Information Sciences 607 (2022) 831–848

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Defending against fake VIP in scant-transparency information systems with QoS differentiation

Jerzy Konorski

Gdańsk University of Technology, ul. Narutowicza 11/12, 80-233 Gdańsk, Poland

ARTICLE INFO

Article history:
Received 18 March 2021
Received in revised form 2 May 2022
Accepted 5 June 2022
Available online 8 June 2022

Keywords:
Client–Server
QoS differentiation
QoS abuse
Stackelberg game

ABSTRACT

In client–server information systems with quality of service (QoS) differentiation, Client may deplete Server's resources by demanding unduly high QoS level. Such *QoS abuse* has eluded systematic treatment; known defenses using Client authorization, payments, or service request inspection prior to QoS assignment, are heuristic and environment-specific. We offer a game-theoretic approach on the premise that a service request is occasionally trusted to reduce the inspection cost. We call *Fake VIP attack* (FVA) a form of QoS abuse that consciously exploits Server's trust. An *FVA strategy* instills trust to maximize Client's utility gained from successful FVAs, whereas a *trust strategy* maximizes Server's utility by trading her loss due to successful FVAs against the request inspection cost. We consider a realistic *scant-transparency* setting where only long-term utilities are observable. Against a probabilistic FVA strategy we design a trust strategy based on *double-blind reputation*. Assuming a memoryless service request stream we analyze the impact of the request inspection cost and information leakage on the utilities at the Stackelberg equilibrium of the arising game. Experimental comparison with a real-world internally correlated stream is also shown.

1. Introduction

Today's information systems feature intelligent autonomous agents interacting with one another to exchange services. Agents are organized into multi-access service outlets with a massive customer population, e.g., Internet service providers, e-commerce portals, or public cloud systems, or into distributed collaborative environments, e.g., Mobile Edge Computing (MEC) systems [23], vehicular networks [46], or cellular networks using cooperative communication [31]. During an interaction, Client issues *service requests* and Server is responsible for service provision, where *service* refers to handling a specific service request [38], or a service request class [14,39].

The *quality of service* (QoS) concept subsumes Client's service requirements and satisfaction in terms of key performance indicators, such as throughput or response latency. We assume that Server has QoS differentiation capabilities, i.e., enough resources for service provision at different QoS levels and suitable interfaces for requesting a specific QoS level. QoS can be differentiated at various granularity levels (e.g., per service request or per service request class [30]) and with various motivations (e.g., revenue maximization subject to a Service Level Agreement, or optimization of selected Clients' performance under resource overload [15]). Each service request has a *native class* attribute representing the QoS level it is entitled to.

We consider two proxy agents, *Demand Proxy* (DProxy) and *Assign Proxy* (AProxy), acting respectively on behalf of Client and Server, and connected by a communication system (Fig. 1). For a service request of native class n, DProxy decides the

E-mail address: jekon@eti.pg.edu.pl

https://doi.org/10.1016/j.ins.2022.06.024
0020-0255/

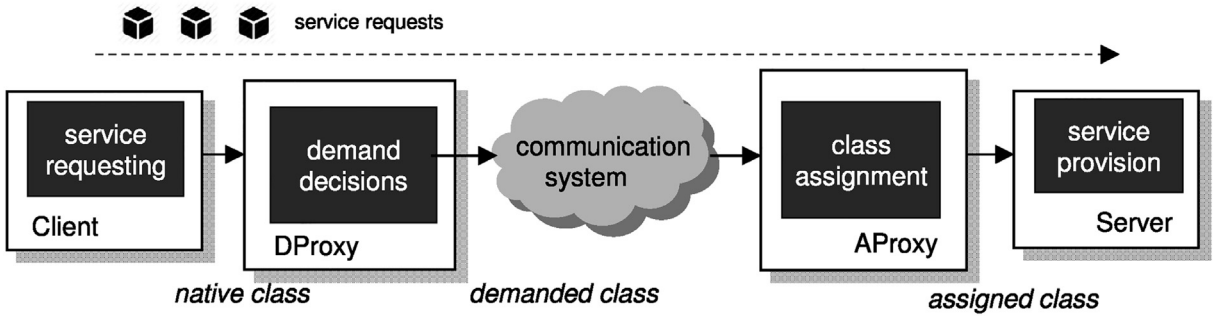


Fig. 1. Information system model involving Client, DProxy, AProxy, and Server.

demanded class d and passes the request to AProxy, which decides the *assigned class a*, i.e., the QoS level to be granted at Server. Occasional demanding of $d > n$ is termed *QoS abuse* [39]. It results in Client's unduly high QoS perception and wastage of Server's resources, thus warrants occasional assignment of $a < d$ at AProxy. In a QoS abuse scenario, DProxy aims to maximize Client's QoS perception, whereas AProxy protects Server by minimizing assignment of undue QoS. Examples of QoS abuse include falsifying a data packet priority to expedite its transfer through a multiservice IP network [40] or a Delay Tolerant Network [42], stating a false task execution deadline to expedite task offloading in a fog computing environment [3], or posing as a "trusted friend" to access a richer content repository in a Social Internet of Things [9] or a MEC system [43].

Known defenses against QoS abuse employ request inspection at AProxy to verify n, resource overprovisioning at Server, or heuristic incentive mechanisms. We add a strategic perspective to QoS abuse by explicitly accounting for the request inspection cost. To reduce it, AProxy is inclined to occasionally skip request inspection, i.e., *trust* a service request, in which case $a = d$. QoS abuse behavior that builds on this inclination will be called *Fake VIP attack* (FVA), a $d > n$ event being called an *FVA episode*.[1] DProxy's *FVA strategy* aims to instill trust in AProxy so as to maximize the frequency of successful FVA episodes ($a = d > n$ events). AProxy's defense is a *trust strategy* that dictates trust/inspection decisions so as to optimally trade request inspection cost for the loss caused by successful FVA episodes. (Note that FVA can be viewed both as a new variety of QoS abuse and a framework for systematic defense against QoS abuse.) We cast FVA as a Stackelberg security game in which AProxy, the leader, commits to a trust strategy anticipating a best-reply FVA strategy adopted by DProxy, the follower.

We realistically assume that either agent's per service-request behavior (honest/FVA episode and trust/inspection) is non-observable to the other: n is not revealed to AProxy and cannot be reliably verified due to imperfect request inspection, whereas DProxy is not notified of a and only has long-term QoS perception. Hence, either agent's strategy must only rely on characteristics observable to her; the game lacks perfect information and standard equilibrium analyses or request-by-request learning fail. Our approach allows for untraceable Clients, and abstracts from the underlying service model and performance indicators. This precludes comparisons with generic QoS abuse scenarios, but enables a systematic design and evaluation of defense solutions if the request inspection cost is known. We offer the following contributions:

- We formalize the notion of FVA and formulate modeling postulates for a *scant-transparency* setting, in which per service-request decisions and local variables controlling them are either agent's private information.
- We design a probabilistic FVA strategy and a trust strategy based on *double-blind reputation*, and derive the two agents' long-term utilities reflecting the cost of request inspection and undue QoS assignment.
- For two distinguished QoS levels we find the Stackelberg equilibrium (SE) and demonstrate the two agents' utility improvement vis á vis a reference trust strategy; the impact of the request inspection cost, strategy parameters, and information leakage is also shown.

The remainder of the paper is organized as follows. In Section 2 we outline related work. Section 3 presents the proposed treatment of FVA, describes a scant-transparency security framework, and the agents' strategies and utilities. In Section 4 we analyze the Stackelberg FVA game and characterize its SE. Numerical and experimental illustration is provided in Section 5. Section 6 presents conclusions and briefly discusses open issues.

2. Related Work

QoS abuse by an attacker Client undermines Server's QoS policies and honest Clients' QoS perception [39]. Generic QoS abuse has not been treated systematically. For some environments, such as multiservice packet networks under Traffic Remapping attacks [40] or public cloud systems under Economic Denial of Sustainability (EDoS) attacks [33], heuristic defense solutions have been proposed. They can be classified as prevention, attribution, mitigation, mustering extra resources, payment-based, and game-theoretic.

An example of *prevention* is a multiservice packet network guarded by a Token Bucket [36], where entry traffic suspected of QoS abuse is downgraded to a lower native class. Another preventive solution is Client authorization [26,33]. *Attribution*

[1] The term FVA was coined in conference papers [27,28] preceding this work.

consists in recognizing attacker Clients based on resource usage histories [19], feature extraction from request streams [12,13,16], or current CPU utilization [1]. Such solutions are environment-specific and nonviable with untraceable Clients; they prescribe inspection of each service request, hence regard the involved cost as negligible. *Mitigation* employs per service-request parsing or challenge-response exchange [2]; the latter uses image-recognition Turing tests [29] or proof-of-work cryptographic puzzles [7,32]. Some of these solutions are conscious of the request inspection cost (e.g., puzzle difficulty is adjusted depending on the likelihood of QoS abuse), though its influence is not quantified. *Mustering extra resources* [45] dispenses with request inspection, regarding the involved cost as prohibitive. This relieves the strain caused by QoS abuse, but can only be made economical given a detailed service model, which our approach abstracts from. *Payment-based* defenses, common for Internet service providers, charge Client for demanding a high QoS level; a Vickrey-Clarke-Groves payment-based variety was proposed for wireless networks [10]. Such solutions assume that the demanded QoS level is known prior to request inspection, which need not be true; they also require a trusted authority, which may be unavailable.

The notion of FVA stresses the strategic aspect of QoS abuse and mandates *game-theoretic* solutions, where QoS perception is a player's utility. E.g., in [11] too frequent requests for a high QoS level are punished with lower utilities due to blockage at a virtual firewall; this may lead to undeserved punishment of some requests, which our model disallows. A reciprocity game arises if multiple Clients share a critical resource, e.g., a wireless channel; a victim of QoS abuse can then degrade the attacker's QoS perception by responding in kind [40]. Our approach is less restrictive in that no shared resource is assumed. In a full- rather than scant-transparency setting, FVA can be encompassed by several game models. E.g., each service request triggers a two-stage game in which AProxy responds with a trust/inspection decision to DProxy's decision d. However, optimum strategies forming a perfect equilibrium [37] can only be found if d is observable to AProxy, which need not be true for a trusted request, and if the request inspection cost is known to DProxy, which cannot be assumed. In the Contract game model [17], a learner (DProxy) attempts to influence an adversary's (AProxy's) play so as to produce frequent coincidences of FVA episodes and trust decisions. Such an FVA strategy fares well against computationally bounded trust strategies, but fails under scant transparency. Finally, the online prediction model [5] envisages a zero-sum game in which the learner minimizes her total regret due to ill-decided d (e.g., a low QoS level when the service request is trusted), while the adversary (AProxy) maximizes it by adjusting her decisions on a. However, scant transparency invalidates this model.

FVA under scant transparency was first considered in [27,28]. Both these preliminary works cast QoS abuse as a two-person Stackelberg security game under uncertainty. In such games, defense restricts feasible attack scenarios so that an optimum attack causes the defender minimum loss [6,22,41]. Applications range from wireless networks [24] to cyber-physical systems [44] to cloud systems [20]. However, scant transparency invalidates perfect-information and information-limitations models [4,41]. Bayesian inference is used in [34], where the defender infers the attacker's type from successive attack targets to update her target protection strategy. The attacker may benefit by imitating a different type, which necessitates sophisticated defense. Identification of attacked targets implies reliable detection of FVA episodes, which our approach does not require. In [47], the attacker advances toward a target vertex in a conceptual attack graph with a success probability associated with each edge. Upon attack detection, the defender performs Bayesian updating of currently protected edges, assuming that the attacker will follow an optimum unprotected path. Our approach does not require prior knowledge of attack scenarios or success probabilities.

In summary, known approaches to QoS abuse: (a) often pose stringent, environment-specific requirements, (b) do not explicitly account for the request inspection cost, thus losing from sight the prospect of FVA, and (c) lack analytical foundations in the presence of scant transparency. In view of these shortcomings we design a trust strategy ensuring that frequently suspected FVA episodes diminish the frequency of trust decisions. This is achieved by a double-blind reputation scheme calibrated so as to maximize AProxy's long-term utility at a Stackelberg equilibrium.

In organizational design, control vs. trust has been a topic of a long-standing debate [18]. Our analysis, weighing undue QoS assignment against the cost of request inspection, contributes to an avenue of thought captured by the phrase "control is good, but trust is cheaper" [21].

3. FVA Treatment

In this section we outline a scant-transparency setting and describe the proposed treatment of FVA. We formulate a model of service request handling and derive from it the two agents' strategies and utilities. We use the notation $(q_1, \ldots, q_K)$ for a K-tuple of quantities $q_k, k = 1, \ldots, K$.

3.1. Service Request Stream

Two native classes, *low* (L) and *high* (H) are distinguished.[2] Service requests are handled in the order they are issued at Client, and their native classes follow a stationary memoryless random process with $\rho = Pr[n = \mathrm{H}], 0 < \rho < 1$. To verify a native class, AProxy employs a request inspection scheme producing a signature $s \in \{\mathrm{L}, \mathrm{H}\}$ based on a request's features and a classifier function. A *signature error* can occur due to feature confusion or classifier ambiguities at AProxy. Let $\epsilon_{\mathrm{L}} = Pr[s = \mathrm{H}|n = \mathrm{L}]$ and

[2] Generalization to multiple QoS levels leads to similar but more tedious calculations.

$\epsilon_{\mathrm{H}} = Pr[s = \mathrm{L}|n = \mathrm{H}]$ be the signature error rates, and let a per service-request cost C model the handling overhead involved in request inspection. The assigned class $a \in \{\mathrm{L}, \mathrm{H}\}$ depends on the trust/inspection decision; the latter is influenced by DProxy's current reputation r, which AProxy updates on a request-by-request basis. DProxy employs a signature prediction scheme returning a service request's *predicted signature* $p \in \{\mathrm{L}, \mathrm{H}\}$. Let $\zeta_{\mathrm{L}} = Pr[p = \mathrm{H}|s = \mathrm{L}]$ and $\zeta_{\mathrm{H}} = Pr[p = \mathrm{L}|s = \mathrm{H}]$ be the signature prediction error rates.

3.2. Scant-Transparency Setting

To provide a conservative security framework we let the following modeling postulates govern the handling of service requests at DProxy and AProxy:

MP1. Assignment of $a = \mathrm{H}$ is costly for AProxy and beneficial for DProxy.
MP2. Demanding $d = \mathrm{H}$ is costless for DProxy.
MP3. C, r, s, and a are not revealed to DProxy.
MP4. n and p are not revealed to AProxy.
MP5. AProxy is unaware of d at trust/inspection decision time and cannot learn d if a service request is trusted.
MP6. DProxy is aware of Client's long-term QoS perception quantified by $Pr[a = \mathrm{H}]$.
MP7. $(s, a) = (\mathrm{H}, \mathrm{L})$ is disallowed at AProxy.

In particular, MP2 implies that provision of high QoS is not subject to charges or limits, hence FVA is costless. MP3 reflects the view that request-by-request feedback to DProxy would be impractical, and that C depends on the local implementation of request inspection, which is AProxy's private information. MP4 reflects the presence of signature errors and the fact that the signature prediction scheme is DProxy's private information. MP5 covers the case when d only reveals itself during subsequent service at Server, details of which are not reported to AProxy. MP6 states that Client's long-term QoS perception is proportional to the average assigned QoS level; this is reasonable if service requests represent independent tasks or queries. Finally, MP7 implies that AProxy honors agreed-upon service provision contracts and protects service requests from undeserved punishment.

These postulates can be justified in various security contexts. E.g., in a public cloud system under EDoS [33], attackers issue resource-greedy service requests to raise a cloud consumer's bill. Client's attacker vs. legitimate nature and the demanded scale-up/scale-out of cloud resources correspond to n and d, respectively. Typically, no request-by-request feedback from the cloud runtime environment to the admission control is available, hence MP3 is valid. The cloud admission control cannot verify n, as MP4 stipulates, since service requests from attackers and legitimate Clients are formatted identically. MP5 is valid, since the necessary scale-up/scale-out can reveal itself only during service, after a trust/inspection decision has been taken by the cloud admission control. (In [27], MP1–MP7 are justified for a multiservice wireless network under a Traffic Remapping attack [40].).

MP3, MP4, and MP5 constitute a scant-transparency setting, where either agent's per service-request decisions and local characteristics are her private information, namely n, p, and d at DProxy, and s, a, r, and C at AProxy. Only d is observable to both agents, and only if request inspection is decided at AProxy, as summarized in Table 1. Moreover, one expects that request inspection is a public-knowledge scheme, whereas the signature prediction scheme is DProxy's private information. This in particular implies that $\rho, (\epsilon_{\mathrm{L}}, \epsilon_{\mathrm{H}})$, and $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$ are only known to DProxy.

3.3. FVA and Trust Strategies

If C is large enough, AProxy is inclined to occasionally trust a service request; exploiting this inclination sets FVA apart from general QoS abuse. Observe that MP1 and MP2 create incentives for FVA. MP5 and MP7 preclude straightforward defense invoking request inspection only if $d = \mathrm{H}$ or threatening with undeserved punishment, i.e., $(s, a) = (\mathrm{H}, \mathrm{L})$. Finally, MP4 and MP6 thwart request-by-request learning of the other agent's strategy. FVA strategy is represented by a decision rule $\sigma(\cdot)$ dictating d at DProxy, and a trust strategy is a decision rule $\tau(\cdot)$ dictating trust/inspection decisions. We design a probabilistic FVA strategy of the form $\sigma(n, p, random\_event)$ and a reputation-based trust strategy of the form $\tau(d, s, r)$.

3.3.1. Probabilistic FVA Strategy

DProxy decides d based on (n, p) and a biased coin toss:

$$d = \sigma(n, p, random\_event) = \begin{cases} \mathrm{H}, & rand_{x_{\mathrm{NP}}}, \\ \mathrm{L}, & \text{otherwise}, \end{cases} \tag{1}$$

where $rand_{\alpha}$ denotes a random event occurring with probability α, and $x_{\mathrm{NP}} = Pr[d = \mathrm{H}|(n, p) = (\mathrm{N}, \mathrm{P})], (\mathrm{N}, \mathrm{P}) \in \{\mathrm{L}, \mathrm{H}\}^2$, are parameters of the FVA strategy. An $(n, d) = (\mathrm{L}, \mathrm{H})$ event signifies an FVA episode. Let

Table 1
Observability of relevant characteristics.

Symbol	Characteristic	Observable at	
		DProxy	AProxy
n	native class	yes	no
p	predicted signature	yes	no
d	demanded class	yes	yes [*]
s	signature	no	yes [*]
a	assigned class	no	yes
r	current reputation state	no	yes
C	request inspection cost	no	yes

[*] Only if request inspection is decided.

$$\begin{aligned} x &= Pr[d=\mathrm{H}] = \sum_{(\mathrm{N,P})\in\{\mathrm{L,H}\}^2} Pr[(n,p)=(\mathrm{N,P})]x_{\mathrm{NP}} \\ &= (1-\rho)((1-\epsilon_{\mathrm{L}})(1-\zeta_{\mathrm{L}})+\epsilon_{\mathrm{L}}\zeta_{\mathrm{H}})x_{\mathrm{LL}} + \rho(\epsilon_{\mathrm{H}}(1-\zeta_{\mathrm{L}})+(1-\epsilon_{\mathrm{H}})\zeta_{\mathrm{H}})x_{\mathrm{HL}} + (1-\rho)((1-\epsilon_{\mathrm{L}})\zeta_{\mathrm{L}}+\epsilon_{\mathrm{L}}(1-\zeta_{\mathrm{H}}))x_{\mathrm{LH}} \\ &\quad +\rho(\epsilon_{\mathrm{H}}\zeta_{\mathrm{L}}+(1-\epsilon_{\mathrm{H}})(1-\zeta_{\mathrm{H}}))x_{\mathrm{HH}}. \end{aligned} \tag{2}$$

E.g., $x=1$, i.e., $(x_{\mathrm{LL}},x_{\mathrm{HL}},x_{\mathrm{LH}},x_{\mathrm{HH}})=(1,1,1,1)$, corresponds to persistent FVA, whereas $(x_{\mathrm{LL}},x_{\mathrm{HL}},x_{\mathrm{LH}},x_{\mathrm{HH}})=(0,0,1,1)$ combined with $(\zeta_{\mathrm{L}},\zeta_{\mathrm{H}})=(0,0)$ mark DProxy's *benchmark* behavior; the latter implies $d=p=s$ and $Pr[d=\mathrm{H}]=Pr[s=\mathrm{H}]=1-\omega$, where

$$\omega = Pr[s=\mathrm{L}] = (1-\rho)(1-\epsilon_{\mathrm{L}})+\rho\epsilon_{\mathrm{H}}. \tag{3}$$

FVA strategy *aggressiveness* can be measured by

$$Pr[d=\mathrm{H}]-Pr[s=\mathrm{H}] = x-1+\omega. \tag{4}$$

An $(s,d)=(\mathrm{L,H})$ event signifies a suspected FVA episode, with

$$\begin{aligned} Pr[(s,d)=(\mathrm{L,H})] &= \sum_{(\mathrm{N,P})\in\{\mathrm{L,H}\}^2} Pr[(n,s,p)=(\mathrm{N,L,P})]x_{\mathrm{NP}} \\ &= (1-\rho)(1-\epsilon_{\mathrm{L}})(1-\zeta_{\mathrm{L}})x_{\mathrm{LL}} + \rho\epsilon_{\mathrm{H}}(1-\zeta_{\mathrm{L}})x_{\mathrm{HL}} + (1-\rho)(1-\epsilon_{\mathrm{L}})\zeta_{\mathrm{L}}x_{\mathrm{LH}} + \rho\epsilon_{\mathrm{H}}\zeta_{\mathrm{L}}x_{\mathrm{HH}}. \end{aligned} \tag{5}$$

Benchmark behavior yields $Pr[(s,d)=(\mathrm{L,H})]=0$.

3.3.2. Reputation-Based Trust Strategy

A reputation scheme at AProxy defines DProxy's reputation states $r\in\{0,1,2,\ldots\}$; $r<t$ and $r\geqslant t$ are called *trust states*, and *non-trust states*, respectively, where $t\geqslant 1$ is a parameter. In a trust state, an arriving service request is trusted, otherwise request inspection is invoked:

$$a=\tau(d,s,r)=\begin{cases} d, & r<t, \\ s, & r\geqslant t. \end{cases} \tag{6}$$

DProxy's reputation is raised or lowered (r is decremented or incremented, respectively) on a request-by-request basis. If $r\geqslant t$ and $(s,d)=(\mathrm{L,H})$ (i.e., an FVA episode is suspected) then r is incremented, while $d=\mathrm{L}$ is perceived as honest behavior and r is decremented. If $r<t$ then s and d are non-observable at AProxy; r is then incremented with probability Q and decremented with probability $1-P$, where $Q\approx Pr[(s,d)=(\mathrm{L,H})]$ and $P\approx Pr[d=\mathrm{H}]=x$ are estimates based on previous occurrences of $(s,d)=(\mathrm{L,H})$ and $d=\mathrm{H}$ in non-trust states. An additional parameter $y\in[0,1]$ calibrates AProxy's inclination to decrement r. Formally,

$$r\leftarrow\begin{cases} r+1, (r<t\wedge rand_{yQ})\vee(r\geqslant t\wedge rand_{y}\wedge(s,d)=(\mathrm{L,H}), r-1, (0<r<t\wedge rand_{(1-y)(1-P)}) \\ \quad\vee(r\geqslant t\wedge rand_{1-y}\wedge d=\mathrm{L}), r, \text{otherwise}. \end{cases} \tag{7}$$

The intuition behind (7) is that frequently suspected FVA episodes diminish their success rate as r is driven away from the trust states, whereas frequent perception of DProxy's honest behavior drives r toward the trust states, i.e., encourages AProxy to economize on request inspection. Fig. 2 depicts the per service-request workflow at AProxy. The presence of signature errors and y justifies MP3 in that DProxy is unable to keep track of r. Thus the reputation scheme is *double-blind*: r is non-observable at DProxy and n is non-observable at AProxy.[3]

[3] We call r reputation even though it is not made public as standard definitions stipulate [25], since it stems from DProxy's past behavior and influences AProxy's trust decisions.

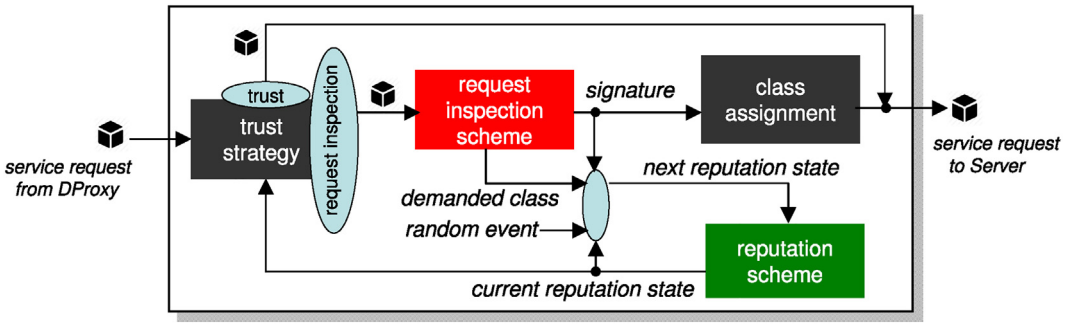


Fig. 2. Service request-related workflow at AProxy.

3.4. Reputation State Probabilities

Reputation states induced by the memoryless service request stream of Section 3.1 follow a homogeneous random walk on $\{0,1,2,\ldots\}$, with transition probabilities shown in Fig. 3 and stationary probabilities $\pi_i = Pr[r = i]$. Call $\Pi = \pi_0 + \ldots + \pi_{t-1}$ the *probability of trust*. If $y = 0$, we take $\Pi = 1$. If $y = 1$, we reason as follows. Assume that the initial reputation state is any trust state. With $(\zeta_L, \zeta_H) = (0,0)$ and y approaching 1, DProxy's best reply disallows $(s,d) = (L,H)$, i.e., the initial trust state persists forever. However, this prompts $(s,d) = (L,H)$, hence is off-equilibrium. In Section 4.3 we show that at equilibrium, $\pi_i = \frac{1}{t+1}$ for $i \in \{0,\ldots,t\}$, and so $\Pi = \frac{t}{t+1}$. We envisage that upon a service request arrival, r is drawn at random from $\{0,\ldots,t\}$ if $Q = 0$ and $rand_y$ occurs; otherwise r is updated according to (7) (cf. Algorithm 1). Hence, if $y = 0$, the initial trust state persists forever, and if $y = 1, r$ randomizes within $\{0,\ldots,t\}$ until the first occurrence of $(s,d) = (L,H)$ in reputation state t, after which no trust state is ever visited. Such operation resembles the *Grim* strategy [37]: a player cooperates until her opponent defects, whereupon she ceases to cooperate. Under DProxy's best reply to $y = 1$, cooperation lasts forever.

If $0 < y < 1$, we take $\Pi = 0$ if the random walk is transient or null recurrent. If it is positive recurrent, local balance yields $yPr[(s,d) = (L,H)]\pi_i = (1-y)(1-x)\pi_{i+1}, i \in \{0,1,2,\ldots\}$, and using $\pi_0 + \pi_1 + \ldots = 1$ we get

$$\Pi = 1 - \left(\frac{y}{1-y} \cdot \frac{Pr[(s,d) = (L,H)]}{1-x}\right)^t. \tag{8}$$

Algorithm 1: Reputation scheme operation

```
r ← any trust state      //initialization
Q ← 0
P ← 0
foreach arrived service request do
    if rand_y ∧ Q = 0 then
        if rand_{t/(t+1)} then
            r ← any trust state
        else
            r ← t
        end
    else
        update r according to (7)
    end
    if r ≥ t then
        update Q and P
    end
end
```

3.5. Utilities

The two agents' utilities will be derived as mean per service-request benefits and losses related to undue QoS assignment and request inspection costs.

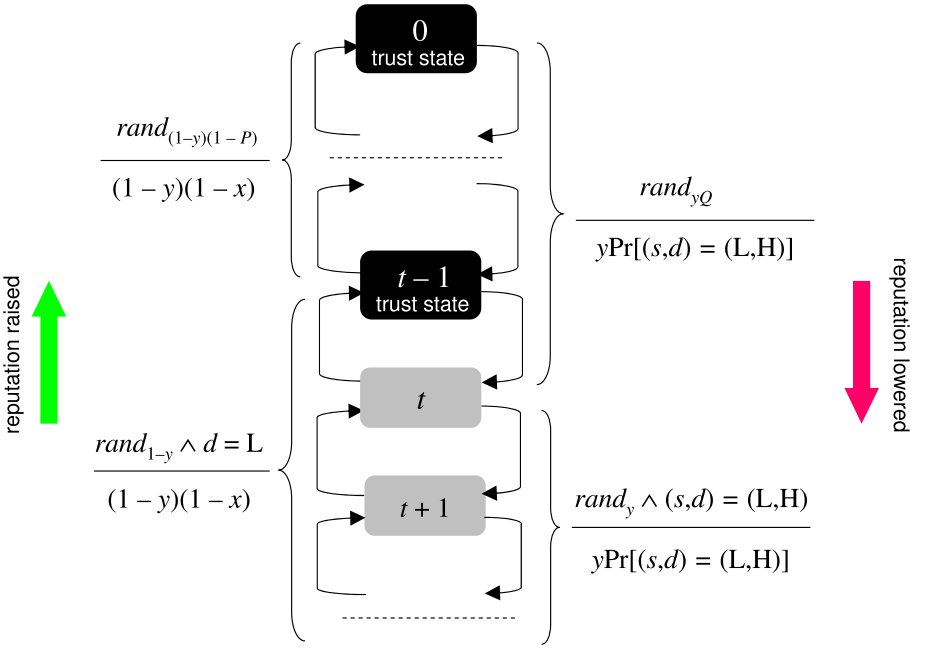


Fig. 3. Reputation state transitions; triggering events are indicated above the horizontal bars, beneath are the corresponding transition probabilities.

3.5.1. AProxy's Utility

AProxy registers a per service-request loss amounting to a unit loss of undue QoS assignment in a trust state when $a \neq s$,[4] and the request inspection cost C in a non-trust state. Using $Pr[d = \mathrm{H}] = x$ and $Pr[s = \mathrm{H}] = 1 - \omega$, we calculate the expected loss as

$$\begin{aligned} loss_{\mathrm{AProxy}} &= (1-\Pi)C + Pr[(s,a) = (\mathrm{L},\mathrm{H})] + Pr[(s,a) = (\mathrm{H},\mathrm{L})] \\ &= (1-\Pi)C + (Pr[(s,d) = (\mathrm{L},\mathrm{H})] + Pr[(s,d) = (\mathrm{H},\mathrm{L})])\Pi \\ &= (1-\Pi)C + (1-\omega-x+2Pr[(s,d) = (\mathrm{L},\mathrm{H})])\Pi. \end{aligned} \tag{9}$$

We consider two reference trust strategies:

- *never-trust*, whereby AProxy trusts no service request, i.e., reputation is not employed, and so (9) evaluates to C, and
- *always-trust*, whereby AProxy trusts each service request, i.e., request inspection is not employed either; this prompts DProxy to reply with $x = 1$, producing $Pr[(s,d) = (\mathrm{L},\mathrm{H})] = \omega$, and so (9) evaluates to ω.

A *trivial* trust strategy coincides with never-trust if $C \leqslant \omega$ and with always-trust if $C > \omega$, hence $loss_{\mathrm{AProxy}}|_{\mathrm{trivial}} = \min\{C, \omega\}$. (Note that it is an optimum trust strategy among ones using a *fixed* probability of trust. Also, if $C > \omega$, request inspection is pointless without a non-trivial trust strategy.) AProxy's utility is defined as

$$\begin{aligned} u_{\mathrm{AProxy}} &= loss_{\mathrm{AProxy}}|_{\mathrm{trivial}} - loss_{\mathrm{AProxy}} \\ &= \min\{0, \omega - C\} + (C + x - 1 + \omega - 2Pr[(s,d) = (\mathrm{L},\mathrm{H})])\Pi. \end{aligned} \tag{10}$$

AProxy estimates (10) from the observed frequency of trust states and the statistics of s and d observed in non-trust states, which, under DProxy's unawareness of r, are independent of r. Even if t, y, and (7) are public knowledge, DProxy cannot infer (10) without the knowledge of C.

3.5.2. DProxy's Utility

DProxy registers a unit per service-request benefit when $(n,a) = (\mathrm{L},\mathrm{H})$, and a unit loss when $(n,a) = (\mathrm{H},\mathrm{L})$. The expected net benefit is

$$\begin{aligned} net\_benefit_{\mathrm{DProxy}} &= Pr[(n,a) = (\mathrm{L},\mathrm{H})] - Pr[(n,a) = (\mathrm{H},\mathrm{L})] \\ &= Pr[a = \mathrm{H}] - Pr[n = \mathrm{H}]. \end{aligned} \tag{11}$$

[4] Here, s is the signature that would have been detected in a non-trust state. In fact, undue QoS assignment occurs when $a \neq n$; however, n is non-observable at AProxy.

In particular, $net\_benefit_{\text{DProxy}}|_{\text{never-trust}} = Pr[s = \text{H}] - Pr[n = \text{H}]$. Also, $Pr[a = \text{H}] = Pr[d = \text{H}]\Pi + Pr[s = \text{H}](1 - \Pi)$. Define DProxy's utility as

$$\begin{aligned} u_{\text{DProxy}} &= net\_benefit_{\text{DPproxy}} - net\_benefit_{\text{DProxy}}|_{\text{never-trust}} \\ &= Pr[a = \text{H}] - Pr[s = \text{H}] = (x - 1 + \omega)\Pi. \end{aligned} \tag{12}$$

Note that (12) is the product of (4) and (8); using (3), DProxy can infer it from $Pr[s = \text{H}] = 1 - \omega$ and Client's long-term QoS perception $Pr[a = \text{H}]$.

4. Analysis of FVA Game

In this section we analyze a Stackelberg security game [41] between AProxy, the leader, and DProxy, the follower. From Section 3.2 it follows that the per service-request game with honest/FVA episode and trust/inspection actions lacks perfect information even up to known observation errors; therefore, a static game with long-term utilities (10) and (12) is considered. AProxy's trust strategy $\hat{y} \in [0, 1]$ maximizes (10) in anticipation of a best-reply FVA strategy $\hat{x}(y) \in [0, 1]$ maximizing (12) given y. A *Stackelberg equilibrium* (SE) is derived to characterize the effectiveness of defense against FVA as mediated by C. Several special scenarios of the game are also analyzed.

4.1. Best-Reply Probabilistic FVA Strategy

To maximize (12) given y, DProxy first minimizes $Pr[(s, d) = (\text{L}, \text{H})]$ subject to $Pr[d = \text{H}] = x$, and next sets x optimally. The former task amounts to a linear program over $x_{\text{NP}} \in [0, 1], \text{N}, \text{P} \in \{\text{L}, \text{H}\}$:

$$\text{minimize } Pr[(s, d) = (\text{L}, \text{H})] = \sum_{(\text{N},\text{P}) \in \{\text{L},\text{H}\}^2} Pr[(n, s, p) = (\text{N}, \text{L}, \text{P})] x_{\text{NP}} \tag{13}$$

$$\text{subject to } Pr[d = \text{H}] = \sum_{(\text{N},\text{P}) \in \{\text{L},\text{H}\}^2} Pr[(n, p) = (\text{N}, \text{P})] x_{\text{NP}} = x, \tag{14}$$

where the coefficients in (13) and (14) are specified in (5) and (2), respectively. Denote the minimum of (13) by $Pr[(s, d) = (\text{L}, \text{H})]_{\min}(x)$. Clearly, $Pr[(s, d) = (\text{L}, \text{H})]_{\min}(1) = \omega$. To find a solution for $x < 1$, let us order the ratios of respective coefficients in (13) and (14): define $\text{N}_j, \text{P}_j \in \{\text{L}, \text{H}\}$ and $\kappa_j = Pr\big[(n, s, p) = (\text{N}_j, \text{L}, \text{P}_j)\big] / Pr\big[(n, p) = (\text{N}_j, \text{P}_j)\big] \leqslant 1$ so that $\kappa_1 \leqslant \ldots \leqslant \kappa_4$, with ties resolved arbitrarily. For $i \in \{0, \ldots, 4\}$, define $\Phi_i = \sum_{1 \leqslant j \leqslant i} Pr\big[(n, p) = (\text{N}_j, \text{P}_j)\big]$ and $\varphi_i = \sum_{1 \leqslant j \leqslant i} Pr\big[(n, s, p) = (\text{N}_j, \text{L}, \text{P}_j)\big]$. (Note that $\varphi_0 = \Phi_0 = 0, \varphi_4 = \omega$, and $\Phi_4 = 1$.)

Proposition 1. For $\Phi_i \leqslant x < \Phi_{i+1}, i = 0, \ldots, 3$, the minimum of (13) is given by a piecewise linear function of x:

$$Pr[(s, d) = (L, H)]_{min}(x) = (x - \Phi_i)\kappa_{i+1} + \varphi_i \tag{15}$$

and is attained at

$$x_{\text{N}_j\text{P}_j} = 1 \text{ for } j \leqslant i, x_{\text{N}_{i+1}\text{P}_{i+1}} = \frac{x - \Phi_i}{\Phi_{i+1} - \Phi_i}, \; x_{\text{N}_j\text{P}_j} = 0 \text{ for } j > i. \tag{16}$$

Proof. Compare $Pr[(s, d) = (\text{L}, \text{H})]$ at $(x_{\text{LL}}, x_{\text{HL}}, x_{\text{LH}}, x_{\text{HH}})$, where $x_{\text{N}_j\text{P}_j} > 0$ and $x_{\text{N}_{j'}\text{P}_{j'}} < 1$ for some N_j, P_j and $\text{N}_{j'}, \text{P}_{j'}$ such that $j > j'$, with that upon an incremental transfer $x_{\text{N}_j\text{P}_j} \leftarrow x_{\text{N}_j\text{P}_j} - \delta, x_{\text{N}_{j'}\text{P}_{j'}} \leftarrow x_{\text{N}_{j'}\text{P}_{j'}} + \delta'(\delta, \delta' > 0)$ that keeps x unchanged. That is, $-\delta Pr\big[(n, p) = (\text{N}_j, \text{P}_j)\big] + \delta\prime Pr\big[(n, p) = (\text{N}_{j\prime}, \text{P}_{j\prime})\big] = 0$ and the resulting change in $Pr[(s, d) = (\text{L}, \text{H})]$ is

$$\begin{aligned} &\delta\prime Pr\big[(n, s, p) = (\text{N}_{j\prime}, \text{L}, \text{P}_{j\prime})\big] - \delta Pr\big[(n, s, p) = (\text{N}_j, \text{L}, \text{P}_j)\big] \\ &= -\delta Pr\big[(n, p) = (\text{N}_j, \text{P}_j)\big](\kappa_{j\prime} - \kappa_j) \leqslant 0. \end{aligned}$$

Hence, $Pr[(s, d) = (\text{L}, \text{H})]$ is minimized when no incremental transfer as above is possible, in other words when $x_{\text{N}_{i+1}\text{P}_{i+1}} > 0$ implies $x_{\text{N}_j\text{P}_j} = 1$ for $j \leqslant i$. Recalling (14), we arrive at (15) and (16). □

Proposition 1 implies that a best-reply FVA strategy is fully characterized by x. Calculation shows that if $\frac{\epsilon_{\text{L}}}{1-\epsilon_{\text{L}}} \cdot \frac{\epsilon_{\text{H}}}{1-\epsilon_{\text{H}}} < 1$ and $\frac{\zeta_{\text{L}}}{1-\zeta_{\text{L}}} \cdot \frac{\zeta_{\text{H}}}{1-\zeta_{\text{H}}} < 1$ then $(\text{N}_1, \text{P}_1) = (\text{H}, \text{H})$ and $(\text{N}_4, \text{P}_4) = (\text{L}, \text{L})$. Thus by (16), DProxy always demands H if $(n, p) = (\text{H}, \text{H})$ unless she never demands H if $(n, p) \neq (\text{H}, \text{H})$; moreover, she never demands H if $(n, p) = (\text{L}, \text{L})$ unless she always demands H if $(n, p) \neq (\text{L}, \text{L})$. If in addition we have $\frac{\epsilon_{\text{L}}}{1-\epsilon_{\text{L}}} \cdot \frac{\epsilon_{\text{H}}}{1-\epsilon_{\text{H}}} > \frac{\zeta_{\text{L}}}{1-\zeta_{\text{L}}} \cdot \frac{\zeta_{\text{H}}}{1-\zeta_{\text{H}}}$ then $(\text{N}_2, \text{P}_2) = (\text{L}, \text{H})$ and $(\text{N}_3, \text{P}_3) = (\text{H}, \text{L})$. That is, DProxy can demand H if $p = \text{L}$ only if she always demands H if $p = \text{H}$.

We redefine DProxy's utility (12) and define her best-reply correspondence as follows:

$$U_{\text{DProxy}}(x,y) = (x-1+\omega)\left(1-\left(\frac{yg(x)}{1-y}\right)^t\right), \tag{17}$$

$$\hat{x}(y) \in \text{argmax}_{x\in[0,1]} U_{\text{DProxy}}(x,y), \tag{18}$$

where $g(x) = Pr[(s,d) = (\text{L},\text{H})]_{\min}(x)/(1-x)$. It is easy to check that if $g(1-\omega) \geqslant \frac{1}{y} - 1$ for a given y then $U_{\text{DProxy}}(x,y) \leqslant 0$ for all $x \in [0,1]$; in that case, $\hat{x}(y) = 1-\omega$ is the unique best reply. In the opposite case the best reply is also unique (hence, (18) is a function), as stated below.

Proposition 2. If $U_{\text{DProxy}}(x,y) > 0$ for some $x \in [0,1]$ then DProxy's best reply $\hat{x}(y)$ is unique and a decreasing function of y.

Proof. Clearly, $\hat{x}(y) > 1-\omega$ and y must be such that the second factor in (17) is positive. Let $g'(x)$ be the (piecewise continuous) derivative of $g(x)$. Substituting (8) and $g(x)$ into (17), and recalling (15), we find $U_{\text{DProxy}}(x,y)$ piecewise differentiable with respect to x. At $x \neq \Phi_i$,

$$\frac{\partial U_{\text{DProxy}}(x,y)}{\partial x} = 1 - \left(\frac{y}{1-y}\right)^t \left(g^t(x) + (x-1+\omega)tg^{t-1}(x)g'(x)\right). \tag{19}$$

From (15) we conclude that the function $g(\cdot)$ is continuous, nonnegative and increasing in x, with $g(0) = 0$ and infinite $g(1)$, and $g'(x)$ is nonnegative and nondecreasing in x. Hence for $x > 1-\omega$, the bracketed sum in the derivative (19) is a nonnegative increasing function of x, continuous except at $x = \Phi_i$, and with arbitrarily large positive values as x approaches 1. Therefore, (19) is decreasing in x, with discontinuities at $x = \Phi_i$, and is positive at $x = 1-\omega$, since $U_{\text{DProxy}}(1-\omega,y) = 0$. It follows that $U_{\text{DProxy}}(x,y)$ is quasi-concave with a unique $\hat{x}(y) > 1-\omega$ occurring either when (19) becomes 0 or at $x = \Phi_i$. Since (19) is decreasing in y, so is $\hat{x}(y)$. □

Using (15) and (12) we get $\hat{x}(y) = \text{argmax}_{i\in\{0,1,2,3\}} U_{\text{DProxy}}(x_i,y)$, where

$$x_i \in \text{argmax}_{x\in[\Phi_i,\Phi_{i+1}]}(x-1+\omega)\left(1-\left(\frac{y}{1-y}\cdot\frac{(x-\Phi_i)\kappa_{i+1}+\varphi_i}{1-x}\right)^t\right).$$

If the random walk (7) is not positive recurrent, (8) can nevertheless be applied in conjunction with (12) to yield $\hat{x}(0) = 1$ and $\hat{x}(1) = 1-\omega$.

The following proposition states that larger y disincentivize FVA more.

Proposition 3. $U_{\text{DProxy}}(\hat{x}(y),y)$ is nonincreasing in y.

Proof. We will make use of the following fact. Let $\mathbb{I}$ be a compact interval in $\mathbb{R}$ and $f : \mathbb{R}^2 \to \mathbb{I}$ be a continuous bivariate function monotonous in one argument: $f(w,z) \leqslant (\geqslant) f(w,\bar{z})$ for all $w,z,\bar{z}$ with $z \leqslant \bar{z}$. Assume that $\hat{w}(z) \in \text{argmax}_w f(w,z)$. Then $f(\hat{w}(z),z) \leqslant (\geqslant) f(\hat{w}(\bar{z}),\bar{z})$. Indeed, for the $\leqslant$ case, $f(\hat{w}(z),z) \leqslant f(\hat{w}(z),\bar{z})$ by assumption and $f(\hat{w}(z),\bar{z}) \leqslant f(\hat{w}(\bar{z}),\bar{z})$ by definition; for the $\geqslant$ case the reasoning is similar. This fact will be referred to as the *dominance argument* with respect to w. The proposition follows by applying to (12) the dominance argument with respect to x. □

4.2. SE Trust Strategy

Upon the optimization specified by Proposition 1, the strategy profile becomes (x,y). Analogously to (17), redefine AProxy's utility (10) as

$$U_{\text{AProxy}}(C,x,y) = \min\{0,\omega-C\} + (C-(1-x)(1+2g(x))+\omega)\left(1-\left(\frac{yg(x)}{1-y}\right)^t\right) \tag{20}$$

(note that $\lim_{x\to 1}(1-x)g(x) = Pr[(s,d) = (\text{L},\text{H})]_{\min}(1) = \omega$). Further we explicitly indicate the dependence of AProxy's SE trust strategy on C.

Definition 1. With redefined utilities (17) and (20), SE occurs at $(\hat{x},\hat{y}(C))$ such that

$$\hat{x} = \hat{x}(\hat{y}(C)), \tag{21}$$

$$\hat{y}(C) \in \text{argmax}_{y\in[0,1]} U_{\text{AProxy}}(C,\hat{x}(y),y). \tag{22}$$

Clearly, the search of an SE trust strategy is irrelevant if C does not admit a positive AProxy's utility, i.e., if $U_{\text{AProxy}}(C,\hat{x}(y),y) \leqslant 0$ for all $y \in [0,1]$, so assume otherwise. In some cases, including all the numerical examples in Section 5.1, $\hat{y}(C)$ is unique – namely if $\partial U_{\text{DProxy}}(x,y)/\partial x = 0$ at $x \geqslant 1-\omega$ such that $x \neq \Phi_i$. We then have from (19):

$$1 - \left(\frac{yg(\hat{x}(y))}{1-y}\right)^t = \frac{t(\hat{x}(y)-1+\omega)g'(\hat{x}(y))}{g(\hat{x}(y)) + t(\hat{x}(y)-1+\omega)g'(\hat{x}(y))}, \tag{23}$$

and the mapping between y and $\hat{x}(y)$ is one-to-one (for suppose that y varies and x does not, then the left-hand side of (23) varies and the right-hand side does not). Substituting (23) into (20) and recalling (15), one can express $U_{\text{AProxy}}(C,\hat{x}(y),y)$ as a rational function of $\hat{x}(y)$ whose maximization over $\hat{x}(y) \in [1-\omega,1]$, equivalent of searching for an SE trust strategy, yields a single value in the generic case. If (22) admits multiple maximizers then we assume that besides maximizing (10), AProxy attempts to minimize DProxy's utility (12); by Proposition 3, she selects the largest maximizer of (22). Combined with Definition 1, the above proviso ensures a unique SE.

In general, trusting service requests can worsen AProxy's loss with respect to the trivial trust strategy and improve DProxy's benefit with respect to the never-trust strategy. The following proposition states that at the SE only the latter is possible, and that the SE utilities are monotonous in C.

Proposition 4. At the SE, DProxy's utility is nondecreasing in C, whereas AProxy's utility is nonnegative and unimodal in C, with a peak at $C = \omega$.

Proof. Based on (20) one verifies that for the trivial trust strategy, i.e., $y = \mathbf{1}_{C\leqslant\omega}$, where $\mathbf{1}_{(\cdot)}$ is the indicator function, we have $U_{\text{AProxy}}(C,\hat{x}(y),y) = 0$ for any C; this is because $\hat{x}(1) = 1-\omega, g(1-\omega) \geqslant 0$, and $\hat{x}(0) = 1$. Hence, by (22), AProxy's SE utility never goes negative. Recall that by Proposition 2, $\hat{x}(y)$ is unique for a given y. Consider (20) as a function of C and y. In the case of $C \leqslant \omega$ it is nondecreasing in C and in the case of $C \geqslant \omega$, nonincreasing in C; hence, by the dominance argument with respect to y, so is $U_{\text{AProxy}}(C,\hat{x}(\hat{y}(C)),\hat{y}(C))$. Since these two cases coincide at $C = \omega$, the second part of the proposition follows. The fact that for $C \leqslant \omega$, (20) is nondecreasing in C and at the same time nonincreasing in y implies that $\hat{y}(C)$ is nonincreasing in C. Finally, DProxy's utility (17) is a function of x and y, nonincreasing in y. Substituting $y = \hat{y}(C)$ and by the dominance argument with respect to y, we find that $U_{\text{DProxy}}(\hat{x}(\hat{y}(C)),\hat{y}(C))$ is nondecreasing in C except for a discontinuity at $C = \omega$. □

It follows that for large enough C, AProxy's SE utility becomes zero and the trivial trust strategy is used. For $(\zeta_{\text{L}},\zeta_{\text{H}}) = (0,0)$, the critical value of C is easy to derive, cf. Proposition 6.

4.3. Controlled Signature Prediction Errors

One conjectures that DProxy should be interested in accurate signature prediction, i.e., $(\zeta_{\text{L}},\zeta_{\text{H}}) = (0,0)$. Yet calculation of optimum signature prediction error rates maximizing DProxy's SE utility for a given C, i.e.,

$$(\zeta_{\text{L}}^*,\zeta_{\text{H}}^*)(C) \in \text{argmax}_{(\zeta_{\text{L}},\zeta_{\text{H}})\in[0,1]^2} U_{\text{DProxy}}\left(\hat{x}(\hat{y}(C))|_{(\zeta_{\text{L}},\zeta_{\text{H}})},\hat{y}(C)\right), \tag{24}$$

in general yields $(\zeta_{\text{L}},\zeta_{\text{H}}) \neq (0,0)$, cf. Section 5.1. Suppose now that DProxy is able to control $(\zeta_{\text{L}},\zeta_{\text{H}})$. E.g., having somehow obtained an accurate prediction $p = s$ for a given service request of native class n, she deliberately falsifies it by taking $\bar{p} \neq s$ with a probability possibly depending on n, and next sets d according to (1). Thus the $(\zeta_{\text{L}},\zeta_{\text{H}})$ turn from exogenous parameters into parameters of the FVA strategy to control deviations from benchmark behavior. We will use the notation $(\cdot)|_{(\zeta_{\text{L}},\zeta_{\text{H}})}$ to indicate dependence on $(\zeta_{\text{L}},\zeta_{\text{H}})$. Best-reply signature prediction error rates are defined as

$$\left(\hat{\zeta}_{\text{L}},\hat{\zeta}_{\text{H}}\right)(y) \in \text{argmax}_{(\zeta_{\text{L}},\zeta_{\text{H}})\in[0,1]^2} U_{\text{DProxy}}\left(\hat{x}(y)|_{(\zeta_{\text{L}},\zeta_{\text{H}})},y\right). \tag{25}$$

It turns out that the maximizers of (25) differ from (24). Namely, we have:

Proposition 5. For any $y \in [0,1]$, $\left(\hat{\zeta}_L,\hat{\zeta}_H\right)(y) = (0,0)$.

Proof. Consider (12) as a function of x and $Pr[(s,d) = (\text{L},\text{H})]$. Using the dominance argument with respect to x it is enough to show that given $x, Pr[(s,d) = (\text{L},\text{H})]_{\min}(x)|_{(0,0)} \leqslant Pr[(s,d) = (\text{L},\text{H})]_{\min}(x)|_{(\zeta_{\text{L}},\zeta_{\text{H}})}$ for any $(\zeta_{\text{L}},\zeta_{\text{H}}) \in [0,1]^2$. Recall that $Pr[(s,d) = (\text{L},\text{H})]_{\min}(x)|_{(\zeta_{\text{L}},\zeta_{\text{H}})}$ is nondecreasing and piecewise linear in x, with successive slopes $\kappa_1 \leqslant \ldots \leqslant \kappa_4$ and $Pr[(s,d) = (\text{L},\text{H})]_{\min}(1)|_{(\zeta_{\text{L}},\zeta_{\text{H}})} = \omega$. Moreover, for $(\zeta_{\text{L}},\zeta_{\text{H}}) = (0,0)$ we have $(\kappa_1,\ldots,\kappa_4) = (0,0,1,1), (\Phi_1,\ldots\Phi_4) = (\rho(1-\epsilon_{\text{H}}), 1-\omega, 1-\omega+\rho\epsilon_{\text{H}}, 1)$ and $(\varphi_1,\ldots,\varphi_4) = (0,0,\Phi_3,\omega)$, leading to $Pr[(s,d) = (\text{L},\text{H})]_{\min}(x)|_{(0,0)} = \max\{0, x-1+\omega\}$. Let $(\zeta_{\text{L}},\zeta_{\text{H}}) \neq (0,0)$ and suppose $Pr[(s,d) = (\text{L},\text{H})]_{\min}(x_0)|_{(\zeta_{\text{L}},\zeta_{\text{H}})} < Pr[(s,d) = (\text{L},\text{H})]_{\min}(x_0)|_{(0,0)} = x_0 - 1 + \omega$ for some

$x_0 > 1-\omega$. The average slope of $Pr[(s,d)=(\mathrm{L},\mathrm{H})]_{\min}(x)|_{(\zeta_\mathrm{L},\zeta_\mathrm{H})}$ in the range $x \in [x_0, 1]$ is therefore greater than $(\omega - (x_0 - 1 + \omega))/(1-x_0) = 1$, which is impossible, since $\kappa_j \leqslant 1$ by definition. □

To calculate the resulting DProxy's SE utility we note that the strategy profile now has the form $((x, \zeta_\mathrm{L}, \zeta_\mathrm{H}), y)$; an SE is defined by (22) and

$$\text{FVA strategy} = \left(\hat{x}, \hat{\zeta}_\mathrm{L}, \hat{\zeta}_\mathrm{H}\right)(\hat{y}(C)). \tag{26}$$

Given DProxy's best reply prescribed by Proposition 5, one can show that if C is not too large to admit a positive AProxy's utility then AProxy's SE trust strategy is Grim; otherwise it is always-trust.

Proposition 6. At the SE defined by (22) and (26), $\hat{y}(C) = 1$ if $C \leqslant (t+1)\omega$, otherwise $\hat{y}(C) = 0$.

Proof. By Proposition 2, $\hat{x}(y)|_{(0,0)}$ is decreasing in y. For $(\zeta_\mathrm{L}, \zeta_\mathrm{H}) = (0,0)$ we calculate $g(x) = \frac{x-1+\omega}{1-x}$ for $x \geqslant 1-\omega$, cf. the proof of Proposition 5. Substituting this into (19) and equating to 0 yields $\left(\frac{yg(x)}{1-y}\right)^t (1 + \frac{t\omega}{1-x}) = 1$. This implies that $\hat{x}(y)|_{(0,0)}$ moreover satisfies

$$1 - \left(\frac{yg\left(\hat{x}(y)|_{(0,0)}\right)}{1-y}\right)^t = \frac{t\omega}{1 - \hat{x}(y)|_{(0,0)} + t\omega}. \tag{27}$$

From (20) we now get

$$U_{\mathrm{AProxy}}\left(C, \hat{x}(y)|_{(0,0)}, y\right) = \min\{0, \omega - C\} + t\omega \frac{C - \omega + 1 - \hat{x}(y)|_{(0,0)}}{1 - \hat{x}(y)|_{(0,0)} + t\omega},$$

which is nondecreasing in $-\hat{x}(y)|_{(0,0)}$, hence also in y, if $C \leqslant (t+1)\omega$, and decreasing otherwise. (In particular, for $t=1$ calculation yields $\hat{x}(y)|_{(0,0)} = 1 - \omega\sqrt{y}$ and $U_{\mathrm{AProxy}}\left(C, \hat{x}(y)|_{(0,0)}, y\right) = \min\{0, \omega - C\} + \omega + \frac{C-2\omega}{1+\sqrt{y}}$.) □

Thus except for $C > (t+1)\omega$, DProxy's SE strategy is $\left(\hat{x}, \hat{\zeta}_\mathrm{L}, \hat{\zeta}_\mathrm{H}\right)(\hat{y}(C)) = (1-\omega, 0, 0)$: optimal control of the signature prediction error rates disincentivizes FVAs, since DProxy's best reply reduces to benchmark behavior and does not yield a positive SE utility. That is, DProxy's follower position is weakened. At the same time, the probability of trust does not tend to 0 even as y approaches 1. Indeed, recalling that $\lim_{y\to 1} \hat{x}(y)|_{(0,0)} = 1-\omega$, we have from (27) that $\lim_{y\to 1}\left(1 - \left(\frac{yg\left(\hat{x}(y)|_{(0,0)}\right)}{1-y}\right)^t\right) = \frac{t}{t+1}$. Likewise, $\lim_{y\to 1}(\pi_0 + \ldots + \pi_{i-1}) = \frac{i}{t+1}$ for $i \in \{1, \ldots, t\}$. Due to the fact that $Pr[(s,d)=(\mathrm{L},\mathrm{H})]_{\min}(1-\omega)|_{(0,0)} = 0$, reputation state transitions beyond t are impossible, thus the limiting probability distribution of reputation states is uniform over $i \in \{0, \ldots, t\}$. By (20), AProxy's SE utility for $C \leqslant (t+1)\omega$ becomes $\min\{0, \omega - C\} + C\frac{t}{t+1}$, i.e., positive except at $C = (t+1)\omega$.

4.4. Effects of Multiple Trust States and Reputation Information Leakage

Under fixed x and y, the utilities (10) and (12) increase with t. At the SE, this influence depends on C due to SE locations varying with t, cf. Fig. 4: increasing t, while a mixed blessing for DProxy, is always beneficial for AProxy. Thus AProxy can increase her SE utility without incentivizing FVA more. However, larger t imply longer spells of skipped request inspection, which DProxy might exploit. Indeed, using first-passage probabilities [35] we find the mean uninterrupted sojourn time of the random walk (7) within $\{0, \ldots, t-1\}$ to be $\frac{1}{yQ}\sum_{0\leqslant i<t}\left(\frac{1-y}{y}\cdot\frac{1-P}{Q}\right)^i$.

Moreover, if DProxy recognizes $r < t$ contrary to MP3 (e.g., acquires AProxy's pseudorandom number generator seed to follow (7)), she can adopt a *trust-aware* FVA strategy: set $d = \mathrm{H}$ if $r < t$, otherwise use (1). Then (12) and (10) become

$$\begin{aligned} U_{\mathrm{DProxy}} &= \Pi + Pr[s=\mathrm{H}](1-\Pi) - Pr[s=\mathrm{H}] = \omega\Pi, \\ U_{\mathrm{AProxy}} &= \min\{C, \omega\} - ((1-\Pi)C + Pr[s=\mathrm{L}]\Pi) = (C-\omega)(\Pi - \mathbf{1}_{C>\omega}). \end{aligned} \tag{28}$$

Compared to (1), the trust-aware FVA strategy distinctly improves DProxy's utility and drives AProxy's utility below that yielded by the trivial trust strategy, cf. Section 5.1; this bears out the importance of preventing reputation information leakage. If the estimates P and Q are accurate, the reputation state statistics under the trust-aware FVA strategy and under (1) are identical. Furthermore, s and d are non-observable to AProxy in trust states. Thus AProxy is unable to detect a deviation from a probabilistic FVA strategy.

4.5. SE Gauging and Cost Information Leakage

Without the knowledge of ρ, (ϵ_L, ϵ_H), and (ζ_L, ζ_H), AProxy is unable to calculate $\hat{x}(\cdot)$ from (18), hence to set $\hat{y}(C)$ directly using (22). Instead, she can perform *SE gauging* through multistage play, with a fixed M service requests per stage. At the

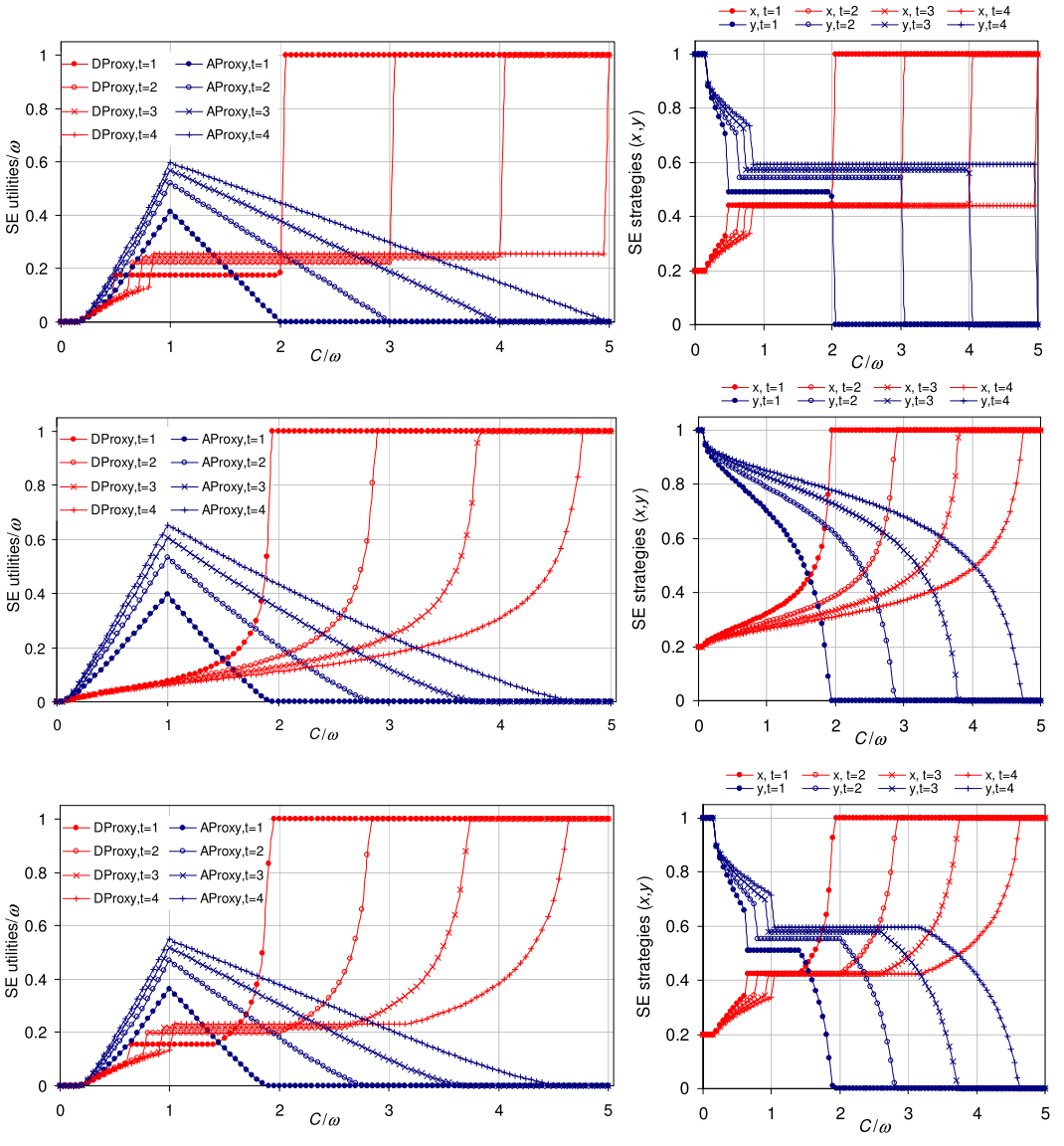


Fig. 4. SE utilities (*left*) and strategies (*right*); $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$ = (0.3,0) (*top*), (0,0.3) (*middle*), (0.3,0.3) (*bottom*).

start of stage $k \in \{1, \ldots, K-1\}$, AProxy announces the trust strategy $y^{(k)} = k/K$ she will commit to in this stage, and DProxy sets $\hat{x}(y^{(k)})$ using (18). Subsequently, AProxy records the frequencies $\Omega^{(k)}, P^{(k)}$, and $Q^{(k)}$ of $s = \mathrm{L}, d = \mathrm{H}$, and $(s,d) = (\mathrm{L},\mathrm{H})$ events in non-trust states, respectively. At the end of stage k, AProxy's estimates (20) as

$$U^{(k)}_{\mathrm{AProxy}}(y^{(k)}) = \min\left\{0, \Omega^{(k)} - C\right\} + \left(C - \left(1 - P^{(k)}\right) - 2Q^{(k)} + \Omega^{(k)}\right)\left(1 - \left(\frac{y^{(k)}}{1-y^{(k)}} \cdot \frac{Q^{(k)}}{1-P^{(k)}}\right)^{t}\right). \tag{29}$$

For all post-SE-gauging stages $k \geqslant K, y^{(k)} = y^{(\hat{k})} = \hat{k}/K$ is set, where $\hat{k} \in \mathrm{argmax}_{k \in \{1,\ldots,K-1\}} \tilde{U}^{(k)}_{\mathrm{AProxy}}(y^{(k)})$ and $\tilde{(\cdot)}$ represents a smoothening operation, e.g., moving average. For large enough K and M, AProxy's approximation of her SE utility becomes accurate in that the relative difference

$$\left(\frac{1}{J}\sum_{0 \leqslant j \leqslant J} U^{(\hat{k})}_{\mathrm{AProxy}}\left(y^{(\hat{k})}, j\right) / U_{\mathrm{AProxy}}(C, \hat{x}(\hat{y}(C)), \hat{y}(C)) - 1\right) \cdot 100\% \tag{30}$$

becomes arbitrarily small as J increases, where $U^{(\hat{k})}_{\mathrm{AProxy}}\left(y^{(\hat{k})}, j\right)$ is the SE utility estimated after j^{th} SE gauging. On the other hand, K and M should not be too large to prevent unnecessarily long SE gauging.

Suppose now that DProxy acquires knowledge of C prior to the SE gauging, hence can calculate $\hat{y}(C)$ from (22). She can now improve her utility, and worsen AProxy's, using a *false-reply* function $FR(\cdot)$ instead of $\hat{x}(\cdot)$, i.e., setting $FR(y^{(k)}) \neq \hat{x}(y^{(k)})$ in stage k. Let us denote $v = (\rho, (\epsilon_{\mathrm{L}}, \epsilon_{\mathrm{H}}))$ and use the notation $(\cdot)|_{v,(\zeta_{\mathrm{L}},\zeta_{\mathrm{H}})}$ to indicate dependence on v and $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$. DProxy falsifies her reply by precalculating $\hat{x}(y)|_{\bar{v},(\bar{\zeta}_{\mathrm{L}},\bar{\zeta}_{\mathrm{H}})}$ from (18) using some $\bar{v} = (\bar{\rho}, (\bar{\epsilon}_{\mathrm{L}}, \bar{\epsilon}_{\mathrm{H}})) \neq v$ and $(\bar{\zeta}_{\mathrm{L}}, \bar{\zeta}_{\mathrm{H}}) \neq (\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$. The value of ω given by (3) must be preserved in $\bar{v}$, otherwise AProxy can detect the falsification. Indeed, analyzing (20) as well as the estimates $\Omega^{(k)}$, $P^{(k)}$, and $Q^{(k)}$, AProxy can calculate the function $Pr[(s,d) = (\mathrm{L},\mathrm{H})]_{\min}(\cdot)$ given by (15), and find that no $(\bar{v}, (\bar{\zeta}_{\mathrm{L}}, \bar{\zeta}_{\mathrm{H}}))$ produces the observed $U^{(k)}_{\mathrm{AProxy}}(y^{(k)})$. Let

$$V_\omega = \{\bar{v} | 0 < \bar{\rho} < 1 \wedge \bar{\epsilon}_{\mathrm{L}}, \bar{\epsilon}_{\mathrm{H}} < 0.5 \wedge (1-\bar{\rho})(1-\bar{\epsilon}_{\mathrm{L}}) + \bar{\rho}\bar{\epsilon}_{\mathrm{L}} = \omega\}. \tag{31}$$

If AProxy sets y, DProxy anticipates $U_{\mathrm{DProxy}}\left(\hat{x}(y)|_{\bar{v},(\bar{\zeta}_{\mathrm{L}},\bar{\zeta}_{\mathrm{H}})}, y\right)|_{v,(\zeta_{\mathrm{L}},\zeta_{\mathrm{H}})}$, and moreover assumes that $y^{(K)} \approx \bar{y} := \hat{y}(C)|_{\bar{v},(\bar{\zeta}_{\mathrm{L}},\bar{\zeta}_{\mathrm{H}})}$. Then she falsifies her reply so as to maximize her anticipated post-SE-gauging utility, i.e., takes $FR(y) = \hat{x}(y)|_{v^*,(\zeta^*_{\mathrm{L}},\zeta^*_{\mathrm{H}})}$, where

$$(v^*, (\zeta^*_{\mathrm{L}}, \zeta^*_{\mathrm{H}})) \in \mathrm{argmax}_{\bar{v}\in V_\omega, (\bar{\zeta}_{\mathrm{L}},\bar{\zeta}_{\mathrm{H}})\in[0,1]^2} U_{\mathrm{DProxy}}\left(\hat{x}(\bar{y})|_{\bar{v},(\bar{\zeta}_{\mathrm{L}},\bar{\zeta}_{\mathrm{H}})}, \bar{y}\right)|_{v,(\zeta_{\mathrm{L}},\zeta_{\mathrm{H}})}. \tag{32}$$

Since the calculation of (32) subsumes (24) as a special case $\bar{v} = v$, DProxy's false-reply produced utility is not less than her SE utility at $(\zeta^*_{\mathrm{L}}, \zeta^*_{\mathrm{H}})$.

5. Numerical and Experimental Illustration

In this section we present sample numerical results of the analysis of Section 4 and describe experimental results to compare a memoryless service request stream with a real-world internally correlated one.

5.1. Numerical Results

For numerical illustration we take $\rho = 0.125, (\epsilon_{\mathrm{L}}, \epsilon_{\mathrm{H}}) = (0.1, 0.1)$ (yielding $\omega = 0.8$), and various $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$ and t. Fig. 4 (*left*) depicts DProxy's and AProxy's SE utilities (17) and (20), normalized to ω, for various $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}}) \neq (0,0)$ ($(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}}) = (0,0)$ is discussed in Section 4.3). AProxy's SE utility curves owe their quasi-triangular shape to the fact that at $C = \omega$ the trivial trust strategy changes from never-trust to always-trust. By (10), the trivial trust strategy would yield AProxy a zero utility, while DProxy's SE utility would be $\omega \cdot \mathbf{1}_{C>\omega}$. The double-blind reputation scheme is uniformly beneficial to AProxy, notably around $C = \omega$. For $C \leqslant \omega$ it yields a win–win, inducing DProxy to launch FVA with restraint and AProxy to trust service requests frequently. Since $\hat{y}(C)$ is nonincreasing in C, $\hat{x}(\hat{y}(C))$ is nondecreasing in C, cf. Fig. 4 (*right*). Both these functions are constant for some ranges of C, causing a similar feature in DProxy's SE utility curves.

In Fig. 5 (*left*), the optimum error rates (24) are plotted against C/ω. For $C \leqslant \omega$ they are of the form $(z, 0)$, where $z > 0$ increases with C, and for a subrange of $C > \omega$, of the form $(0, z)$, where $z > 0$ decreases with C. The corresponding SE utilities, depicted in Fig. 5 (*right*), show a distinct improvement for DProxy compared with Fig. 4 (*left*). As mentioned in Section 4.3, if the $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$ are exogenous to DProxy then $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}}) \neq (0,0)$ is desirable for her, as opposed to the case when DProxy is in control of $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$ (cf. Proposition 5).

To illustrate the effects of multiple trust states (Section 4.4), Fig. 6 (*left*) shows how increasing t lengthens the spells of skipped request inspections, assuming that the P and Q estimates are accurate at the SE. For a fair comparison of the probabilistic and trust-aware FVA strategies we substitute into (28) the SE strategies optimizing (12) and (10). Fig. 6 (*right*) confirms the adverse influence of increased t from AProxy's viewpoint when $C \leqslant \omega$, whereas for $C > \omega$ the nature of this influence depends on C.

To illustrate the accuracy of SE gauging (Section 4.5), Monte Carlo simulations were conducted taking K = 50, M = 500 or 1000, and moving averages over stages $\{k-4, \ldots, k+4\}$. Fig. 7 presents sample plots of $\tilde{U}^{(k)}_{\mathrm{AProxy}}(y^{(k)})$ against $y^{(k)} = k/K$. The dashed curves correspond to $U_{\mathrm{AProxy}}(C, \hat{x}(y^{(k)}), y^{(k)})$, i.e., accurate estimation of (20). Though the SE gauging is slightly inaccurate ($y^{(k)} \neq \hat{y}(C)$), the plot of (20) is satisfactorily recreated by $\tilde{U}^{(k)}_{\mathrm{AProxy}}(y^{(k)})$ if $M = 1000$. For 10000 independent game repetitions with J = 50, Table 2 shows the average relative differences (30), and percentages of game repetitions in which (30) was below 5% and above 20%.

Fig. 8 plots the post-SE-gauging utilities produced by DProxy's best reply (21) and false reply (32). As expected, DProxy's utility exceeds that in Fig. 5 (*right*): leakage of information on C prior to the SE gauging permits DProxy to optimize her false reply, which is visibly harmful to AProxy.

5.2. Experimental Validation

In contrast with the memoryless model of Section 3.1, real-world service request streams can exhibit internal correlation. To investigate its impact, we have conducted experimental validation of the presented modeling framework in an IP networking environment. We found that

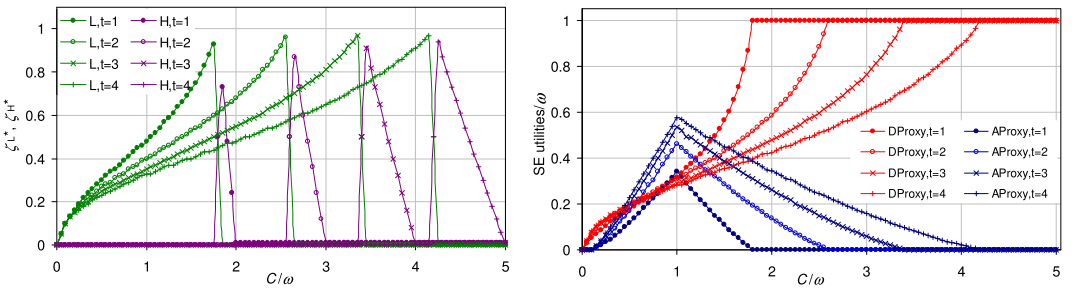


Fig. 5. Optimum signature prediction error rates (*left*) and corresponding SE utilities (*right*).

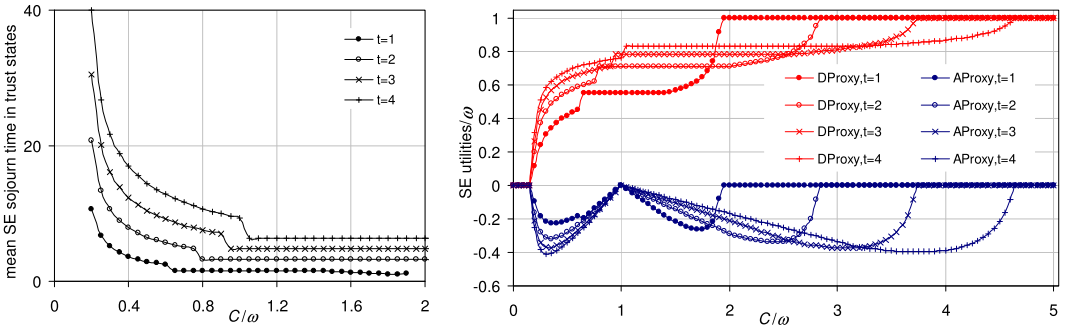


Fig. 6. Mean uninterrupted sojourn time within $\{0,\ldots,t-1\}$ (*left*), SE utilities for trust-aware FVA (*right*); $(\zeta_L,\zeta_H)=(0.3,0.3)$.

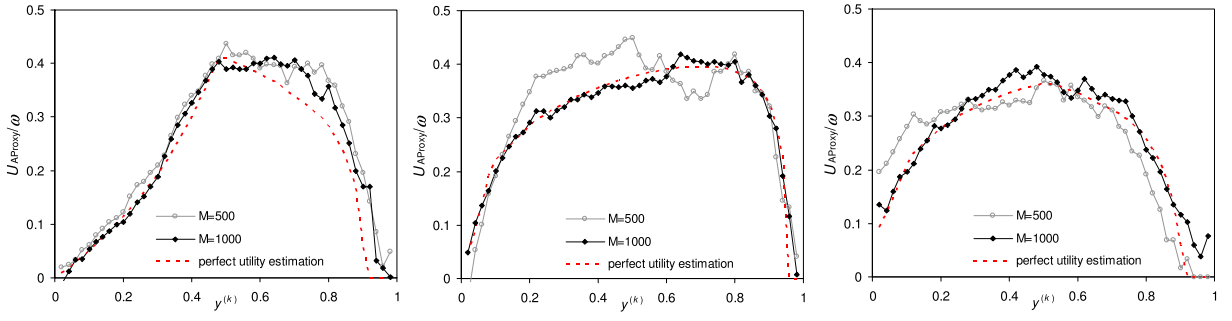


Fig. 7. Smoothened AProxy's utility estimates during SE gauging; (ζ_L,ζ_H) = (0.3,0) (*left*), (0,0.3) (*middle*), (0.3,0.3) (*right*).

Table 2
Relative differences (30) in 10000 game repetitions; K = 50, J = 50.

	Average of (30)	Repetitions with (30)...	
		below 5%	above 20%
$(\zeta_L,\zeta_H)=(0.3,0),\ M=500$	5.8%	69.7%	3.7%
$M=1000$	3.2%	82.3%	0.4%
$(\zeta_L,\zeta_H)=(0,0.3),\ M=500$	9.6%	56.6%	7.2%
$M=1000$	6.4%	57.6%	2.8%
$(\zeta_L,\zeta_H)=(0.3,0.3),\ M=500$	9.6%	56.6%	7.2%
$M=1000$	6.4%	57.6%	2.8%

- the double-blind reputation scheme remains beneficial to AProxy and incentivizes her to seek an SE, and
- the memoryless model captures salient features of the two agents' utilities as functions of FVA and trust strategy parameters.

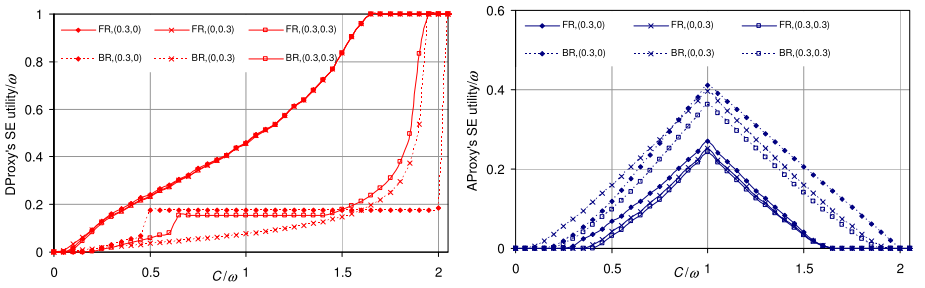


Fig. 8. Post-SE-gauging utilities, DProxy's (*left*) and AProxy's (*right*); BR – best reply, FR – false reply, $t = 1, (\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$ indicated in legend.

We used the Canadian Institute for Cybersecurity CIC-DDoS2019 public dataset [8]. It records a large number of IP packet flows targeting a small-size IP domain as part of a Distributed Denial of Service (DDoS) attack. Each flow, regarded as a service request issued by a conceptual DProxy, was entitled to a high or low QoS level at the target domain representing AProxy. A C# script excerpted I flows from the dataset and filtered useful attributes of each flow $i \in \{1, \ldots, I\}$: $LABEL_i \in \{benign, DDoS\}$ – nature of flow as determined by the target domain's intrusion detection system, DST_i – destination address, S_i – start time, DUR_i – duration of activity, and PKT_i – number of constituent packets. For each flow i, the number of concurrent flows with the same destination address was derived as $CONCUR_i = |\{j = 1, \ldots, I | DST_j = DST_i \wedge [S_j, S_j + DUR_j] \cap [S_i, S_i + DUR_i] \neq \varnothing\}|$. Upon the above preprocessing, the dataset $((LABEL_i, CONCUR_i, PKT_i), i = 1, \ldots, I)$ was aligned with our model based on the following principles:

- $LABEL_i$ is a ground-truth attribute akin to native class and only *benign* flows are entitled to the high QoS level. Hence, $n = \mathrm{H}$ if $LABEL_i$ = *benign* and $n = \mathrm{L}$ if $LABEL_i$ = *DDoS*.
- At AProxy, $CONCUR_i$ is compared with a threshold thr_{s}, and class-H signatures are detected for sub-threshold flows, as motivated by $CONCUR_i$ averaging 1.44 and 6.83 for *benign* and *DDoS* flows, respectively. Hence, $s = \mathrm{H}$ if $CONCUR_i < thr_{\mathrm{s}}$, otherwise $s = \mathrm{L}$.
- At DProxy, PKT_i is compared with a threshold thr_{p} and class-H signatures are predicted for sub-threshold flows, as motivated by PKT_i averaging 1.84 and 143.54 for *benign* and *DDoS* flows, respectively. Hence, $p = \mathrm{H}$ if $PKT_i < thr_{\mathrm{p}}$, otherwise $p = \mathrm{L}$.

Using a succinct notation $\%(condition_i) = |\{i = 1, \ldots, I | condition_i = \texttt{true}\}|/I$, one estimates:

$$
\begin{aligned}
&\rho \approx \%(LABEL_i = benign),\\
&\omega \approx \%(CONCUR_i \geqslant thr_{\mathrm{s}}),\\
&x \approx \%\big(PKT_i < thr_{\mathrm{p}}\big),\\
&Pr[(s,d) = (\mathrm{L},\mathrm{H})] \approx \%\big(CONCUR_i \geqslant thr_{\mathrm{s}} \wedge PKT_i < thr_{\mathrm{p}}\big),\\
&\epsilon_{\mathrm{L}} \approx \%(LABEL_i = DDoS \wedge CONCUR_i < thr_{\mathrm{s}})/(1-\rho),\\
&\epsilon_{\mathrm{H}} \approx \%(LABEL_i = benign \wedge CONCUR_i \geqslant thr_{\mathrm{s}})/\rho,\\
&\zeta_{\mathrm{L}} \approx \%\big(CONCUR_i \geqslant thr_{\mathrm{s}} \wedge PKT_i < thr_{\mathrm{p}}\big)/\omega,\\
&\zeta_{\mathrm{H}} \approx \%\big(CONCUR_i < thr_{\mathrm{s}} \wedge PKT_i \geqslant thr_{\mathrm{p}}\big)/(1-\omega).
\end{aligned}
$$

Thus ω and $(\epsilon_{\mathrm{L}}, \epsilon_{\mathrm{H}})$ are determined by thr_{s}, x by thr_{p}, and $(\zeta_{\mathrm{L}}, \zeta_{\mathrm{H}})$ and $Pr[(s,d) = (\mathrm{L},\mathrm{H})]$ jointly by thr_{s} and thr_{p}. Table 3 presents the experiment setup along with the estimated characteristics. The dataset-based stream $(LABEL_i, i = 1, \ldots, I)$ exhibits strong internal correlation, as measured by the average *DDoS* spell length of 140.5 (a *DDoS* spell of length l features $LABEL_{i-1} \neq LABEL_i = \ldots = LABEL_{i+l-1} = DDoS \neq LABEL_{i+l}$). A memoryless stream with the same characteristics as in Table 3 was also generated according to the model of Section 3.1, with the average *DDoS* spell length $1/\rho = 58.8$. Both stream types were input to the reputation scheme (7) and class assignment (6) to obtain the utilities (10) and (12) via Monte Carlo simulation. The adopted deterministic FVA strategy introduces dependence between x and $(\zeta_{\mathrm{H}}, \zeta_{\mathrm{H}})$, therefore the obtained utility values differ from those in Section 5.1. On the other hand, by eliminating randomness from (1), it enables a fair comparison of the two stream types.

For $y \in [0,1]$ and $x = 0.019..0.997$ corresponding to $thr_{\mathrm{p}} = 2..201$, Fig. 9 depicts the obtained utilities $u_{\mathrm{DProxy}}(x,y)$ and $u_{\mathrm{AProxy}}(x,y)$. Neither their positivity (which merits the use of double-blind reputation) nor monotonicity (which determines the SE location) is affected by the stream's internal correlation. As y increases, i.e., the trust strategy moves toward never-trust, the drift away from the trust state becomes stronger, cf. Fig. 3. Hence, both utilities decrease. For the same reason $u_{\mathrm{AProxy}}(x,y)$ decreases with x, i.e., as $d = \mathrm{H}$ is demanded more aggressively, while $u_{\mathrm{DProxy}}(x,y)$ increases for small y, when the drift is weak, and decreases for large y, when the drift strengthens. Thus the $u_{\mathrm{DProxy}}(x,y)$ curves intersect and DProxy's best reply $\hat{x}(y) = \mathrm{argmax}_{x \in [0,1]} u_{\mathrm{DProxy}}(x,y)$ varies with y, cf. (18).

Table 3
Experiment setup and estimated characteristics.

CIC-DDoS2019 dataset file	DrDoS_DNS 1.csv
I	$15,000 \Rightarrow \rho \approx 0.017$
thr_s	$3 \Rightarrow \omega \approx 0.957, (\epsilon_L, \epsilon_H) \approx (0.028, 0.076)$
thr_p	$2..201 \Rightarrow x \approx 0.019..0.997$
FVA strategy	$(x_{LL}, x_{HL}, x_{LH}, x_{HH}) = (0, 0, 1, 1) \Rightarrow d = p$
Trust strategy	$t = 1, y \in [0, 1]$
C	0.6

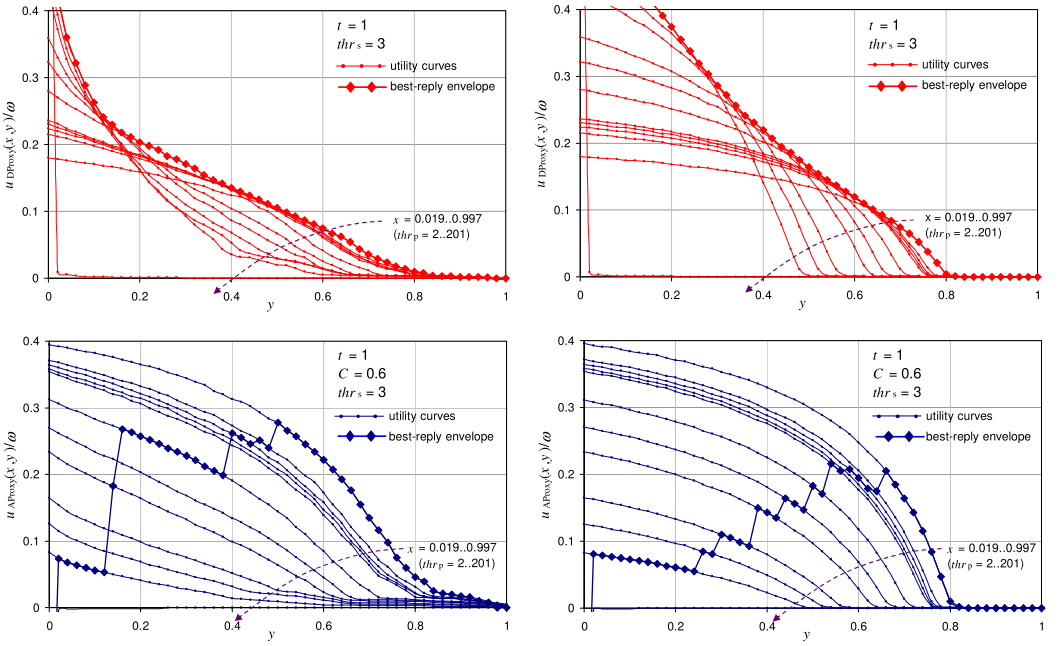


Fig. 9. DProxy's (*top*) and AProxy's (*bottom*) utilities and best-reply envelopes for dataset-based (*left*) and memoryless (*right*) service request streams; 95% confidence intervals based on 100 independent simulation runs are within ±0.005 of the depicted values.

The curves marked with diamonds in the top and bottom plots are best-reply envelopes $u_{\text{DProxy}}(\hat{x}(y), y)$ and $u_{\text{AProxy}}(\hat{x}(y), y)$, respectively. They coincide with segments of utility curves with parameter x for y such that $x = \hat{x}(y)$; $u_{\text{AProxy}}(\hat{x}(y), y)$ owes its "ragged" look to only finitely many x values estimated from the dataset. Similarly to Fig. 7, the plots confirm that AProxy should seek an SE at $\hat{y} \in \text{argmax}_{y\in[0,1]} u_{\text{AProxy}}(\hat{x}(y), y)$, as formalized in Section 4. The memoryless stream produces higher utilities for given (x, y), however, it lowers AProxy's best-reply envelope: shorter *DDoS* spells weaken the drift away from the trust state and allow DProxy to demand $d = \text{H}$ more aggressively, i.e., $\hat{x}(y)|_{\text{memoryless}} > \hat{x}(y)|_{\text{dataset}}$ for all $y \in [0, 1]$. Hence, the memoryless stream model lower bounds AProxy's SE utility.

6. Conclusion and Future Work

FVA distinguishes itself from general QoS abuse in that DProxy consciously exploits AProxy's occasional trust. In the Stackelberg FVA game, a Stackelberg equilibrium (SE) is mediated by the inspection cost C. Under scant transparency, the per service-request game lacks perfect information; this precludes characterizations derived from stage equilibria for information-limitations or learning (e.g., Bayesian) models. Since only long-term utilities are observable, all the optimizations pertain to FVA and trust strategies rather than per service-request decisions. Our model abstracts from Server's internal mechanisms and performance indicators. For two distinguished QoS levels, we show that double-blind reputation improves AProxy's SE utility against a probabilistic FVA strategy. Specifically, we find that:

- The agents' SE utilities are expressible through the synthetic characteristics x and y of their FVA and trust strategies.
- If $C > \omega$, request inspection is pointless without trust.

- DProxy's best-reply x to AProxy's y is unique and decreases in y, while her utility is nonincreasing in y. This induces AProxy to select the largest y maximizing her utility and leads to a unique SE.
- At the SE, AProxy's y is nonincreasing in C and her utility is not lower than that yielded by the trivial trust strategy, with a maximum at $C = \omega$; DProxy's best-reply x and utility are nondecreasing in C.
- At the SE, the reputation scheme is beneficial to AProxy for all C, especially around ω. It can also yield a win–win, at which FVA episodes and request inspection decisions are relatively infrequent.
- Counterintuitively, by controlling signature prediction errors, DProxy weakens her Stackelberg follower position against AProxy, whose SE trust strategy then becomes Grim and disincentivizes FVA.
- By adding trust states, AProxy can increase her SE utility without incentivizing FVA more. However, she must keep current reputation state secret lest her best option become the trivial trust strategy.
- The SE can be fairly accurately gauged through multistage play. However, if DProxy acquires information on C, a precalculated false reply can significantly improve her SE utility and worsen AProxy's.
- Sample experimental evidence suggests that the above conclusions carry over to real-world internally-correlated service request streams.

Known environment-specific defenses against QoS abuse either assume mutual observability of DProxy's and AProxy's behavior, or disregard its strategic aspects. Our approach yields an analytical framework for generic defense solutions in a scant-transparency setting, provided that C is known.

The effectiveness of smarter FVA strategies (e.g., using machine learning) is an open issue. Also, multiple QoS levels and internally correlated service request streams need to be studied systematically. Finally, interference between multiple Client–Server pairs or temporary resource shortages at Server may affect the agents' utilities as the system scales up; these phenomena deserve future research in the context of FVA.

CRediT authorship contribution statement

Jerzy Konorski: Conceptualization, Methodology, Investigation, Writing - original draft, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

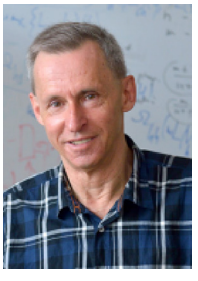
Acknowledgment

This work was supported by the National Science Center, Poland, under Grant UMO-2016/21/B/ST6/03146.

References

[1] N. Agrawal, S. Tapaswi, A proactive defense method for the stealthy edos attacks in a cloud environment, International Journal of Network Management 30 (2) (2020) e2094.
[2] F. Al-Haidari, K. Salah, M. Sqalli, S.M. Buhari, Performance modeling and analysis of the EDoS-shield mitigation, Arabian Journal for Science and Engineering 42 (2) (2017) 793–804.
[3] M. Al-khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, Y. Jararweh, Improving fog computing performance via Fog-2-Fog collaboration, Future Generation Computer Systems 100 (2019) 266–280.
[4] T. Alpcan, T. Basar, Network Security: A Decision and Game-Theoretic Approach, Cambridge University Press, 2010.
[5] P. Bartlett, Optimal online prediction in adversarial environments, in: M. Hutter, F. Stephan, V. Vovk, T. Zeugmann (Eds.), Algorithmic Learning Theory, volume 6331 of Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2010.
[6] N. Basilico, A. Celli, G.D. Nittis, N. Gatti, Coordinating multiple defensive resources in patrolling games with alarm systems, in: Proc. 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'17), 2017.
[7] K. Bhushan, B.B. Gupta, Network flow analysis for detection and mitigation of Fraudulent Resource Consumption (FRC) attacks in multimedia cloud computing, Multimedia Tools and Applications 78 (2) (2019) 4267–4298.
[8] Canadian Institute for Cybersecurity. DDoS evaluation dataset, 2019. URL www.unb.ca/cic/datasets/ddos-2019.html (accessed 2022-01-16)..
[9] I.-R. Chen, F. Bao, J. Guo, Trust-based service management for Social Internet of Things systems, IEEE Transactions on Dependable and Secure Computing 13 (6) (2016) 684–696.
[10] M.H. Cheung, A.H. Mohsenian-Rad, V.W.S. Wong, R. Schober, Random access protocols for WLANs based on mechanism design, in: Proc. IEEE International Conference on Communications, 2009.
[11] F.Z. Chowdhury, M.Y.I. Idris, L.M. Kiah, M.A.M. Ahsan, EDoS Eye: A game theoretic approach to mitigate economic denial of sustainability attack in cloud computing, in: Proc. 8th Control and System Graduate Research Colloquium, 2017.
[12] J.B. Dennis, M.S. Priya, A profile-based novel framework for detecting edos attacks in the cloud environment, Wireless Personal Communications 117 (4) (2021) 3487–3503.
[13] P.T. Dinh, M. Park, Economic Denial of Sustainability (EDoS) detection using GANs in SDN-based cloud, in: Proc. IEEE 8th International Conference on Communications and Electronics, 2021.
[14] Y. Duan, G. Fu, N. Zhou, X. Sun, N.C. Narendra, B. Hu, Everything as a Service (XaaS) on the cloud: Origins, current and future trends, in: Proc. 8th IEEE International Conference on Cloud Computing, 2015.
[15] K. Dutta, A. Datta, D. VanderMeer, H. Thomas, K. Ramamritham, ReDAL: An efficient and practical request distribution technique for application server clusters, IEEE Transactions on Parallel and Distributed Systems 18 (11) (2007) 1516–1528.
[16] Endace Ltd. Troubleshooting quality of service (QoS) issues, 2021. URL endace.com/solutions/network-performance-monitoring/ network-quality-of-service-troubleshooting (accessed 2022-01-16)..

[17] Y. Freund, M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R.E. Schapire, Efficient algorithms for learning to play repeated games against computationally bounded adversaries, in: Proc. 36th Annual Symp. Foundations of Computer Science, 1995.
[18] M.J. Gallivan, Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies, Information Systems Journal 11 (2001) 277–304.
[19] J. Idziorek, M. Tannian, and D. Jacobson. Attribution of Fraudulent Resource Consumption in the cloud. In Proc. IEEE International Conference on Cloud Computing, 2012..
[20] A. Jakóbik, F. Palmieri, J. Kołodziej, Stackelberg games for modeling defense scenarios against cloud security threats, Journal of Network and Computer Applications 110 (2018) 99–107.
[21] K. Jensen. Control is good, but trust is cheaper. Forbes, Dec. 2014. URL www.forbes.com/sites/keldjensen/2014/12/08/control-is-good- trust-is-cheaper/?sh=4d1bd3571322 (accessed 2022–01-16)..
[22] F. Jia, K. Zhou, C. Kamhoua, Y. Vorobeychik, Blocking adversarial influence in social networks, in: Proc. International Conference on Decision and Game Theory for Security, 2020.
[23] C. Jiang, X. Cheng, H. Gao, X. Zhou, J. Wan, Toward computation offloading in edge computing: A survey, IEEE Access 7 (1) (2019) 131543–131558.
[24] W. Jiang, Z. Ma, X. Deng, An attack-defense game based reliability analysis approach for wireless sensor networks, International Journal of Distributed Sensor Networks 15 (4) (April 2019).
[25] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decision Support Systems 43 (2) (2007) 618–644.
[26] M. Karami, S. Chen, Attribution of Economic Denial of Sustainability attacks in public clouds, in: R. Deng, J. Weng, K. Ren, V. Yegneswaran (Eds.), Security and Privacy in Communication Networks, Springer Cham, 2017, pp. 373–391.
[27] J. Konorski, Fake VIP attacks and their mitigation via double-blind reputation, in: Proc. International Telecommunication Networks and Applications Conference (ITNAC), 2017.
[28] J. Konorski, Double-blind reputation vs. intelligent Fake VIP attacks in cloud-assisted interactions, Proc. 17th IEEE TrustCom/12th IEEE BigDataSE (2018).
[29] M.N. Kumar, P. Sujatha, V. Kalva, R. Nagori, A.K. Katukojwala, M. Kumar, Mitigating economic denial of sustainability (EDoS) in cloud computing using in-cloud scrubber service, in: Proc. 4th International Conference on Computational Intelligence and Communication Networks (CICN), 2012.
[30] E.B. Lakew, C. Klein, F. Hernandez-Rodriguez, E. Elmroth, Performance-based service differentiation in clouds, in: Proc. 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2015.
[31] P.K. Malik, D.S. Wadhwa, J.S. Khinda, A survey of device to device and cooperative communication for the future cellular networks, International Journal of Wireless Information Networks 27 (2020) 411–432.
[32] M. Masood, Z. Anwar, S.A. Raza, M.A. Hur, EDoS Armor: A cost effective economic denial of sustainability attack mitigation framework for e-commerce applications in cloud environments, in: Proc. 16th International Multi-Topic Conference (INMIC), 2013.
[33] M.A.S. Monge, J.M. Vidal, G.M. Pérez, Detection of economic denial of sustainability (EDoS) threats in self-organizing networks, Computer Communications 145 (2019) 284–308.
[34] T.H. Nguyen, A. Butler, and H. Xu. Tackling imitative attacker deception in repeated Bayesian Stackelberg security games. In Proc. European Conference on Artificial Intelligence, 2020..
[35] E. Parzen, Stochastic Processes, Holden-Day, San Fransisco, 1962.
[36] S. Rabie and O. Aboul-Magd. A Differentiated Service Two-Rate, Three-Color Marker with efficient handling of in-profile traffic. RFC 4115, 2005..
[37] E. Rasmusen, Games and Information: An Introduction to Game Theory, Wiley-Blackwell (2006).
[38] J. Riordan, Stochastic Service Systems, Wiley, New York, 1962.
[39] T. Szigeti, C. Hattingh, R. Barton, K. Briley Jr., End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks, Cisco Press (2013).
[40] S. Szott, J. Konorski, Traffic remapping attacks in ad hoc networks, IEEE Communications Magazine 56 (4) (2018) 218–224.
[41] M. Tambe, Security and Game Theory: Algorithms, Deployed Systems, Cambridge University Press, Lessons Learned, 2011.
[42] A. Verma, P. Verma, S.K. Dhurandher, I. Woungang, Opportunistic Networks: Fundamentals, Applications and Emerging Trends, CRC Press, 2021.
[43] R. Wang, M. Li, L. Peng, Y. Hu, M.M. Hassan, A. Alelaiwi, Cognitive multi-agent empowering mobile edge computing for resource caching and collaboration, Future Generation Computer Systems 102 (2020) 66–74.
[44] K. Xiao, C. Zhu, J. Xie, Y. Zhou, X. Zhu, W. Zhang, Dynamic defense against stealth malware propagation in cyber-physical systems: A game-theoretical framework, Entropy 22 (8) (2020) 894.
[45] S. Yu, Y. Tian, S. Guo, D. Wu, Can we beat DDoS attacks in clouds?, IEEE Transactions on Parallel and Distributed Systems 25 (9) (2014) 2245–2254
[46] S. Zeadally, J. Guerrero, J. Contreras, A tutorial survey on vehicle-to-vehicle communications, Telecommunication Systems 73 (3) (2020) 469–489.
[47] Y. Zhang, P. Malacaria, Bayesian Stackelberg games for cyber-security decision support, Decision Support Systems 148 (2021) 113599.



Jerzy Konorski received his M. Sc. degree in telecommunications from the Gdansk University of Technology, Poland, and his Ph. D. degree in computer science from the Polish Academy of Sciences. He is currently with the Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, where he conducts research and teaching. He has authored 170 papers on computer communications, led scientific projects funded by the EU, US Air Force, and National Science Center, Poland, and served on the TPC for 120 international conferences. His current work focuses on game theory and reputation systems in wireless networks and security architectures.