Malicious Users Discrimination in Organized Attacks Using Structured Sparsity

Mehmet YAMAÇ
Electrical and Electronics Engineering
Boğaziçi University
Bebek 34342, Istanbul, Turkey
Email: mehmet.yamac@boun.edu.tr

Bülent Sankur
Electrical and Electronics Engineering
Boğaziçi University
Bebek 34342, Istanbul, Turkey
Email: bulent.sankur@boun.edu.tr

Ali Taylan Cemgil
Dept. of Computer Engineering
Boğaziçi University
Bebek 34342, Istanbul, Turkey
Email:taylan.cemgil@boun.edu.tr

***Abstract*—Communication networks can be the targets of organized and distributed attacks such as flooding-type DDOS attack in which malicious users aim to cripple a network server or a network domain. For the attack to have a major effect on the network, malicious users must act in a coordinated and time correlated manner. For instance, the members of the flooding attack increase their message transmission rates rapidly but also synchronously. Even though detection and prevention of the flooding attacks are well studied at network and transport layers, the emergence and wide deployment of new systems such as VoIP (Voice over IP) have turned flooding attacks at the session layer into a new defense challenge. In this study a structured sparsity based group anomaly detection system is proposed that not only can detect synchronized attacks, but also identify the malicious groups from normal users by jointly estimating their members, structure, starting and end points. Although we mainly focus on security on SIP (Session Initiation Protocol) servers/proxies which are widely used for signaling in VoIP systems, the proposed scheme can be easily adapted for any type of communication network system at any layer.**

***Index Terms*—Compressive Sensing, Network Security, DDOS, Voice over IP**

I. Introduction

Voice over IP (VoIP) is a technology that allows making voice calls using a broadband Internet connection; its rising popularity indicates that it will completely take over both public switched telephone networks (PSTN) and cellular networks, for example, in 5G systems. The Internet telephony protocol, namely the Session Initiation Protocol (SIP) not only provides more flexible and low-cost communication, but also simplifies data sharing, multimedia conmmunication, web conferences etc. In the application layer, most VoIP systems use Session Initiation Protocol [18] (SIP) to setup communication calls and Real-Time Transport Protocol (RTP) to deliver media. Despite the key advantages of internet telephony, the downside is that VoIP systems are still vulnerable to a wide range of attacks, the most common one being Denial-of-Servise/Distributed-Denial-of-Service (DOS/DDOS) flooding attacks [9].

DOS/DDOS flooding attacks can be done at both transport/network layer and application layer levels. The flooding attacks at transport/network layers and their countermeasures have been well documented in the literature. However, flooding attacks in the application layer targeting certain services, and in particular aiming to consume network server resources to make the system unresponsive to valid user requests is a new trend. One type of SIP flooding is just sending a large number of fake SIP packets to exhaust processor bandwidth. Another type floods the proxy server with INVITE or REGISTER messages [19], [20]. An INVITE message is used to establish a communication session among two or more users, and when such a message is received in the proxy server, the session initiation state is kept for up to 3 minutes [18]. Therefore an INVITE flooding can exhaust the resources of the server. Similarly, attackers can flood REGISTER messages to cause a DOS in the SIP registrar.

For SIP based VoIP applications, there are a number of flooding attack detection methods in the literature. Reynolds et al. [10] have developed a detection scheme based on the well-known cumulative sum method [11]. In [5], the authors propose a statistical anomaly detection system based on the Hellinger distance. In [6], a hybrid anomaly detection scheme is suggested to detect flooding attacks used both at SIP and RTP layers. In [8] the authors suggest and compare three algorithms: adaptive threshold, cumulative sum and Hellinger distance. The surveys [9], [12] contain a more detailed review of literature on SIP flooding attacks. One shortcoming of the existing detection methods is that the behaviour of individual network users is not directly observed, so that it is not easy to distinguish malicious users. In this work, we focus on SIP packet flooding attacks and we propose a sparsity-based anomaly detection scheme with the goal to model the behavior of individual terminals and to identify malicious users. Another advantage of our scheme is that it does not require any training phase or setting of any thresholds.

In Section II-A, we discuss organized attacks typical of DDOS and the way they can be related to structured sparsity-based signal representation. Then, we give the sparse signal representation of organized attacks in II-B. In Section III, the DOS/DDOS detection problem is defined as a constrained optimization problem. In section IV, we discuss proximal methods suitable for the convex optimization problem given in the problem definition. Finally, we show the performance of the proposed algorithm in the Experimental part and draw conclusions.

ISBN 978-0-9928626-7-1

II. Preliminaries

A. Assumptions for Organized Attacks

We assume the following characteristics of an organized attack such as DDOS: (i) Attacks occur very rarely compared to normal packet traffic, implying that anomalies appear sparsely in the time index. (ii) In any attack event, only a small group from the entire set of users is expected to act maliciously, which means that anomalies are also sparse in the user index. (iii) The contributions of individual attackers for a particular attack are assumed to be sychronized. (iv) No one attacker can dominate the change in the overall statistics. (v) Normal background SIP packet traffic is also sparse in both time and user index. Session initiation packets are expected to be rare as compared to the transport protocol data units, such as RTP traffic which carries the media streams.

B. Sparse Representation of the organized attack problem

Let Y be the $m \times N$ measurement matrix whose ith row, y_i includes the vector of message counts of the ith user observed over N instances. The measurements can represent the count of total SIP messages or the count of individual types of SIP message, such as INVITE. All measurements correspond to count of events within fixed time quanta, e.g., 1 second. We assume that unless there is an attack, the user state remains the same in the network. Then the measurement vector, y_i can be approximated by a piece-wise constant function u_i representing current state of the user i and b_i representing the normal background traffic, i.e.,

$$Y = U + B, \tag{1}$$

In this equation, U is the state matrix whose i'th row consists of the piece-wise constant function u_i and B is the background measurement matrix whose i'th row is the b_i. In Figure 1, an example $m \times N$ measurement matrix is simulated. The anomalies (or change in the states) occur only in the data of four groups of 40 users from the total of 100, each group consists of 10 users. In this scenario, the anomalous increases in traffic intensity take place concurrently, which may imply a synchronized attack.

In the case of d-dimensional feature vectors, for example the counts of the d-types of individual SIP messages, the measurements will constitute a $d \times m \times N$ measurement tensor Y.

III. Problem definition

In the model presented in II-B, the problem can formulated as estimating the state matrix U given the observation matrix Y in view of the assumptions listed in II-A. The estimation of U can be expressed mathematically as follows:

$$\hat{U} = \arg\min_{U} \left\{ F(U) \equiv \sum_{i=1}^{4} w_i f_i(U) \right\} \tag{2}$$

where each of the terms, $f_i(U)$ corresponds to one of the constraints in Section II-A, and the w_i are their associated weights.

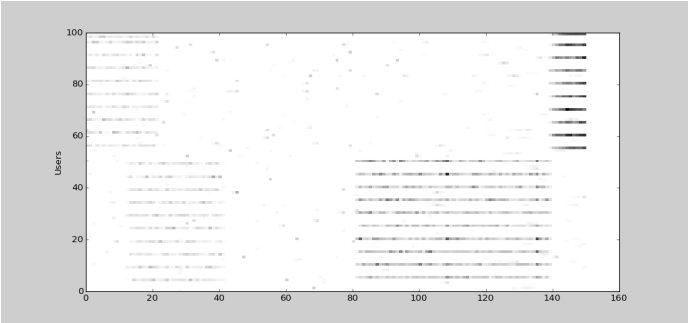

Fig. 1: An example $m \times N$ measurements matrix where $N = 150$ current observations of $m = 100$ users. The measurements consist of the number of received SIP messages per second. In this figure four different attack instances from different groups and at different times (0-20, 15-40, 80-140, 140-180) can be observed against the normal background traffic.

A. Data Fidelity

In the cost function $F(U)$, fidelity of the model U to the data Y can be set as the mean squared error $f_1(U)$ as $w_1 \|Y - U\|_F^2$.

B. Sparsity of anomalies in time

We assume that anomalies are sparse in the time index. Such a penalty function, $f_2(U)$, can be expressed as $w_2 \sum_{i=1}^{N-1} \sum_{k=1}^{m} |U_{k,i+1} - U_{k,i}|$. This is indeed the total variation in one direction and minimizing this term forces the functions, $u_i, i = 1, ..., N$, to be piece-wise constant [16], [17].

C. Sparsity of attackers

We assume that only a minority from the entire user set would organize themselves to mount an attack. This assumption can be satisfied by enforcing sparseness of the number of rows of U, i.e., by choosing $f_2(U)$ as $\|U\|_{0,2} = \sum_{k=1}^{m} I\left(\|U_{k,:}\|_2\right)$, where $I(.)$ is the indicator function. This term is non-convex and can be relaxed to $\|U\|_{1,2} = \sum_{k=1}^{m} \|U_{k,:}\|_2$ [2].

D. Synchronization of attacks in a group

An organized attack DDOS attack can be realized if the flooding attempt from malicious users are mostly correlated and synchronized in time. This assumption can be satisfied by choosing the nuclear norm of U, i.e., $f_4(U)$ as $\sum_{k=1}^{\min\{m,n\}} I(\sigma_k)$, where σ_k is kth singular value of U and $I(.)$ is the indicator function. Since this term is not convex it can be relaxed to $\|U\|_* = \sum \sigma_i(U)$ [4].

IV. Proximal Splitting Methods

The formulation of the cost function given in Eq. (2) involves the sum of both smooth and non-smooth terms, for which the solution method is non-trivial. Among alternate algorithms in literature for the solution of such convex optimization problems, we chose the proximal method to solve Eq. (2) for the following reasons:

ISBN 978-0-9928626-7-1 © EURASIP 2017

- The proximal method is suitable for distributed optimization. In the case of a large sized problem with thousands of users and large dimensional feature vectors ($m \gg 1$ and $d \gg 1$, respectively in the $d \times m \times N$ measurement tensor), this method can be very convenient.
- Proximal methods work for both smooth and non-smooth extended real valued functions. Then, any modification in problem definition given in Equation (2) can be easily accommodated in our system.

A. Proximal Mapping

Proximal methods use proximal operators to solve convex optimization problems. A proximal operator or proximal mapping [1] of a function f_i with weight γ can be defined as

$$\text{prox}_{\gamma f}(\mathbf{z}) = \arg\min_{\mathbf{u}} \{ f(\mathbf{u}) + \frac{1}{2\gamma} \|\mathbf{u} - \mathbf{z}\|^2 \}$$

B. Parallel proximal algorithm

Following parallel proximal algorithm given in [3], [21], which is derived from Dougles-Rachford algorithm [22] can be used to solve in distributed manner convex optimization problems that incorporate more than one non-smooth function as in (2).

Algorithm 1 Parallel proximal algorithm

Input: Y
Determine: λ, $\{w_i\}_{i=1}^4$ s.t. $\sum w_i = 1$, γ
Initialize: $U, Z, p_i = Y, i = 1, ...4, k = 0$
repeat
  $p_i \leftarrow prox_{\gamma f_1}(z_i)$, $i = 1, ..., 4$
  $P \leftarrow \sum_{i=1}^4 w_i p_i$
  $z_i \leftarrow z_i + \lambda (2P - U - p_i)$, $i = 1, ..., 4$
  $U \leftarrow U + \lambda (P - U)$
  $k \leftarrow k + 1$
  $\gamma \leftarrow g(\gamma)$
until Convergence or maximum iteration ($k == maxit$)
return U

In this algorithm Y is our $m \times N$ measurement matrix for $d = 1$, and γ is the step size. We have four weighting parameters such that $\sum_i w_i = 1$ whose selection will be discussed later. Although the step size γ can also be chosen as a constant, we opt for a monotonically decreasing function $g = \frac{n*maxit-(n-1)*k}{maxit}$ for faster convergence. The proximal operators of each term are as follows:

1) Proximal Operator of Data Fidelity Term: Using the definition, one has:

$$prox_{\gamma f_1}(Z) = \arg\min_U \frac{1}{2} \|U - Z\|_F^2 + w_1 \|Y - U\|_F^2$$

where Z is the $m \times N$ current solution. The proximal solution of this term is simply the gradient of f_1,

$$prox_{\gamma f_1}(Z) = (z + 2\gamma Y)/(1 + 2\gamma).$$

2) Proximal Operator of the term enforcing the sparsity in the user index:

$$prox_{\gamma f_3}(Z) = \arg\min \frac{1}{2} \|U - Z\|_F^2 + \gamma \|U\|_{\ell_{1,2}},$$

which yields

$$(prox_{\gamma f_3}(Z))(j,:) = \begin{cases} \left(1 - \frac{\gamma}{\|Z(j,:)\|}\right) Z(j,:) & \text{if } \gamma < \|Z(j,:)\|, \\ 0 & \text{else} \end{cases}$$

3) Proximal Operator of the term enforcing synchronization of group attackers:

$$prox_{\gamma f_4}(Z) = \arg\min \frac{1}{2} \|U - Z\|_F^2 + \gamma \|U\|_*,$$

which yields

$$prox_{\gamma f_4}(Z) = A * prox_{\gamma \ell_1}(diag(D)) * B^T,$$

where $Z = ADB^T$ is singular value decomposition of Z and $prox_{\gamma \ell_1}(x)$ is given as

$$prox_{\ell_1}(x_i) = \begin{cases} x_i + \gamma & \text{if } x_i \leq -\gamma \\ 0 & \text{if } -\gamma \leq x_i \leq +\gamma, \\ x_i - \gamma & \text{if } x_i \geq 0 \end{cases}$$

when x_i is the i.th element of vector x.

4) Proximal Operator of the term enforcing the sparsity of anomalies in time: As stated in Section III-B, this term can be achieved using proximal operator of TV function in one dimension. However, there is no closed form of the proximal map of TV regularizers. In [23], authors use a special case of parallel proximal algorithm, Proximal Dykstra to compute proximal of 2-D TV using proximal operator of 1-D in distributed manner.

V. Experimental Design

A. Network Traffic Generator

Network Traffic Generator (NTG) is a tool [7] to generate calls among users registered in a SIP server. NTG aims to mimic genuine user behaviors in term of call frequency, call duration and call direction by initiating sessions, holding and ending them. Notice that NTG is not concerned with the generation of voice packets as in Real-time Transport (RTP) traffic, but generates only SIP message traffic.

NTG is based on a probabilistic generative model and the behavior of the model changes according to the predefined parameters such as number of users, number of social groups, average call duration etc. The probabilistic generative model can be thought as a library that gives us the probability distribution of reactions of individual users for every possible scenario. Details and default parameters are given in [7]. NTG is built using Python and open source SIP call generator library PJSIP [14]. A free of charge Asterisk PBX based SIP server, Trixbox [15] is used in our experiments.

ISBN 978-0-9928626-7-1 © EURASIP 2017

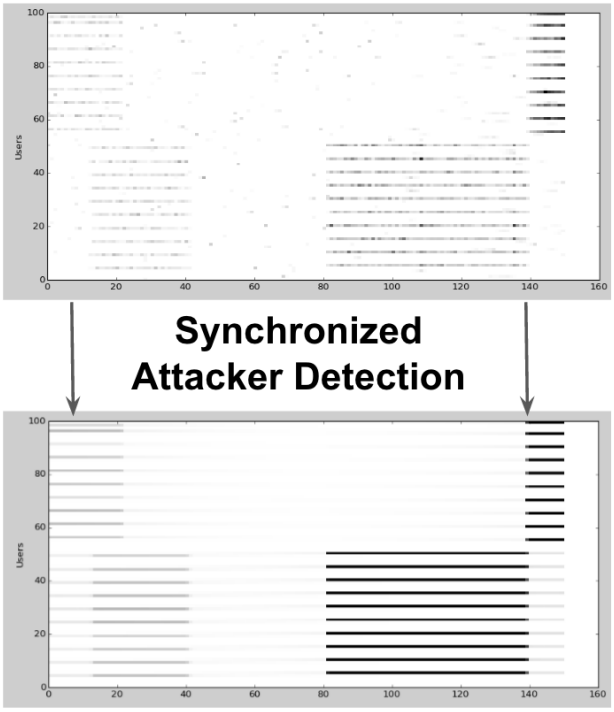

Fig. 2: (i) Monitor visualizes the number of packets received from each users. (ii) Estimations of state, $\hat{U}$. This is used in discrimination of the attacker groups.

B. Attack Generator

The organized attacks are produced using the NETAS NOVA V-SPY tool [13]. In this tool DDOS attacks can be produced by selected malicious users with given attack intensity values. Attack intensity is adjusted vis-á-vis the background normal traffic intensity.

C. Features

Most commonly used features for the detection of flooding type attacks comprise the traffic intensity at servers/proxies. In this study, incoming and outgoing packet traffic at the SIP server is monitored. Our monitoring system captures the counts of the incoming and outgoing packets per second and distinguishes 14 different types of request and 14 type of responses. These packet types list as follows:

- Requests : Register, Invite, Subscribe, Notify, Options, Ack, Bye, Cancel, Prack, Publish, Info, Refer, Message, Update
- Responses : 100, 180, 183, 200, 400, 401, 403, 404, 405, 481, 486, 487, 500, 603

D. Experiments

1) Performance measurements: The performance of the system is measured in terms of precision, recall and F-measure, detection latency, correct detection of the start (onset) and end points (offset) of the attacks.

$$\text{Precision (P)} = \frac{\text{\# assigned true (attack/group members)}}{\text{\# assigned attackers}} \tag{3}$$

$$\text{Recall (R)} = \frac{\text{\# assigned true (attack/group members)}}{\text{\# all attackers}} \tag{4}$$

$$\text{F-Measure (F)} = 2\frac{P \times R}{P + R} \tag{5}$$

For each suspicious group, the time elapsed until all the group members are detected is also recorded. To this effect, We define the delay performance criteria as follows;

$$\text{D} = \frac{\text{The elapsed time until all group members detected}}{\text{Total number of the suspicious groups}} \tag{6}$$

Start/end point detection success rate is recorded as

$$\text{SPD/EPD} = \frac{\text{\# correctly assigned start/end points}}{\text{total number of attacks}} \tag{7}$$

2) Experiment 1: In this experiment, a VoIP network consisting of 100 users is simulated using the network simulator described in Section V-A, while the group attacks are simulated using VSPY. In each group, 10 of the total users are selected as malicious and each such group attacks the system at different times with different attack intensity values. We use a total number of 28 incoming and outgoing message types, and their occurrences are totaled over 1 second intervals. The experiment runs for most recent 150 seconds. Therefore we have measurements collected in a 100×150 matrix Y_t at time t, including information of recent 150 seconds in the streaming sense. We also record the malicious users' identity in each attacker group, and the start and end points of the attacks. Each attacker group is identified with its start point. The elapsed time until all members of an attacker group are detected is also automatically recorded by the monitoring system. We generate flooding attacks with five types of messages: REGISTER, INVITE, OPTIONS, CANCEL and BYE packets. The attack intensities are selected as 50, 100, 250 and 750 (e.g., if we choose INVITE type attack with intensity of 50 for 10 attackers, VSPY floods INVITE messages with a mean of 5 packets per user in each second). Average performance results from 80 attacks with mixed message types are given in Table I. As expected, the detection delay decreases as the attack intensity increases.

	Attack intensity	P	R	F	SPD	EPD	Delay
Group members	50	1	1	1	N/A	N/A	13
	100	1	1	1	N/A	N/A	6.11
	250	1	1	1	N/A	N/A	5.08
	750	1	1	1	N/A	N/A	2
Organized Attack	50	1	1	1	1	0.75	N/A
	100	1	1	1	1	0.75	N/A
	250	1	1	1	1	83	N/A
	750	1	1	1	1	91	N/A

TABLE I: The results for Experiment 1.

ISBN 978-0-9928626-7-1 © EURASIP 2017

3) Experiment 2: This time we fixed the attack generator to a very low attack intensity of 50. In this case, the expected message count in a second per attacker becomes 2 if we assume a population of 25 malicious users. The performance of the system as the number of malicious users change is estimated by generating 80 separate attacks. The total number of active users at the VoIP server is selected as 100, and the attack durations range from 30 seconds to 120 seconds. Table II shows that the performance of the proposed method is still satisfactory even for low intensity attacks. We have observed that most of the missed attacks occur when the attack intensity (per user) is very low and it is of short duration. To quote some figures, failures to detect occur when the expected total packet count per user (when 25 malicious users are considered) is 4 (2 for request and 2 for response) and the attack duration is less then 30 seconds. One way to improve the proposed method is to consider additional features from the server resource usage such as CPU and memory usage.

	Attacker p.	P	R	F	SPD	EPD	Delay
Group members	5	1	1	1	N/A	N/A	5.12
	10	1	1	1	N/A	N/A	12.37
	15	0.995	0.987	0.99	N/A	N/A	24.62
	20	0.94	0.763	0.84	N/A	N/A	37.46
	25	1	0.64	0.78	N/A	N/A	66
Organized Attack	5	1	1	1	1	0.75	N/A
	10	1	1	1	1	0.75	N/A
	15	0.952	1	0.975	1	0.81	N/A
	20	0.952	0.95	0.951	0.95	0.90	N/A
	25	1	0.75	0.87	0.75	0.62	N/A

TABLE II: Results for Experiment 2

4) The weighting parameters: A few words about the parameters are in order. The tuning of the four weighting parameters in the optimization problem forms trade-off between precision and recall. For instance, if we decrease the weight of the term enforcing user index sparsity, it will cause detection delay to decrease and recall rate to increase, however at the cost of an increase in false alarm rate. Likewise, an emphasis on the nucleus norm will improve the detection precision of attacking groups, but it will deteriorate recall rate of both groups and group members since the remaining 3 weights must decrease. After experimentation with our simulation environment, we have determined the parameters, $w_1, .., w_4$ to be $0.037, 0.25, 0.25, 0.463$, respectively.

VI. Conclusion

In this study, a synchronized flooding attack detection scheme has been proposed for SIP networks and its effectiveness demonstrated with simulation experiments. It has been observed that sparse models are very effective in flooding attack detection given the sparse nature of the SIP messaging traffic. The proposed scheme is also highly effective in the identification of attackers, individually and as a group. One advantage of the algorithm is that it does not need any parameter settings such as threshold values. Furthermore, the solution of the optimization problem via proximal operators opens the way to distributed computing, which can be crucial in large-scale problems.

Acknowledgment

This study is a Bogazici University - NETAS collaboration and it is funded with TEYDEB project number 3140701, "Realization of Anomaly Detection and Prevention with Learning System Architectures, Quality Improvement, High Rate Service Availability and Rich Services in a VoIP Firewall Product", by the Scientific and Technological Research Council Of Turkey (TUBITAK). NOVA V-Gate is a trademark cybersecurity product of NETAS.

References

[1] Parikh, Neal, and Stephen P. Boyd. "Proximal Algorithms." Foundations and Trends in optimization 1.3 (2014): 127-239.
[2] Adler, Amir, et al. "Sparse coding with anomaly detection." Journal of Signal Processing Systems 79.2 (2015): 179-188.
[3] Combettes, Patrick L., and Jean-Christophe Pesquet. "Proximal splitting methods in signal processing." Fixed-point algorithms for inverse problems in science and engineering. Springer New York, 2011. 185-212.
[4] Cai, Jian-Feng, Emmanuel J. Cands, and Zuowei Shen. "A singular value thresholding algorithm for matrix completion." SIAM Journal on Optimization 20.4 (2010): 1956-1982.
[5] Sengar, Hemant, et al. "Detecting VoIP floods using the Hellinger distance." IEEE transactions on parallel and distributed systems 19.6 (2008): 794-805. APA
[6] Sengar, Hemant, et al. "Fast detection of denial-of-service attacks on IP telephony." 200614th IEEE International Workshop on Quality of Service. IEEE, 2006.
[7] Ceritli, Taha Yusuf, et al. "A probabilistic SIP network simulation system." Signal Processing and Communication Application Conference (SIU), 2016 24th. IEEE, 2016.
[8] Akbar, M. Ali, Zeeshan Tariq, and Muddassar Farooq. "A comparative study of anomaly detection algorithms for detection of SIP flooding in IMS." Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on. IEEE, 2008.
[9] Keromytis, Angelos D. "A comprehensive survey of voice over IP security research." IEEE Communications Surveys & Tutorials 14.2 (2012): 514-537.
[10] Reynolds, Brennen, and Dipak Ghosal. "Secure IP Telephony using Multi-layered Protection." NDSS. 2003.
[11] Wang, Haining, Danlu Zhang, and Kang G. Shin. "Change-point monitoring for the detection of DoS attacks." IEEE Transactions on Dependable and Secure Computing 1.4 (2004): 193-208.
[12] Ehlert, Sven, Dimitris Geneiatakis, and Thomas Magedanz. "Survey of network security systems to counter SIP-based denial-of-service attacks." Computers & Security 29.2 (2010): 225-243.
[13] http://novacybersecurity.com/en/nova-vspy.html
[14] http://www.pjsip.org/
[15] http://www.fonality.comltrixbox
[16] Rudin, Leonid I., Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms." Physica D: Nonlinear Phenomena 60.1 (1992): 259-268.
[17] Levy-leduc, Cline, and Zad Harchaoui. "Catching change-points with lasso." Advances in Neural Information Processing Systems. 2008.
[18] Rosenberg, Jonathan, et al. SIP: session initiation protocol. No. RFC 3261. 2002.
[19] Geneiatakis, Dimitris, Costas Lambrinoudakis, and Georgios Kambourakis. "An ontology-based policy for deploying secure SIP-based VoIP services." computers & security 27.7 (2008): 285-297.
[20] Geneiatakis, Dimitris, et al. "Survey of security vulnerabilities in session initiation protocol." IEEE Communications Surveys and Tutorials 8.1-4 (2006): 68-81.
[21] Combettes, P.L., Pesquet, J.C.: A proximal decomposition method for solving convex variational inverse problems. Inverse Problems 24, 27 (2008). Art. 065014
[22] Douglas, Jim, and Henry H. Rachford. "On the numerical solution of heat conduction problems in two and three space variables." Transactions of the American mathematical Society 82.2 (1956): 421-439.
[23] Barbero, Alvaro, and Suvrit Sra. "Modular proximal optimization for multidimensional total-variation regularization." arXiv preprint arXiv:1411.0589 (2014).