$a \in Assets \tag{6}$$

Single attack is targeted into a specific asset. All considered attacks against the asset a are gathered in the form of the set:

$$Attacks^a = \{attack_1^a, \ldots, attack_m^a\} \tag{7}$$

where m is the number of considered attacks. A countermeasure is an action taken to protect the asset. A set of considered countermeasures against the asset a is denoted by:

$$controls^a = \{c_1^a, \ldots, c_n^a\}. \tag{8}$$

If we denote by $P^a(attack_i^a, c_j^a)$ be the probability of a successful attack on asset a by using threat number $i \in \{1, 2, \ldots m\}$ that are protected by the countermeasure $j \in \{1, 2, \ldots n\}$.

Therefore the j-th pure strategy,[19] s_1^j for the Defender is applying the countermeasure c_j^a. Then,

$$s_j^1 = 1, s_{-j}^1 = 0 \tag{9}$$

if c_j was chosen by player 1. Additionally, the i-th pure strategy s_2^i strategy for the Attacker is choosing the threat number i, that is

$$s_i^2 = 1, s_{-i}^2 = 0 \tag{10}$$

if a_i^a was chosen by player 2.

Val^a is income that player 1 gains from a protected asset a when it is working. $CostDef_{c_j^a}^a$ is cost of applying for asset a control number $j \in \{1, 2, \ldots n\}$. By $Gain^a$ let us denote reward for the player 2 for the successful attack into asset a, and by $CostAttack_{attack_i^a}^a$ let us denote the cost of such attack.

The payoff player 1 was modelled as:

$$\begin{aligned} &u_1(s_1^a, b_1(h))^a = \textstyle\sum_{i=1,\ldots,m} \sum_{j=1,\ldots,n} \\ &s_1^j \beta_1^i [P^a(attack_i^a, c_j^a) * (-Val^a - CostDef_{c_j^a}^a) \\ &+(1 - P^a(attack_i^a, c_j^a)) * (Val^a - CostDef_{c_j^a}^a)] + \ldots \end{aligned} \tag{11}$$

$$\begin{aligned} &+ \textstyle\sum_{i=1,\ldots,m} \sum_{j=1,\ldots,n} \\ &(1 - s_1^j)(\beta_1^i [\bar{P}^a(attack_i^a, c_j^a) * (-Val^a) \\ &+(1 - \bar{P}^a(attack_i^a, c_j^a)) * (Val^a)] \end{aligned} \tag{12}$$

where $\bar{P}^a(attack_i^a, c_j^a)$ is the probability of the successful attack number i on asset a considering the fact that the countermeasure c_j was chosen for this asset. Val^a indicates the value obtained by the Cloud provider from the asset working properly. This form of the payoff function assumes the worst case scenario. If the attack was successful, the task must be performed ones again therefore the provider lost the computational cost and payed for the protection resulting in $-Val^a - CostDef_{c_j^a}^a$, see eq.(11). If the Attack was unsuccessful, he invested in protection but his asset was working producing the income: $Val^a - CostDef_{c_j^a}^a$. If asset was not protected the provider may expect $-Val^a$ in case of successful attack, and Val^a income in case of not successful attack. Those costs are lower, but the probabilities of being attacked when unprotected are higher.

Analogously, the payoff for player 2 was as:

$$\begin{aligned} &u_2(s_2, b_2(h))^a = \\ &= \textstyle\sum_{i=1,\ldots,m} \sum_{j=1,\ldots,n} \\ &\beta_2^j s_2^i [P^a(attack_i^a, c_j^a) * (Gain^a - CostAtack_{attack_i^a}^a) \\ &+(1 - P^a(attack_i^a, c_j^a)) * (-CostAtack_{attack_i^a}^a)] + \ldots \end{aligned} \tag{13}$$

$$\begin{aligned} &+ \textstyle\sum_{i=1,\ldots,m} \sum_{j=1,\ldots,n} \\ &\beta_2^j (1 - s_2^i) [\bar{P}^a(attack_i^a, c_j^a) * (Gain^a - CostAtack_{attack_i^a}^a) \\ &+(1 - \bar{P}^a(attack_i^a, c_j^a)) * (-CostAtack_{attack_i^a}^a)] \end{aligned} \tag{14}$$

$$b_1(h) = (\beta_1^1, \beta_1^2, ..., \beta_1^m) \tag{15}$$

$$\beta_1^1 + \beta_1^2 + ... + \beta_1^m = 1 \tag{16}$$

$$b_2(h) = (\beta_2^1, \beta_2^2, ..., \beta_2^n) \tag{17}$$

$$\beta_2^1 + \beta_2^2 + ... + \beta_2^n = 1 \tag{18}$$

$$s_1^a = (c_1^a, c_2^a, ..., c_n^a) \tag{19}$$

and

$$s_2^a = (attack_1^a, attack_2^a, ..., attack_m^a) = (a_1^a, a_2^a, ..., a_m^a) \tag{20}$$

For simulating the attacks to the Cloud infrastructure, the numerical test was performed on a CloudSim environment [1]. The cloud infrastructure model is presented in Table 2 discussed in [14].

Asset	VM	Speed	Energetic profile
number a	type	GFLOPS	$min^a : max^a$ in Watts
-1	1	0.02	90:105
0	1	0.02	90:105
1-20	1	0.02	90:105
21-40	2	0.05	93:110
41- 60	3	0.1	100:120
61-80	4	0.2	150:170
81-100	5	0.3	200:230

TABLE 2: SimGrid VMs used for simulation, $Val^a = (max^a - min^a)/2$

The $Attacks^a$ were chosen according to the Cloud Security Alliance list of 7 most dangerous threats for cloud systems, see [7],see Table 3. :

1. a_1^a, $attack_1^a$: Task injection
2. a_2^a,$attack_2^a$: Denial-of-service attack (DoS attack)
3. a_3^a, $attack_3^a$: Task modification
4. a_4^a, $attack_4^a$: Distributed DoS attack (DDoS)
5. a_5^a,$attack_5^a$: Tasks loss
6. a_6^a,$attack_6^a$: Energy denial-of-service attack (eDOS) see [?].
7. a_7^a,$attack_7^a$: Unknown attack: asset not working.

The the tested $controls^a$ were selected from the cloud controls matrix [2][5]:

1. c_1^a: RSA digital signature with 1024 bit key for each task batch
2. c_2^a: "anti-virus" job to check input connections into the asset

3. c_3^a: "firewall" job to monitor tasks
4. c_4^a: escaping- closing infected VM, opening the new one
5. c_5^a: task integrity monitoring by SHA-2 hashing for each task batch
6. c_6^a energy cupping - scaling up the VM, see [20].

During tests on SimGrid environment we simulated the execution of tasks and measured the energy consumed by simulated VMs. All Virtual Machines were monitored during task execution, idle time and scaling and escaping (cloning).

Asset	Input protection	Inner protection	Output protection
-1-100	c_1^a,c_2^a,c_3^a	c_4^a , c_5^a, c_6^a	c_1^a,c_2^a,c_3^a
-1	RSA verific. of user	escaping VM	RSA sign.
0	RSA verif. of task collector	escaping VM	RSA sign.
1-100	RSA verif. of task scheduler	escaping VM	RSA sign.

TABLE 3: Asset protection scheme

The energy expenditure is presented in Table 2 and see Table 4.

					$Cost$	Def	^{a}	c_j^a		
type	P_i^i	P_b^i	P_o^i	P_c^i	c_1^a	c_5^a	c_2^a	c_3^a	c_6^a	c_4^a
1	90	106.8	63	27	20	18.4	0.53	1.06	54	90
2	93	114.6	65	28	18	14	0.57	1.14	56	93
3	100	130	70	30	12	12	0.65	1.3	60	100
4	150	200	105	45	10	3.4	1.0	2.0	90	150
5	200	290	140	60	6.5	2	1.45	2.9	120	200

TABLE 4: Measured power for a 10 GFLOPs workload [Watts] for all considered VM types, P_i^i power consumed in idle state, P_b^i power consumed in busy mode, P_o^i power consumed opening new VM, P_c^i power consumed for closing VM, the last six columns defines the values of $CostDef_{c_j^a}^a$

The payoff function for the player 2 was simulated in points, assuming bigger utility in case of successful attack on more powerful assets (see table 5).

Asset nr	Gain	$att._1^a$ cost	$att._2^a$ cost	$att._3^a$ cost	$att._4^a$ cost	$att._5^a$ cost	$att._6^a$ cost	$att._7^a$ cost
-1	10	1	2	2	1	2	5	9
0	60	2	4	2	1	2	5	50
1-20	10	3	6	2	1	2	10	9
21-40	20	10	10	12	1	12	15	18
41- 60	30	10	10	12	2	12	20	25
61-80	40	10	10	12	3	12	25	35
81-100	50	10	10	12	4	12	30	45

TABLE 5: Gain and Cost of Attacks $attack_1^a$-$attack_7^a$ on simulated Cloud in points

The probability of successfully attack was modelled based on [3] and are presented in Table 6 and Table 7.

Counterm.	at_1^a	at_2^a	at_3^a	at_4^a	at_5^a	at_6^a	at_7^a
c_1^a RSA	0	0.95	0.9	0.8	0	0.8	0.5
c_2^a Anti virus	0.95	0.9	0.6	0	0.8	0.8	0.5
c_3^a Firewall	0.95	0	0.6	0.8	0.8	0.8	0.5
c_4^a Escape	0.95	0	0.6	0	0	0	0
c_5^a SHA-2	0.95	0.9	0.6	0.8	0.8	0.8	0.5
c_6^a Cupping	0.95	0.9	0.6	0.8	0.8	0	0.5

TABLE 6: $P^a(attack_i^a, c_j^a)$, of successful attacks $attack_1^a - attack_7^a$ on assets protected by countermeasures $c_1^a - c_6^a$, for the sake of presentation simplicity assumed constant

The below Python code was used to simulate attack on different assets using the available attack methods and available control methods. The defender's gain is presented in Table 8 where the cost of the defence was

Counterm.	at_1^a	at_2^a	at_3^a	at_4^a	at_5^a	at_6^a	at_7^a
c_1^a RSA	0.5	1	0.95	0.8	0.2	1	0.6
c_2^a Anti virus	0.95	0.9	0.6	0.2	1	1	0.6
c_3^a Firewall	0.95	0	0.6	0.9	1	1	0.6
c_4^a Escape	0.95	0	0.6	0.5	0.2	0	0.2
c_5^a SHA-2	1	0.95	0.8	0.9	1	1	0.6
c_6^a Cupping	0.95	0.9	0.6	0.9	1	0	0.6

TABLE 7: Probabilities of successful attacks on assets that not protected, $\bar{P}^a(attack_i^a, c_j^a)$, for the sake of presentation simplicity assumed constant

defined in Table 4. The attacker's gain calculated using the below code is presented in Table 9 where the cost of the attack was defined in Table 5. The relation between attacks and defences was modelled by probabilities of successful attacks on protected assets defined in Table 6 and probabilities of successful attacks on not protected assets defined in Table 7.

```
def u_1(a, s1, beta_1):
    res = 0
    for i in range(num_attacks):
        for j in range(num_controls):
            res += s1[j] * beta_1[i] * (P_a[j][i] *
                 (-Val[a] - CostDef[a][j]) + (1 -
                 P_a[j][i])*(Val[a] - CostDef[a][j]))
    for i in range(num_attacks):
        for j in range(num_controls):
            res += (1 - s1[j]) * beta_1[i] *
                 (P_a_bar[j][i] * (-Val[a]) + (1 -
                 P_a_bar[j][i]) * Val[a])
    return res

def u_2(a, s2, beta_2):
    res = 0
    for i in range(num_attacks):
        for j in range(num_controls):
            res += beta_2[j] * s2[i] * (P_a[j][i] *
                 (Gain[a] - CostAttack[a][i])+ (1 -
                 P_a[j][i]) * (-CostAttack[a][i]))
    for i in range(num_attacks):
        for j in range(num_controls):
            res += beta_2[j] * (1 - s2[i]) *
                 (P_a_bar[j][i] * (Gain[a] -
                 CostAttack[a][i]) + (1 - P_a_bar[j][i])
                 * (-CostAttack[a][i]))
    return res
```

Asset	c_1^a	c_2^a	c_3^a	c_4^a	c_5^a	c_6^a
-1	-34.51	-33.61	-15.8	-16.06	-69.47	-106.08
0	-34.51	-33.61	-15.8	-16.06	-69.47	-106.08
1-20	-34.51	-33.61	-15.8	-16.06	-69.47	-106.08
21-40	-34.44	-31.24	-17.88	-18.14	-73.53	-111.23
41- 60	-31.35	-32.28	-21.01	-21.3	-80.63	-121.44
61-80	-29.35	-23.68	-21.36	-22.0	-110.63	-171.44
81-100	-35.52	-32.42	-32.0	-32.9	-150.94	-232.16

TABLE 8: Defender's gain for controls c_1^a-c_6^a

Asset	$att._1^a$	$att._2^a$	$att._3^a$	$att._4^a$	$att._5^a$	$att._6^a$	$att._7^a$
-1	25.46	26.30	25.77	24.6	24.0	24.94	24.74
0	218.75	223.79	220.62	213.6	210.01	215.66	214.46
1-20	15.46	16.30	15.77	14.6	14.0	14.94	14.74
21-40	16.92	18.60	17.54	15.2	14.0	15.89	15.49
41- 60	31.38	33.89	32.31	28.8	27.0	29.83	29.23
61-80	82.83	86.19	84.08	79.4	77.0	80.77	79.97
81-100	114.29	118.49	115.85	110.0	107.0	111.71	110.72

TABLE 9: Attacker's gain for attacks $attack_1^a$-$attack_7^a$

The results indicate that:
- the losses for Cloud provider due to attack differs according to the considered asset type,

• if the stronger computing units (assets 61-100) are attacked the cloud defender losses are more severe than in case of less power full ones (assets -1-60),
• in the future the Cloud defender should change the controls number 5 and 6 into stronger ones for all the assets in his system, see tab 8.

The simulations also show that, assuming the Cloud attacker rationality:
• the attacker will concentrate his efforts on asset number 1, that is the scheduling unit,
• the most beneficial attack type will be attack number 2. into scheduling unit.

Considering the above results the Cloud defender should invest in protecting the unit that is scheduling the tasks in his system.

V. CONCLUSIONS

In this paper we presented Epistemic Game theory based model with Conditional Believes to build automating system for security decision making process in Clouds. The model considers the separate payoff functions for modelling both attack and defence scenarios. It relates to the different objectives of Cloud defender and Cloud attacker. The model uses the belief concept to represent the rationality of the decision making process. The behaviour of the Cloud defender and Cloud attacker is calculated by using numerical optimisation for the mathematical model presented in eq. (1)-(20). The main result of modelling process is finding the best strategies for the Cloud provider to protect from cyber-security attacks. In the future, we would like to incorporate more advanced game models, that allow mixing Stackelberg Games with Epistemic Game theory with Conditional Believes.

References

[1] Cloudsim, http://www.cloudbus.org/cloudsim/.

[2] Security Guidance for Critical Areas of Focus in Cloud Computing V2.1. Technical report, 2009.

[3] OWASP, Top. 10 2010. The Ten Most Critical Web Application Security Risks. Technical report, 2010.

[4] NIST Special Publication 800-144: Guidelines on Security and Privacy in Public Cloud Computing. Technical report, 2011.

[5] CSA Controls Matrix v.3. Technical report, 2013.

[6] A. H. Anwar, G. Atia, and M. Guirguis. Game theoretic defense approach to wireless networks against stealthy decoy attacks. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 816–821, Sept 2016.

[7] J. Archer, A. Boehme, D. Cullinane, P. Kurtz, N. Puhlmann, and J. Reavis. Top Threats to Cloud Computing V1.0. Technical report, 2010.

[8] N. Basilico, A. Lanzi, and M. Monga. A security game model for remote software protection. In *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pages 437–443, Aug 2016.

[9] O. Dianat and M. Orgun. Modelling bayesian attacker detection game in wireless networks with epistemic logic. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 210–215, 2012.

[10] S. Dlugosz. *Multi-layer Perceptron Networks for Ordinal Data Analysis*. Logos Verlag, 2008.

[11] E. Eisenstadt and A. Moshaiov. Novel solution approach for multi-objective attack-defense cyber games with unknown utilities of the opponent. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(1):16–26, Feb 2017.

[12] M. P. Fanti, M. Nolich, S. Simié, and W. Ukovich. Modeling cyber attacks by stochastic games and timed petri nets. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002960–002965, Oct 2016.

[13] M. T. Hagan, H. B. Demuth, and M. Beale. *Pareto-Nash-Stackelberg Game and Control Theory*. Springer International Publishing, UK, 2018.

[14] A. Jakóbik, F. Palmieri, and J. Kołodziej. Stackelberg games for modeling defense scenarios against cloud security threats. *Journal of Network and Computer Applications*, 110:99 – 107, 2018.

[15] A. Jakóbik. Stackelberg game modeling of cloud security defending strategy in the case of information leaks and corruption. *Simulation Modelling Practice and Theory*, 103:102071, 2020.

[16] Y. Li, D. E. Quevedo, S. Dey, and L. Shi. A game-theoretic approach to fake-acknowledgment attack on cyber-physical systems. *IEEE Transactions on Signal and Information Processing over Networks*, 3(1):1–11, March 2017.

[17] S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.

[18] A. Perea. Epistemic game theory: Reasoning and choice. *Epistemic Game Theory: Reasoning and Choice*, pages 1–561, 01 2012.

[19] S. Tadelis. *Game Theory: An Introduction*. Princeton University Press, 2013.

[20] A. Wilczyński and A. Jakóbik. Using Polymatrix Extensive Stackelberg Games in Security–Aware Resource Allocation and Task Scheduling in Computational Clouds. *Journal of Telecommunications and Information Technology*, 1, 2017.

[21] A. Wilczyński, A. Jakóbik, and J. Kołodziej. Stackelberg security games: Models, applications and computational aspects. 3:70–79, 2016.

[22] X. Yang, X. He, J. Lin, W. Yu, and Q. Yang. A game-theoretic model on coalitional attacks in smart grid. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 435–442, Aug 2016.

[23] J. Zhu, B. Zhao, and Z. Zhu. Leveraging game theory to achieve efficient attack-aware service provisioning in eons. *Journal of Lightwave Technology*, 35(10):1785–1796, May 2017.

AUTHOR BIOGRAPHIES

ŁUKASZ GAŻA He received his M.Sc. in the field of Applied Physics with Computer Modelling at the Tadeusz Kosciuszko Cracow University of Technology. Since 2019 he is a Research and Teaching Assistant at the Tadeusz Kosciuszko Cracow University of Technology. His e-mail address is lukasz.gaza@pk.edu.pl.

AGNIESZKA JAKÓBIK She received her M.Sc. in the field of Stochastic Processes at the Jagiellonian University, Poland and a PhD degree in Artificial Neural Networks at the Tadeusz Kosciuszko Cracow University of Technology, Poland. Since 2009 she is an Assistant Professor at the Tadeusz Kosciuszko Cracow University of Technology, email: ajakobik@pk.edu.pl.