Probabilistic model checking of CAPTCHA admission control for DoS resistant anti-SPIT protection

Emmanouela Stachtiari[1], Yannis Soupionis[2], Panagiotis Katsaros[1], Anakreontas Mentis[1], and Dimitris Gritzalis[2]

[1] Dependability & Security Group Dept. of Informatics, Aristotle Un. of Thessaloniki, Greece
{emmastac, katsaros, anakreon}@csd.auth.gr
[2] Information Security And Critical Infrastructure Protection Research Group, Dept. of Informatics, Athens Un. of Economics and Business, Athens, Greece
{jsoup, dgrit}@aueb.gr

Abstract. Voice over IP (VoIP) service is expected to play a key role to new ways of communication. It takes advantage of Internet Protocols by using packet networks to transmit voice and multimedia data, thus providing extreme cost savings. On the other hand, this technology has inherited drawbacks, like SPAM over Internet Telephony (SPIT). A well-established method to tackle SPIT is the use of CAPTCHAs. CAPTCHAs are vulnerable to Denial of Service (DoS) attacks, due to their excessive demands for bandwidth. We suggest that anti-SPIT protection should be combined with appropriate admission control policies, for mitigating the effects of DoS attacks. In order to identify how effective is this technique, we quantify the costs and the benefits in bandwidth usage through probabilistic model checking four different admission control policies. We conclude with comments on how appropriate is each policy in tackling DoS attacks.

Keywords: admission control, DoS, CAPTCHA, probabilistic model checking

1 Introduction

Voice over IP (VoIP) offers a low-cost and high-quality service of multimedia data transmission over Internet Protocol networks, such as the Internet. Inevitably though, VoIP has "inherited" a major threat of internet communication service abuse that is widely known as SPAM over Internet Telephony SPIT [15]. Key VoIP service providers have recognized SPIT as a critical issue that undermines IP telephony growth and they develop increasingly sophisticated mechanisms to tackle this threat [19, 13].

Most of the systems that provide protection against SPIT adopt basic principles and design considerations from the area of IT Security [11, 6, 12]. A widespread

approach is to develop policy-based mechanisms based on predetermined conditions. These conditions can be formalized as a structured security policy implemented by a set of rules defined by the user organization. Anti-SPIT protection policies include all decisions and actions to perform, in order to provide effective protection. However, it is true that under certain circumstances such a policy cannot identify whether a VoIP communication originates from software robots ("bots") or humans. A common technique to manage those VoIP sessions is the use of audio CAPTCHAs (Completely Automated Public Turing Test to Tell Computer and Humans Apart) [18].

A CAPTCHA is a test that most humans should be able to pass, but computer programs should not. Even though audio CAPTCHA is a well-established technology, as an anti-SPIT protection mechanism it is susceptible to Distributed Denial of Service (DDoS) attacks, due to the computer resources demands associated with the CAPTCHA challenges. DDoS attacks against VoIP servers are notoriously difficult to be countered [16]. They are launched by taking control of numerous machines that belong to unaware users. These machines are used to force a victim server into instantly delivering a huge bulk of CAPTCHA challenges that can exhaust the available bandwidth [9]. Such a bandwidth consumption attack may render the server unavailable for legitimate users.

We propose that anti-SPIT protection should be combined with appropriate admission control policies for mitigating the effects of DDoS attacks. In a VoIP server under attack the vast majority of service requests causing CAPTCHA challenges comes from incoming attack traffic. Admission control filters all service requests according to some bandwidth preservation criterion and therefore opens a possibility to prevent legitimate CAPTCHA challenges. In effect, a cost for bandwidth preservation is induced.

We quantify the costs and the benefits in bandwidth usage through probabilistic model checking of a server model handling DDoS traffic. We opted for a Continuous Time Markov Chain (CTMC) representation developed within the PRISM [7] model checking toolset and parameterized based on data from real audio CAPTCHA implementations [12]. The obtained results provide analytic estimates of appropriate cost and benefit metrics for comparing cost-effectiveness of four admission control policies, namely simple sum [8], cutoff scheme [10], fractional guard [14] and a threshold-based [3] policy. Our cost and benefit metrics are expressed as reward properties quantified over all possible paths of the CTMC reachability graph. Our comparative results have been validated by extensive sensitivity analysis over different combinations of parameter ranges.

The paper is organized as follows. First, in Section 2 we give a short background on probabilistic model checking and the PRISM tool. Section 3 discusses the examined admission control policies and the attack scenario we consider. In Section 4, we provide the details of the performed probabilistic analysis and the developed PRISM models, along with the adopted cost and benefits metrics. Section 5 presents the obtained model checking results and interprets the shown trends. Section 6 refers to the related work for probabilistic analysis of bandwidth abuse. The paper concludes with a review on the findings and a discussion

on their contribution towards the improvement of anti-SPIT protection and our future work plans.

2 Probabilistic model checking

Probabilistic model checking is an algorithmic formal verification technique for analyzing systems with probabilistic behavior. The PRISM model checking toolset [7] generates models from a high-level description of the system's behavior with guarded commands. These commands are grouped into modules and each module's behavior is defined by local variables and transitions of the form:

$$[l]g \rightarrow \phi_l : u_1 + ... + \phi_n : u_n; \quad (1)$$

The guard g is a predicate over model variables and constants, whereas each update u_i specifies how new values are assigned to local variables. Transitions can occur with some likelihood or frequency depending on the specified model type. We opted a CTMC for faithfully representing the contention for a limited amount of bandwidth provided in a VoIP system. For CTMCs, ϕ_i is the transition's rate, the parameter of a negative exponential distribution that governs the waiting time of the transition, provided that the guard is true. Optional label l is used for explicit synchronization with other transitions. The rate of synchronized transitions is the product of the rates of the individual transitions.

Our CTMC model possesses the fundamental Markov property, which is also presumed in [4] for analyzing bandwidth usage: the conditional probability distribution of future states depends only upon the present state. This assumption is justified by the fact that bandwidth sharing, under the common assumption of Poisson session arrivals, is insensitive to the flow size and the session arrival process.

PRISM implements a series of graph-theoretic algorithms for reachability analysis and iterative numerical solvers for computing/checking probabilistic properties expressed in Probabilistic Computation Tree Logic (PCTL). It is also possible to compute reward properties based on some reward structure defined over the model. Path formulas in PCTL consist of temporal operators, such as F (eventually) or G (always) and predicates that can be true or false in the reached states. Models can be queried with properties of the form $P =?[prop]$ about the probability to satisfy a path formula *prop*. Reward structures are used to accumulate reward or cost values when certain states or transitions are observed. Properties of the form $Rr =?[prop]$ query the model for the expected value of the reward r along paths satisfying *prop*. In CTMCs, probabilistic and reward-based properties can be bounded by time t, if a transient analysis is required. In this case, a reward property takes the form $(C \leq t)$.

3 Admission control policies

Let us consider the following scenario: a client group of legitimate users is served by the VoIP infrastructure, while at the same time new legitimate clients and

intruders that generate attack traffic are requested to answer CAPTCHAs. Requests for multimedia services and CAPTCHA resources demand a non-negligible amount of bandwidth and they can be granted process time, as long as the bandwidth has not been exhausted. In a DDoS attack, the number of malicious requests is rapidly increased in order to cause bandwidth exhaustion. The attack succeeds, if the incoming malicious requests consume all the available bandwidth, before a legitimate request for service or a CAPTCHA resource can be accepted.

When a policy is applied for admission control, the available bandwidth is preserved by distinguishing the CAPTCHA challenges that will be served. This is a sufficient way to prevent bandwidth abuse, since the bandwidth for the provided VoIP services can be utilized only by legitimate users (we assume that attackers cannot break/pass the CAPTCHA test and therefore they cannot dispatch directly requests for multimedia services).

We focus on parameter-based admission control policies, as opposed to measure-based policies that cannot be analyzed by a model-based approach. More precisely, we study the following policies: the simple sum [8], the cutoff scheme [10], the fractional guard [14] and a threshold-based [3] policy. In order to discuss the policy approach, we have adopted the following notation:

1. κ representing the consumed bandwidth at a certain instant
2. β referring to the total link capacity
3. ρ denoting the expected bandwidth for a service request, and
4. c the expected bandwidth consumed for transferring a CAPTCHA challenge

In all policies, a service request is accepted provided that there is available bandwidth to serve it ($\kappa + \rho < \beta$). The applied control approach for admission of the arrived CAPTCHA requests is as follows:

Simple Sum: a CAPTCHA challenge is delivered only if $\kappa + c < \beta$. This means that a new request is accepted, only if the available bandwidth suffices for the expected CAPTCHA challenge needs in bandwidth.

Cutoff scheme: requests by already authorized users have a higher priority than the incoming CAPTCHA requests. In effect, a portion of the bandwidth is reserved for assuring that user needs will be always accommodated. The question regarding the ratio of bandwidth that should be reserved poses a significant trade-off. If too much bandwidth is reserved, then new users requesting access to services are likely to be discarded even when the bandwidth is underutilized. On the other hand, underestimation of the bandwidth needed for providing VoIP services can lead to the rejection of VoIP service requests. A CAPTCHA request is accepted if $max(\kappa, \delta)+c < \beta$, where δ is the bandwidth reserved for authorized users.

Fractional guard: CAPTCHA requests are accepted with a probability that depends on the bandwidth consumption. This policy can be expressed by the equation $\pi * rand(0, 1) > \kappa$, where $rand$ returns a uniform positive value less than one and π is a parameter that influences the acceptance rate.

Threshold based: this policy is an adapted version of the call bounding scheme, described in [3]. During a fixed period of time τ at most λ CAPTCHA

requests are accepted. All other requests are rejected until the expiration of the time period, when the number of received requests is set to zero.

4 VoIP system model and cost benefit analysis

4.1 PRISM model description

Four models have been developed in PRISM, with each of them representing a VoIP server under DDoS attack with one of the mentioned admission control policies. The applied policy affects the model's behavior regarding whether an arriving CAPTCHA request can be accepted or not. CAPTCHA requests are generated from one legitimate and one malicious source with different rates, thus representing a race for acquiring the still available bandwidth resource. We consider that if a legitimate CAPTCHA request is admitted, the model's execution ends with an attack failure. If a legitimate CAPTCHA request cannot be accepted due to bandwidth exhaustion, then the malicious traffic results in a DDoS success. Since the period of time needed to solve a CAPTCHA challenge (5-15 seconds) exceeds the time period in which an attack success is observed (<1 second), it is reasonable to take into account the additional bandwidth consumption by all accepted CAPTCHA requests, until reaching the model's final state. The following model variables are used to encode state information for the described VoIP system.

- *captchas*, the number of CAPTCHA challenges currently in progress
- *established_legitimate*, becomes true if a legitimate CAPTCHA request has been accepted
- *successful_attack*, becomes true if the malicious traffic has consumed all available bandwidth
- *captcha_counter*, a counter used in the threshold-based policy for the number of accepted CAPTCHA requests. It is periodically set to zero.

Table 1 highlights the model parameters, with some of them being common in all models. We vary the *malicious_requests* parameter to represent DDoS attacks of various intensities. All other parameters are assigned constant values. The *clients_rate* and *new_clients_rate* parameters are assigned values, such that on average one CAPTCHA request is expected among 20 service requests from legitimate users.This happens due to the fact that most anti-SPIT techniques have the CAPTCHA mechanism as the last obstacle against SPIT attacks [5] and consequently only a portion of SPIT attacks are challenged by CAPTCHA. The *captcha_size* parameter represents bandwidth consumption for transferring the CAPTCHA challenge's audio file and is set to 200 kbps [18]. We assume that the *service_size* is 83 kbps of Session Initiation Protocol (SIP) trunk bandwidth per service request [1], considering the G.711 wave-format codec. We set the *link_capacity* to 5 Mps for a VoIP server with a total bandwidth of 10 Mps, out of which 5 Mps are devoted for servicing already authorized clients.

The decision of whether a request will be accepted or not is made based on two PRISM formulas, namely AdmitService and AdmitCAPTCHA. While

Name	Description
All models	
malicious_requests	Rate of malicious requests. (1-100.000 req/sec)
clients_rate	Rate of requests arriving from authorized clients. (200 req/sec)
new_clients_rate	Rate of requests arriving from unauthorized clients. (10 req/sec)
captcha_size	Bandwidth requirements for a CAPTCHA challenge. (200 kbps)
service_size	Bandwidth requirements for serving authorized clients. (83 kbps)
link_capacity	Total available bandwidth (5 Mbps)
Cutoff scheme	
bandwidth_reserved	Percentage of available bandwidth reserved for serving service requests
Fractional guard	
acceptance_parameter	A value such that the probability of accepting a CAPTCHA is given by $acceptance\_parameter * k$, where k is the currently available bandwidth
Threshold-based	
initialize_rate	Frequency of setting the counter of accepted CAPTCHA requests to zero.
captcha_limit	Threshold of the counter accepted CAPTCHA so that CAPTCHA requests are still accepted.

Table 1. Model Parameters

the AdmitService formula has the form $consumed\_bandwidth + service\_size \leq link\_capacity$ in all models, the AdmitCAPTCHA formula varies for each policy, as it is shown in Table 2.

Policy name	AdmitCAPTCHA variants
simple sum	$consumed\_bandwidth + captcha\_size \leq link\_capacity$
cutoff scheme	$consumed\_bandwidth + captcha\_size \leq link\_capacity$
fractional guard	$consumed\_bandwidth + bandwidth\_reserved * link\_capacity + captcha\_size \leq link\_capacity$
threshold-based	$(captcha\_counter < captcha_limit) \& (consumed\_bandwidth + captcha\_size \leq link\_capacity)$

Table 2. The variants of the AdmitCAPTCHA formula in each policy

The main transitions of the VoIP system model are:

- *admit_malicious*: admit a malicious CAPTCHA request. Occurs with rate *malicious_requests*, if the AdmitCAPTCHA formula evaluates to true.
- *reject_malicious*: drop a malicious CAPTCHA request. Occurs with rate *malicious_requests*, if the AdmitCAPTCHA formula evaluates to false.

- *admit_new_client*: admit a CAPTCHA request from legitimate user. Occurs with rate *new_clients_rate*, if the AdmitCAPTCHA formula evaluates to true.
- *reject_new_client*: drop a CAPTCHA request from legitimate user. Occurs with rate *new_clients_rate*, if the AdmitCAPTCHA formula evaluates to false.
- *admit_service*: admit a service request. Occurs with rate *service_rate*, if the AdmitService formula evaluates to true.
- *reject_service*: drop a service request. Occurs with rate *new_clients_rate*, if the AdmitService formula evaluates to false.

In the fractional guard policy, the typical rate for an accept request transition is multiplied by $accept\_parameter*(link\_capacity-consumed\_bandwidth)/link\_capacity$, where *accept_parameter* is the value of π. Similarly, the typical rate for a drop request transition is multiplied by $1 - (accept\_parameter * (link\_capacity - consumed\_bandwidth)/link\_capacity)$

4.2 Costs and benefit of admission control

The probability of DDoS attack success against a CAPTCHA anti-SPIT mechanism is only one aspect of admission control effectiveness. Evaluating other aspects of bandwidth usage is also very important. For example, an extremely rigid policy that discards most CAPTCHA requests is not vulnerable to DDoS, but it also fails to serve the legitimate users who generate CAPTCHA requests.

A more complete view for the admission control cost-effectiveness is obtained by considering all metrics related to costs and benefits in bandwidth usage, while avoiding to quantify strongly correlated properties such that we will not take into account the same effects twice. Then, it will be possible to compare the different policies based on their net benefit.

We assigned costs and benefits to specific model events by attaching the following reward structures:

- Accepted (A_S) and rejected (R_S) service requests: they compute respectively the expected number of accepted and rejected service requests. Both of them take values in the range of [0, 1], since the server model receives at most one service request at each execution path.
- Accepted (A_N) and rejected (R_N) new clients: they compute the expected number of accepted and rejected CAPTCHA requests from new legitimate clients. They also take values in the range of [0, 1] for the same reason as the previous structures.
- Available bandwidth (A_B) while the system is under DDoS attack: this reward computes the expected bandwidth percentage that remains available during the attack.

We defined the cost and benefit metrics from the server's viewpoint as shown in Table 3. The probability to accept an incoming service request should be as high as possible and therefore we consider it as a benefit metric. On the other

hand, the probability to reject a request initiated by a new client, as well as the percentage of unexploited bandwidth upon a DDoS attack is a cost that admission control should minimize. The net benefit is calculated by

$$netbenefit = 2 * B_1 - C_1 - C_2 \tag{2}$$

The benefit is weighted twice as much as each of the costs. The appropriate weights depend on the optimization priorities that are specific to the context of the performed analysis.

Cost		Benefit	
Metric	Value	Metric	Value
Probability of rejecting an incoming request from a new client (C_1)	$R_N/(A_N + R_N)$	Probability of accepting an incoming service request (B_1)	$A_S/(A_S + R_S)$
Percentage of unexploited bandwidth upon DDoS (C_2)	A_B		

Table 3. Cost and benefit metrics for the analysis

The goal of our analysis is to rank the policies according to their net benefit, at each grade of attack intensity. Since the choice of parameters affects the ranking, we assumed the best net benefit for each policy with some specific attack intensity. By having ran the models with different parameters, we discovered the higher net benefit for increasing numbers of malicious requests.

5 Experimental results

The experimental results for a wide range of malicious requests rate using various model parameters are summarized in Fig. 1. In simple sum, the net benefit decreases exponentially as the malicious requests become more frequent. Specifically, the benefit converges to zero and the probability to reject a new client's request increases at higher attack rates. On the other hand, the available bandwidth is zero for all rates of malicious requests. Cost outperforms benefit when the attack rates are higher than 13.000 requests per second.

The cutoff scheme offers greater net benefit as the reserved bandwidth shrinks, as long as it is more than the percentage of bandwidth occupied by service requests, which we assume to be 0.5. Reserving 60% of bandwidth, as compared to 80%, leads to less unexploited bandwidth and to a higher probability of accepting new clients' requests. On the other hand, accepting an incoming service request is guaranteed in both cases. The maximum net benefit was achieved by setting the reserved bandwidth parameter to 0,51.

In fractional guard, the higher the accept parameter value is, the higher the net benefit for up to 10.000 malicious requests is. At larger attack rates, a

smaller accept parameter is optimal and it achieves better results after 14.000 malicious requests. When the attack rate is low, a small accept parameter leads to a significantly higher possibility of dropping new client requests and a larger percentage of unexploited bandwidth. On the other hand, fractional guard offers a higher probability of serving a service request, which is maintained for very high attack rates. Generally, the net benefit decreases slower for small accept parameters.

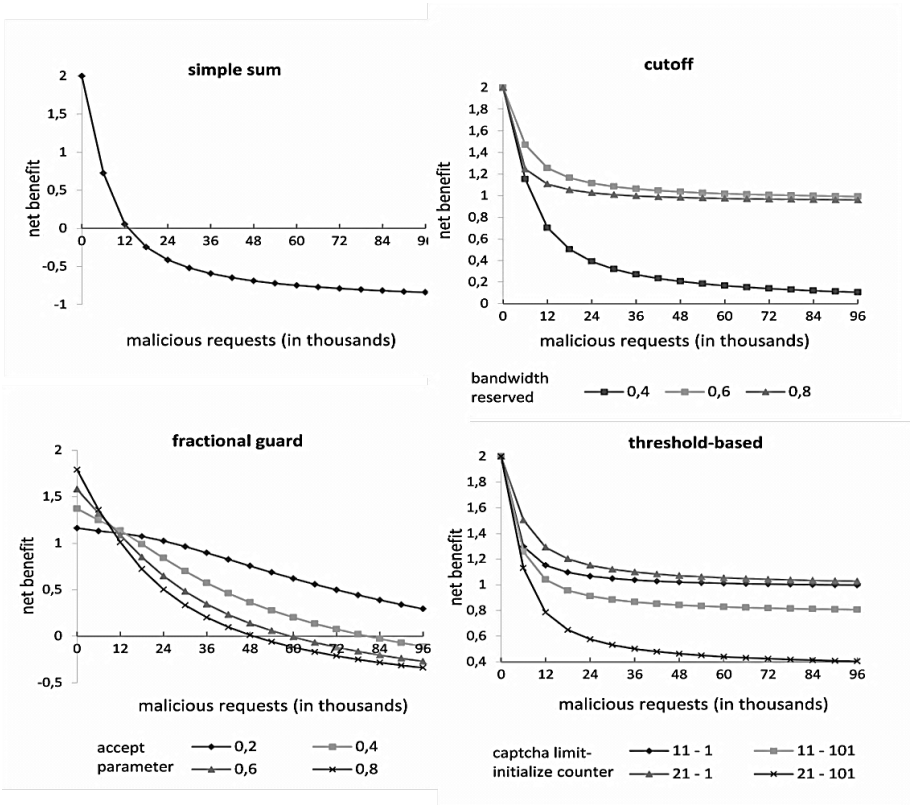

Fig. 1. The net benefit for different parameters of the examined policies

The series of net benefit in the threshold-based policy are associated with sets of the CAPTCHA limit and initialization rate parameters. When the initialization rate equals to 1, setting the CAPTCHA limit to 21 yields better net benefit. A requirement for the CAPTCHA limit is to be smaller than the maximum CAPTCHA sessions that the bandwidth can handle. For an initialization rate of 101, bigger CAPTCHA limit leads to better net benefit. This is observed because the more frequent the initialization of the accepted CAPTCHA counter

happens, the smaller the counter threshold is. The best net benefit was achieved by setting the initialization rate to 1 and the CAPTCHA limit to 24, which is the maximum number of CAPTCHA that the bandwidth can accommodate, while at the same time it has enough space to serve a service request.

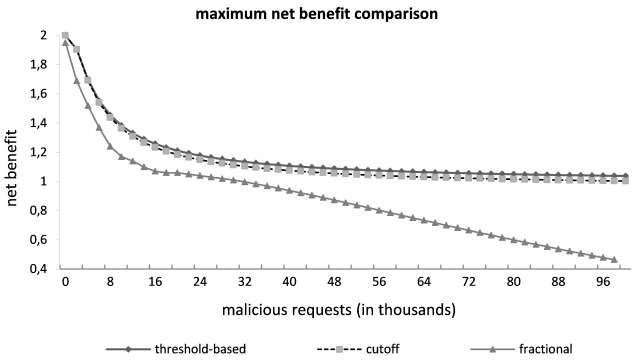

Fig. 2. Maximum net benefit achieved by policies

It is concluded that the threshold-based policy gave the best results in the cost-benefit analysis at all attack rates as Fig. 2 displays. On the other hand, the cutoff policy ranks second, having a very small difference from the first one. This policy requires knowledge of the expected arrival rate of legitimate requests, which is susceptible to change. Fractional guard did not perform very well due to our choice of evaluation metrics. This policy does not provide enough guarantees for serving legitimate requests and randomized decision of accepting CAPTCHA requests can have flaws that are revealed by model checking (all possible results of these decisions are taken into account).

6 Related work

A game-theoretic framework for analyzing the effectiveness of bandwidth attacks is presented in [17]. The game consists of an attacker launching an attack using IP spoofing and a defender, who attempts to detect the attack using coarse-grained statistical filtering. The authors examine various strategies for such games and evaluate the payoffs. Their results show that statistical methods are a promising means for revealing bandwidth abuse attacks. They also pose the idea to use statistical filtering for identifying suspicious groups in the server's traffic, towards directing resource-consuming filtering techniques only to those groups.

In [8], the authors define the evaluation criteria to compare one parameter-based and tree measure-based admission control policies. Parameter-based poli-

cies make decisions based on a priori knowledge of the server's traffic, while measure-based policies rely on actual measurements of load. They evaluate simulation results based on the criteria of the probability for guaranteed service to clients and the level of network utilization. However, model checking provides higher confidence for policy effectiveness than simulation-based approaches, because the results characterize all possible system execution scenarios, as opposed to only a limited number of simulation traces.

PRISM is used in [2] to formally analyze the bandwidth amplification attack against the Domain Name System (DNS), along with three countermeasures: packet filtering, random packet drops, and aggressive retries of legitimate packets. The authors compute the attack probability based on CTMC representation of an attack scenario and measure the cost-effectiveness of each countermeasure, when it is used for completely eliminating the probability of attack success. Our cost-benefit analysis is based on the same principles with those applied in [2].

7 Conclusions and further research

Voice over IP technology becomes a popular communication system, as it is widely used for establishing and maintaining multimedia sessions over the Internet. One of the obvious potential problems of VoIP applications is the growth of the SPIT phenomenon, which is often handled with the use of audio CAPTCHA. Even though this technique is able to distinguish sessions which are initiated by human from those initiated by software (bots), it adds up a serious threat: the possibility of bandwidth abuse through DDoS attacks.

In this work we study the effectiveness of CAPTCHA admission control policies towards eliminating the DDoS threat by discarding a limited number of VoIP sessions. We provide results obtained by probabilistic model checking of a CTMC representing bandwidth consumption in a VoIP system. Costs and benefits for the analyzed policies have been quantified by appropriate reward properties defined and evaluated within the PRISM model checking toolset.

The examined policies are ranked according to the computed net benefit for a range of parameter values that provide insight for the analysis sensitivity. We show that threshold-based and cutoff admission policies offer effective defense against DDoS attacks. Two other policies, namely fractional guard and simple sum are not characterized by decent results in terms of cost-effectiveness. More specifically, it is shown that when applying the simple sum policy, the more attack sessions are initiated the less net benefit is obtained. We note that the parameters used in our analysis do not depend on hardware characteristics.

As future work prospect we aim to study how the admission policies and the use of audio CAPTCHA affect the time needed for call establishment in VoIP sessions, since there are critical timeouts set by most VoIP providers. Moreover, we plan to measure cost effectiveness not only in terms of bandwidth usage, but also in terms of CPU and memory consumption.

References

1. Cisco: Voice over ip per call bandwidth consumption. Document id 7934, Cisco Communication (February 2006)
2. Deshpande, T., Katsaros, P., Basagiannis, S., Smolka, S.A.: Formal analysis of the DNS bandwidth amplification attack and its countermeasures using probabilistic model checking. In: HASE, IEEE Computer Society (2011) 360–367
3. Fang, Y., Zhang, Y.: Call admission control schemes and performance analysis in wireless mobile networks. Vehicular Technology, IEEE Transactions on **51**(2) (March 2002) 371–382
4. Fred, S.B., Bonald, T., Proutiére, A., Régnié, G., Roberts, J.W.: Statistical bandwidth sharing: a study of congestion at flow level. In: SIGCOMM. (2001) 111–122
5. Gritzalis, D., Marias, G.F., Rebahi, Y., Soupionis, Y., Ehlert, S.: Spider: A platform for managing sip-based spam over internet telephony (spit). Journal of Computer Security **19**(5) (2011) 835–867
6. Gritzalis, S., Gritzalis, D.: A digital seal solution for deploying trust on commercial transactions. Inf. Manag. Comput. Security **9**(2) (2001) 71–79
7. Hinton, A., Kwiatkowska, M., Norman, G., Parker, D.: PRISM: A tool for automatic verification of probabilistic systems. In Hermanns, H., Palsberg, J., eds.: TACAS. Volume 3920 of LNCS., Springer (2006) 441–444
8. Jamin, S., Shenker, S., Danzig, P.B.: Comparison of measurement-based call admission control algorithms for controlled-load service. In: INFOCOM. (1997) 973–980
9. Kandula, S., Katabi, D., Jacob, M., Berger, A.: Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds. In: NSDI, USENIX (2005)
10. Lin, Y.B., Mohan, S., Noerpel, A.: Queueing priority channel assignment strategies for PCS hand-off and initial access. Vehicular Technology, IEEE Transactions on **43**(3) (August 1994) 704–712
11. Marias, G., Dritsas, S., Theoharidou, M., Mallios, J., Gritzalis, D.: Sip vulnerabilities and anti-spit mechanisms assessment. In: ICCCN. (2007) 597–604
12. Mitrou, L., Gritzalis, D., Katsikas, S.K., Quirchmayr, G.: Electronic voting: Constitutional and legal requirements, and their technical implications. In Gritzalis, D., ed.: Secure Electronic Voting. Volume 7 of Advances in Information Security. Springer (2003) 43–60
13. Quittek, J., Niccolini, S., Tartarelli, S., Stiemerling, M., Brunner, M., Ewald, T.: Detecting spit calls by checking human communication patterns. In: ICC, IEEE (2007) 1979–1984
14. Ramjee, R., Towsley, D., Nagarajan, R.: On optimal call admission control in cellular networks. Wireless Networks **3** (1997) 2941
15. Rosenberg, J., J.C.: The session initiation protocol (sip) and spam. Rfc 5039, Network Working Group (January 2008)
16. Sisalem, D., Kuthan, J., Ehlert, S.: Denial of service attacks targeting a sip voip infrastructure: attack scenarios and prevention mechanisms. IEEE Network **20**(5) (2006) 26–31
17. Snyder, M.E., Sundaram, R., Thakur, M.: A game-theoretic framework for bandwidth attacks and statistical defenses. In: LCN, IEEE Computer Society (2007) 556–566
18. Soupionis, Y., Gritzalis, D.: Audio captcha: Existing solutions assessment and a new implementation for voip telephony. Computers & Security **29**(5) (2010) 603–618
19. Soupionis, Y., Gritzalis, D.: Aspf: Adaptive anti-spit policy-based framework. In: ARES. (2011) 153–160