DE GRUYTER OPEN

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **15**, No 2

Sofia • 2015

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2015-0033

Trilateral Trust Based Defense Mechanism against DDoS Attacks in Cloud Computing Environment

N. Ch. S. N. Iyengar[1], *Gopinath Ganapathy*[2]

[1] *School of Computing Science & Engineering, VIT University, Vellore-632014, Tamilnadu, India*
[2] *Director, Technology Park Bharathidasan University, Trichy-620023, India*
Emails: nchsniyengar48@gmail.com gganathy@gmail.com

Abstract: *Distributed Denial of Service* (*DDoS*) *in a Cloud leads to a high rate of overload conditions, which subverts the Data Center* (*DC*) *performance and ends up in resource unavailability. This work proposes a "Trilateral Trust mechanism" which helps in detecting different kinds of attack groups at different points of time. It is the direct trust based defense mechanism for segregating legitimate and attack groups from the vast number of incoming requestors. It is a hybrid mechanism of trusts that follows the zero trust approach initially and eventually supports both Mutual trust and Momentary trust. This combinatorial trust mechanism helps in detecting almost all kinds of overload conditions at a cautionary period. Detecting the high rate of an attack at an earlier moment of time could reduce the traffic impact towards DC. The simulation results and profit analysis proved that the mechanism proposed is deployable at an attack-prone DC for resource protection, which would eventually benefit the DC economically as well.*

Keywords: *Cloud computing, DDoS, Data center, Cloud service provider, Momentary trust, Flash crowd, Mutual trust.*

1. Introduction

Cloud computing is a technology that suffered from its security breaches, of which availability is the most serious security issue. Distributed Denial of Service (DDoS) is a kind of resource-availability related attack for subverting the Data Center (DC) for resource unavailability to the legitimate clients. So, in order to avoid the DDoS

attack traffic, the network should only be allowed legitimate traffic, which needs some trust with the incoming requestor not only through authentication, but also by momentary scrutinizing that will also help in monitoring the precise traffic behavior.

Unlike conventional DDoS detection and protection mechanisms, the Trilateral Trust mechanism is a scheme which is extending in direction of detection with three sidelong functionalities. They are the preliminary traffic signaller, Authentic Trust launcher, historical trust Analyzer that scrutinizes several cases of requestors and continuously monitors their behavior and updates the trusteeship of the requestor towards the Cloud Service Provider (CSP). When this optimized Trilateral Trust mechanism is deployed at the DC, the improved efficiency can be achieved as shown through simulation.

DDoS is one of the most high traffic rate overload conditions towards DC. The incoming huge traffic consists of both legitimates and attacking traffic. Segregating them dynamically is a challenging task. Moreover, this DDoS attack is easier to launch from the client end and harder to withstand or detect at DC end. This made us pursue our research towards direction of DDoS Detection and DC resource protection in a cloud network.

Cloud Flare experienced the largest DDoS attack flooding in February 2014, which is a record-breaking from 300 up to 400 Gbps attack. This attack is launched by anti-spammers Spamhaus, which was the largest DDoS attack up to date (2014) [1]. Some other notorious DDoS attack experiences are: Mafiaboy who succeeded in bringing down the world's most popular websites, namely Yahoo, CNN, ebay, Dell and Amazon in February, 2000. Similar attacks were made towards South-Korean's largest newspaper, bank and United States forces created as a botnet over hundred thousands of computers in July, 2009 [2]. Still, several serious DDoS events lead to long outages to be identified and prevented.

2. Related survey of existing techniques

Recommendation-based trust model proposed the transitivity of the trust with certain constraints [3]. If *A* trusts *B*, and *B* trusts *C*, then we could not declare that *A* trusts *C* because the trust is not a transitive scheme. But *A* can trust *C* with a certain hypothetical constraint. Firstly, when *B* explicitly recommends *A* to trust *C*. Secondly, *A* can trust *B*'s recommender. *A* can judge *B*'s recommendation and can decide whether to accept *B*'s recommendation or not.

Trusting the recommender's recommendation could be an uncertainty of risk. The recommendation protocol is simple. It creates a recommendation path.

Suppose that *A* knows that the request can be processed by *D*, but does not have trust in *D*. Hence, it asks for a recommendation from its trustworthy member *B*. Since *A* trusts *B*, but *B* trusts *C*, but does not trust *D*. So it simply forwards *A*'s request to *C*. Eventually, as *C* trusts *D* and can serve the requested service. Finally, now *C* recommends *D* to *B*, which in turn recommends to *A* and serves *A*. This creates the recommended path $A \times B \times C \times D$.

Another Recommendation model [4] uses the "evaluated trust" value to trust any stranger. The trust evaluation involves the degree of satisfaction between two parties along with the balance and considers the number of interactions between them. This model also considers the degree of complaints and degree of interaction history. This model insists that if the constraint must be highly trusted, the number of interactions should always be greater than the preset minimum number of interactions. This scheme also suffers from a couple of assumptions, such that the higher trust value user is always more reliable than the lower trust value user which always affects the newly joined peers, the minimum number of interactions should be more than the threshold in order to have a better trust which again affects the newly joined peers.

Sporas, which is a "reputation" based methodology has the reputation value of any user from 0 up to 3000 [5]. Obviously, the new user will have the minimal reputation value of 0. The current user's reputation will always be higher than the new users. Any two users should have only one rating value range from 0 up to 3000. In case the two users have several interactions, then only the recent value is considered. The evaluation is based on the more recent computed reputation value, because the more recent computed values results are current or much closer to the current behavior. The reputation value increases over the period upon good behavior and does not influence the initial low score.

Eschenauer proposed a framework for the evidence based trust management [6]. This considers the trust as a set of relationships between any two parties with the support of evidence. One way to generate evidence is through public-key cryptography. One of the entities in the network can create evidence for itself and for others. In order to create the evidence, the creator entity creates a piece of entity and signs it with the private key. It mentions its validity period and shares it to others with a public key for identification. Here the drawback is that an entity could also revoke the shared evidence. Since the revoking option could allow any anonymous partner to create evidence and to revoke it, this would create chaos in the network.

Multilevel trust filtration [7] mechanism consists of four modular detection algorithms. Firstly, Link pre-fetch which attempts to identify the location of the incoming requestor. Secondly, the Requestor Scrutinizer which verifies the network-specific data to authenticate the incoming requestor. Thirdly, Traffic data log which logs the request rate and request type and distinguishes the attack category, and eventual access after the right approval decides the differential treatment for any different type of overload conditions and incoming traffic.

All the above techniques utilize the trust management scheme to defend against DDoS and all of them are probabilistic in nature. They achieve either mutual trust or momentary trust with the traffic overhead. However, the proposed "trilateral trust" mechanism supports mutual trust at the authentication phase as identity initiation and momentary trust at the behavior monitoring phase. In addition to it, the traffic overhead is reduced and learned much earlier at the edge of CSP network and acts accordingly.

3. Overview of the trilateral trust mechanism

3.1. Architecture of a trilateral trust mechanism

Whenever the clients require a service, they initiate the service request to the subscribed cloud service provider's DC, which is routed to the Traffic Injection Rate Detector (TIRD). At TIRD, the maximum number of requests ($TIRD_{max}$) that the Data Center (DC) can handle is preset based on the cloud service provider configuration.

If the number of requests exceeds $TIRD_{max}$, the traffic condition is considered "abnormal". TIRD redirects the client request to the firewall. The firewall verifies the log that the incoming client is a defaulter for a service provision. If the incoming client is not a defaulter, then the client ID is forwarded to the Mutual Trust Initiator (MTI) which is a database server that holds the clients secret key. For MTI Session, ID is generated and encrypted with the secret key and sent back to the requestor. Now at the requestor end the session ID can be obtained only if the requestor is a valid and legitimate requestor for the secret key is shared only between MTI and the requestor. When the client sends back the session ID to MTI, the session is established. Otherwise, the behavior is considered as resource hunger activity and the credit point is considerably reduced.

Once the session is established, the client encrypted Trust Tag is passed to the Trust Tag Validator (TTV) where the encrypted trust tag is decrypted with a secret key and the behavior history can be monitored. If the incoming client is a defaulter, then a particular client will not be served until the session expires. This scheme assures that the same client who is defaulted is not given a chance of being served continuously. This allows new users to be served. The trust tag reviews the behavior history and reports the character of the requestor. At the Credit Points Updating Module (CrPU), based on the behavior reported by TTV, the CrPU classifies the requestor as a legitimate behavior or resource hunger behavior. The resource hungers' credit points are reduced considerably. If the behavior is legitimate and calm, the credit points are increased. Once the requestor passes the validations at these three different views, then the requestor is considered legitimate.

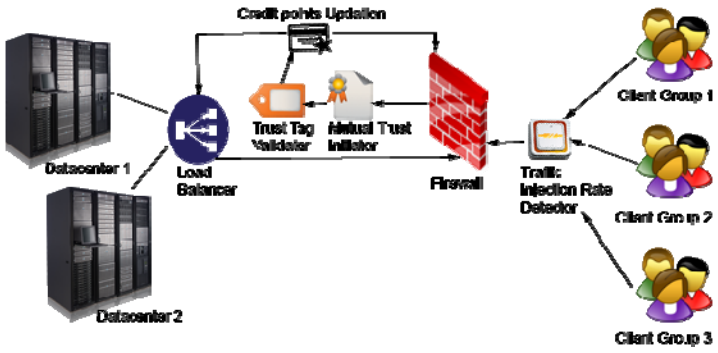

Fig. 1. Architecture of the proposed trilateral trust mechanism

Now the incoming traffic from CrPU to the load balancer is considered legitimate. At the Load Balancer (LB), all the incoming requests are queued and forwarded to the DC based on the DC load, so that the DC is available to all legitimate requestors all the time. It is a necessary requirement that all the new requestors towards DC must be generated a secret key shared between the DC and the requestor via the same cloud service provider channel. Subsequent communication will be authenticated based on the session ID which monitors only active sessions, thus reducing the traffic monitoring overhead. It would be common that the digital certificate would be lost at the traffic prone networks, so that the number of attempts and the timeout period to accept the certificate as valid depends on the cloud service provider network conditions which can also be configured. Higher traffic congestion is proportional to increasing the attempt of accepting the certificate as valid. Here MTI acts as a key manager and also a scalable database server, which also suffices shared secret key validator for any service requestor from the requestor and assures the integrity by validating a digitally signed certificate.

The proposed Trilateral Trust Mechanism (TTM) involves three sequential traffic threat notification levels for authenticating the incoming requestor as a trusted client or a threat. At each level, some kind of a threat is detected and the threat traffic is reduced and narrowed to successive levels. Upon detecting the high rate of attack-prone sources at earliest stages reduces the traffic congestion at DC. This ultimately improves the DC resources available only for legitimates without contaminating other expensive resources for attack-prone traffic threats.

3.2. Protocol of the Trilateral trust mechanism

- **Client ID Acquisition**

Client → TIRD**:** ID_{client}, $ID_{service}$

At TIRD: Computation of $TIRD_{curr}$ at time t.

TIRD → Firewall: Incoming ID_{client} and $TIRD_{curr}$

At Firewall: Comparatively scrutinizes with the log and allows the incoming requestor.

Firewall → MTI: Forwards the ID_{client} which surpasses as legitimate**.**

- **Mutual Trust Establishment**

At MTI: Generate $ID_{session}(X)$ for $ID_{client}(X)$

MTI → Client: $E(K_{DC-X}[ID_{session}(X)])$

At Client: $D(K_{DC-X}, E(K_{DC-X}[ID_{session}(X)]))$

Client→ MTI: $E(K_{DC-X}, [ID_{session}(X)]) \,||\, ID_{client}$

At MTI: Segregate and acquire ID_{client} and fetch K_{DC-X}

$D(K_{DC-X}, E(K_{DC-X}[ID_{session}(X)]))$

Upon successful authentication, $K_{session}$ is generated, ID_{client} forwarded to TTV, otherwise Botnets, spoof attackers are detected and dropped.

- **Historical Behavior Monitoring**

MTI → TTV *:* $ID_{client} \,||\, ID_{session} \,||\, K_{session}$

AtTTV: Acquire ID_{client}

Validate Trust Tag and update $ID_{session}$

Upon successful validation, ID_{client} is forwarded to CrPU, otherwise DDoS, Aggressive legitimates are detected and dropped.

- **Credit points updation**

Credit Points (CrP) are incremented or decremented

If CrP is positive and $TIRD_{curr} > TIRD_{max}$

Flashcrowd

Else

Legitimate

For each incoming requestor

If $CrP\_threshold_{curr} > CrP\_threshold_{safe}$

Session rank

Upon successful classification, ID_{client} is forwarded to LB, based on the Legitimates and the flash crowd event with their rank.

- **Service Provision**

$CrPU \rightarrow LB : ID_{client} \,||\, CrP \,||\, SR$

AtLB: Based on the SR and CrP, the incoming client requests are queued.

$LB \rightarrow DC$*:* Based on the DC load, the requests are redirected to the corresponding DC.

$DC \rightarrow$ client *:* Serve client requests to clients via a firewall.

- **Attacker Exclusion**

AtMTI: upon authentication failure, the deviated clients are reported at the firewall

$MTI \rightarrow$ Firewall: forward ID_{client} (X) via CrPU.

At Firewall: Requests are dropped until a session expiry for ID_{client} (X)

AtTTV: upon successful authentication at a former level and failure at this level, the deviated clients are reported at the firewall

$TTV \rightarrow$ Firewall: forward ID_{client} (X) via CrPU.

At Firewall: Requests are dropped until a session expiry for ID_{client} (X)

At CrPU : upon successful authentication at former levels and traffic condition at this level, the client's load balancing scheme is varied

$CrPU \rightarrow LB$ *:* forward ID_{client} (X).

AtLB: Requests are redirected based on the $TIRD_{curr}$

$LB \rightarrow DC : ID_{client}$ $(X) \,||\, K_{session} \,||\, ID_{service}$

$DC \rightarrow$ Client : Serve client requests to clients via the firewall.

Legends: ID_{client} – Client's ID; $ID_{Service}$ – Service Request ID; $Current_{trc}$ – current Traffic rate; $ID_{Service}(X)$ – Session ID for the client X; K_{DC-X} – secret key between DC and client X; E – Encrypt; D – Decrypt; CrP – Credit Points;

SR – Session Rank; $TIRD_{curr}$ – Current Traffic Rate at TIRD; $TIRD_{max}$ – maximum capacity Traffic Rate at TIRD; $K_{session}$ – Session Key for the client; $CrP\_threshold_{curr}$ – CrP of a client based on the behavior; $CrP\_threshold_{safe}$ – minimum CrP limit to trust a client, as well a trusted client.

4. Working mechanism of the Trilateral trust scheme

This section explains the working principle of the proposed trilateral trust based defense mechanism and the way it protects DC from typical overload threats against DC.

4.1. Detailed description of the TTM

DES is implemented instead of AES to avoid unnecessary overhead of the encryption block length. DDoS is an overload condition, so the detection mechanism should not create additional overhead having an encrypted client request. The sequential view of the proposed TTM is described in Fig. 2.

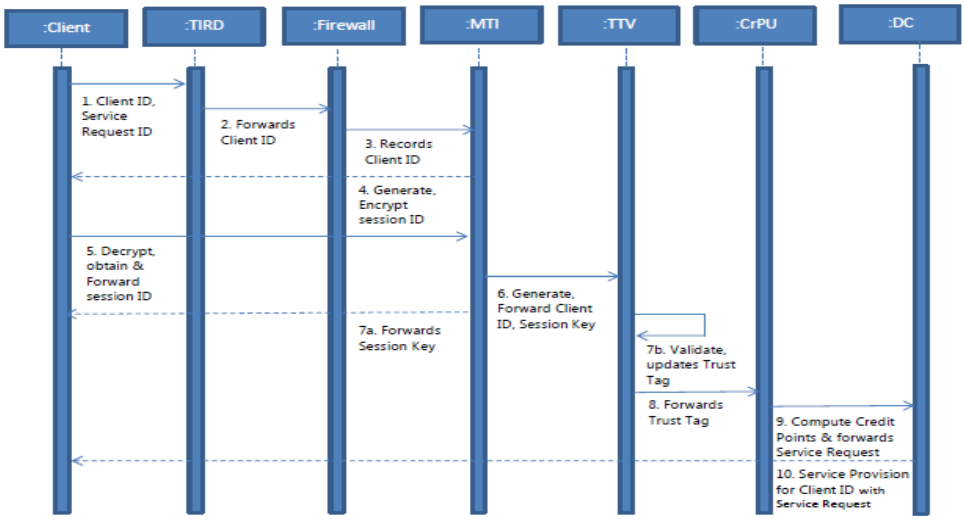

Fig. 2. Sequential flow diagram of the proposed Trilateral trust mechanism

4.1.1. Traffic injection rate detection

Whenever any clients are interested in getting services from DC, they initiate the service request to the Cloud Service Provider (CSP). At the cloud service provider end, the incoming requestor's traffic is fed into TIRD, which keeps sensing the traffic rate. TIRD will have its maximum capacity to handle and allow all incoming requests synchronously without any delay until the traffic rate is normal. TIRD senses the overload condition, if the incoming requestor rate is increased, at that time, TIRD queues the requests, thus creates a negligible delay and acts as a filter only when the overload traffic condition exists. This TIRD instead of being relative

traffic rate detection remains absolute traffic rate detection in nature. This absolute nature helps in indicating the precise deviation in the traffic condition, this indication is itself enough to precaution the CSP regards to the traffic overload, but this traffic overload can be the cause of legitimate traffic or illegitimate traffic. Both Legitimate and DDoS attackers will have the same request type. However, the overloading traffic pattern varies. This important clue gives the notion of segregating and processing the incoming requestor to validate whether the overload cause is due to legitimate or illegitimate and then a special treatment is given to a different overload condition in the for-coming levels before approaching the DC.

4.1.2. Mutual trust establishment

After bypassing TIRD, the traffic is fed into the firewall, when the new incoming requestor approaches towards DC for resource requisition; their request information is then forwarded to MTI. Here the requestor and DC would be trusted mutually via a digitally signed certificate. In order to reduce DC resource starvation, DC creates the $ID_{session}$ instead of maintaining a session before passing the digital signature. At MTI, the $ID_{session}$ is generated and encrypted by the secret key, which is shared between the requestor and DC. This $ID_{session}$ at MTI describes the ID_{client} and their attempt towards the DC resource requisition within a certain time frame. Now the generated encrypted $ID_{session}$ is forwarded to the intended requestor and DC awaits the acknowledgement from the requestor. At the requestor end, when the requestor is able to successfully decrypt with the shared secret key which lets the requestor to know $ID_{session}$, this concords the client that the DC is a valid and trusted CSP.

To obtain the $K_{session}$, the encrypted format of $ID_{session}$ needs to be appended with the ID_{client} which has to be acknowledged to DC. This is sufficient to prove that the requestor is the intended client with an ID_{client}. At DC end, MTI verifies the requestor's acknowledgement by fragmenting the ID_{client} and tries to decrypt with the help of the shared secret key to obtain $ID_{session}$ which is then matched with the MTI ID_{client} and $ID_{session}$ for the particular ID_{client}. Once matched, the requestor is considered legitimate. Then $K_{session}$ is generated and it is shared to the requestor by encrypting it again with the shared secret key and appended with an ID_{client}. Thus, at the client end, $K_{session}$ can be obtained by deciphering which can be used until the session expires. This ensures that secured connection has been established between DC and each client.

4.1.3. Trust tag

Trust tag is the absolute behavioral history tracker maintained for each client. This tag is created or updated at each session key generation, which indirectly states that this tag is computationally updated only for a requestor who passes MTI as a valid requestor and provides the complete historical behavior that helps in identifying the requestor traffic behavior.

Client ID – the identity to represent each client at CSP's DC, which is immutable, this client ID helps at initial authentication at DC end.

Session ID – a session ID is generated for each incoming requestor, but the requestors are considered to be vague clients until the session key generated for the particular session ID.

Secret key – the shared key maintained between any two legitimate clients and DC.

Credit points – helps in trusting the incoming based on the authentication and the momentary behavior.

Former session traffic rate – records the previous traffic behavior (requests/second).

Max data traffic – historic updated maximum traffic exploited towards the DC (requests/second).

Min data traffic – historic updated minimum traffic exploited towards the DC (requests/second).

Request inter-arrival rate – helps in identifying the type of traffic and classifies them to prevent the overload towards the DC based on the size and rate.

IP Address – the network address helps in identifying the client ID and for hashing with other variables to strengthen the key generation.

Current traffic rate – the traffic exploited towards DC in the current generated session.

Client ID	Session ID	Secret key	Credit points	Former session traffic rate	Max data traffic	Min data traffic	Request inter-arrival rate	IP address	Current traffic rate

This Trust tag is the essential component, it contributes to the major discrimination and improving of the primary notion of momentary trust computation. This trust tag is initially generated at TTV and sent to the intended client by encrypting it with the shared secret key. This achieves mutual trust and the historic behavioral transparency at the client end. In addition, this tag also helps in achieving the mutual trust between the CSP and the client. To improve the network's traffic efficiency, this tag is sent to the intended client only at the time of mutual trust initiation. This tag remains read-only at the client end and achieves integrity when encrypted by the shared secret key.

The fields in the trust tag are staged in order to make the detection firmer and quicker. The trust tag module is a kind of a simplex mode of communication. Thus, only the requestors who have successfully bypassed MTI are scrutinized by TTV and directed to CrPU, and vice versa is not applicable. Monitoring the requestors behavior and their periodic update requires consecutive network data exchange instead of updating the behavior at the Trust tag and storing at DC only at the session initiation and session expiration, which saves the network traffic at DC and valuable CPU and DC storage resource. This achieves the optimal traffic rate even if the network trafficker attempts an overwhelming traffic rate towards the victim DC. This Trust Tag exclusively helps in computing the behavior of each requestor which includes the historical behavior and the current session behavior and assists to decide by allowing or disallowing the incoming requestor.

4.1.4. Behavior monitoring

The proposed mechanism considers a zero trust approach. The zero trust approach is a scheme where all the incoming requestors are considered least trusted even if they holds considerable credit points to be a trusted requestor based on the historical behavior. But the requestor's historical behavior is not the only estimation in determining the trusted requestor, because the former trusted requestor can turn out to be the aggressive resource hunger requestor at this session. So, instead of treating the requestors as trusted/least trusted, based on the historical credit points, their behavior ought to be monitored for some initial identity initiation. Though the initial authentication takes considerable time at highly trafficked networks, it is significant for achieving the zero trust approach.

Since the Trilateral Trust mechanism follows the zero trust approach, even the well-trusted requestors must also be monitored and the misbehaving activity is to be navigated to CrPU.

The reason to monitor the behavior is to classify the incoming requestor based on its behavior. Different types of incoming requestor behavior have been discussed in detail in Subsection 4.2 which also unfolds the differential treatment of our trilateral trust mechanism for each and every classified behavior.

4.1.5. Credit delivery decision – session ranking

After the behavior monitor phase, the requestors are primarily classified as legitimate and illegitimate requestors. This would be achieved at the behavior monitoring phase. The trust tag helps in identifying the historical behavior. CrPU is the module responsible for credit points updating for each requestor which acquires the active session currently. CrPU additionally acts as a partial or intermediate server, and its functionalities are a continuous update of the credit points of the active session holders; signalling TTV and updating the Trust Tag at a new session initiation or expiration.

Mathematically, the requestor behavior in the network can be computed by the following equation. Requestor's behavior has an impact on the level of trust.

$$\mathrm{TrCP}_{requestor} = (\mathrm{CR}_{requestor} + \mathrm{RL}_{requestor} + \mathrm{IN}_{requestor}), \tag{1}$$

$\mathrm{TrCP}_{requestor}$ – Trust credit points of the requestor; $\mathrm{CR}_{requestor}$ – Credibility of the requestor; $\mathrm{RL}_{requestor}$ – Reliability of the requestor; $\mathrm{IN}_{requestor}$ – Intimacy of the requestor;

$$\mathrm{CR}_{requestor} = \mathrm{W}_{CR} * \mathrm{TC}_{CR}(t), \tag{2}$$

$$\mathrm{RL}_{requestor} = \mathrm{W}_{RL} * \mathrm{TC}_{RL}(t), \tag{3}$$

$$\mathrm{IN}_{requestor} = \mathrm{W}_{IN} * \mathrm{TC}_{IN}(t), \tag{4}$$

$$\mathrm{W}_{CR} + \mathrm{W}_{RL} + \mathrm{W}_{IN} = 1, \tag{5}$$

W_{CR} = weighted value associated with Credibility; W_{RL} = weighted value associated with Reliability; W_{IN} = weighted value associated with Intimacy; $\mathrm{TC}_{CR}(t)$ = Trust Credits based on credibility at time t; $\mathrm{TC}_{RL}(t)$ = Trust Credits based on reliability at time t; $\mathrm{TC}_{IN}(t)$ = Trust Credits based on intimacy at time t.

$$TC_{CR}(t) = \begin{cases} P.T(t-\Delta t) + (1-P)T(t) & \text{if existing requestor;} \\ T(t-\Delta t) & \text{if new requestor,} \end{cases} \quad (6)$$

$$TC_{RL}(t) = \begin{cases} PT(t-\Delta t) + (1-P)T(t) & \text{if existing requestor;} \\ T(t-\Delta t) & \text{if new requestor,} \end{cases} \quad (7)$$

$$TC_{IN}(t) = \begin{cases} PT(t-\Delta t) + (1-P)T(t) & \text{if existing requestor;} \\ T(t-\Delta t) & \text{if new requestor,} \end{cases} \quad (8)$$

P_{CR} = Historical behavior weighted value associated with Credibility; P_{RL} = Historical behavior weighted value associated with Reliability; P_{IN} = Historical behavior weighted value associated with Intimacy; Δt = time interval of the trust credit points updation in a trust tag;

$$W_{CR} = 1 - \frac{\text{RIRrequesor}}{\text{RIR max Limit}}, \quad (9)$$

$$W_{RL} = 1 - \frac{FSR_{requestor}}{FSR_{\text{ max Limit}}}, \quad (10)$$

$$W_{IN} = 1 - \frac{MDT_{requestor}}{MDT_{\text{ max NetworkLimit}}}, \quad (11)$$

$RIR_{requestor}$ =Request Interarrival Rate of an incoming requestor; $RIR_{\text{ max Limit}}$ = Max limit of the Request Interarrival Rate to be a legitimate requestor; $FSR_{requestor}$ = Former Session Rate of requestor; $FSR_{\text{ max Limit}}$ = Max limit of the Former Session Rate to be a legitimate requestor; $MDT_{requestor}$ = Data Traffic of the requestor; $MDT_{\text{ max NetworkLimit}}$ = Max limit of the Data Traffic to be a legitimate requestor; $TrCP_{requestor}$ is a probability measure ranging from 0 up to 1. To be optimal, the credit points of 0.6 and above it seem to be a trusted requestor based on the network traffic. The optimal credit points could be varied or configured based on the network traffic rate. The network traffic rate is inversely proportional to the trust credit point of requestor. The increase of W_{CR}, W_{RL}, W_{IN} increases TrCP of the requestor.

Active session is ranked based on the credit points and allowed access, which results in allowing innocent and well trusted requestors and will be better provisioned, whereas the least trusted and aggressive requestors are preferred next and the distrusted requestors are prevented and filtered at TIRD.

4.1.6. Scenario based routing

Requestors are primarily of two types which can be seen in Fig. 3. They are trusted requestors and distrusted requestors. Then they are again classified further to have a clear and differential treatment based on the incoming requestor behavior.

Though Trusted Requestors are allowed to access DC, based on the CrPU momentary trust value, the trusted requestors are again classified as the most-trusted and least-trusted requestors. So, even when the most-trusted requestors deviate from the legitimate traffic behavior, they are spotted as aggressive requestors and considered as the least-trusted requestors. Any most-trusted

requestors who violate the legitimate traffic will be considered as the least-trusted requestors for the session and the CrPU is updated respectively. This expires the session for the concern misbehaved requestor and TTV is intimated about the session expiry and the trust tag is also updated. By the way, a different kind of distrusted traffic behavior is detected at different levels and their connection is aborted. Usually the malicious traffic, which is considered to be of a higher rate (i.e., DDoS and Botnets) is detected at MTI. Eventually, the spoof attacker is detected at TTV. In this way the illegitimate traffic is examined and rejected from being entered into CSP. The reason for considering the flash crowd being the most-trusted traffic is that this traffic is originated from several requestors at the same time and lived for a short period of time. Different types of requestor behavior can be seen in Subsection 4.2 in detail. Each type of a requestor is treated differently and routed to the successive level based on the legitimacy proven at each level, otherwise the requestor is occluded and not allowed for service requisition towards DC until the session expiration.

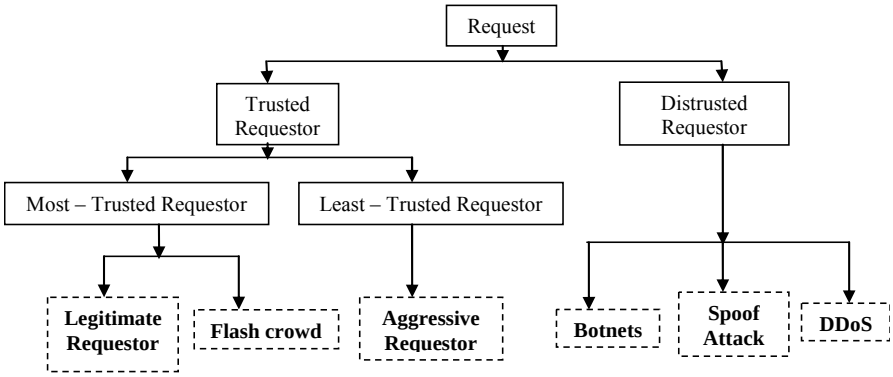

Fig. 3. Behavior based requestor classification

4.2. Cases considered for diminishing the traffic overload at the Data Center

4.2.1. Spoof attack behavior

Spoof attack behavior is the impersonation behavior whose aim is data stealing at DC or creating the loss of trust credit points of the victim requestor. This type of an attack is hard to detect because the attacker imitates the legitimate requestor but the intention differs. The proposed TTM has MTI which supports identity initiation. At MTI, for each incoming requestor, a session ID is generated, ciphered with a shared secret key and forwarded to the intended requestor. The ciphered message can only be deciphered by the intended requestor, upon successful obtaining of the session ID, ciphering again the session ID and appending it with the client ID improves the performance by verifying the appropriate requestor quicker. If the requestor fails in obtaining the session ID, the identity initiation fails and spoof attack behavior is considered. Successful decipher results in generating session keys for further communication in the particular session. The shared secret key cipher is more significant in verifying the incoming requestor and the computation is much

negligible as the identity initiation is performed for all incoming requestors. It is suggested to use simple ciphers rather than the usage of a complex cipher which troubles the mechanism, creates the computation overhead and also results in increased overload instead of detecting the incoming requestor.

4.2.2. DDoS attack behavior

DDoS attack is a group of several hundreds of requestors who simply overload and subvert the DC. At TIRD, the abnormal traffic rate is observed. Based on the request size and request rate at TTV, the DDoS attackers can be easily detected as the request rate and request size deviate the legitimate requestors, even though some attackers follow a legitimate behavior and bypass MTI. This deviation is intimated to TIRD and a firewall to prevent the entry until the session expires.

4.2.3. Flash crowd behavior

Flash crowd is a sudden increase in the traffic rate, which is caused by several legitimate requestors attempting to access DC at the same point of time. This event can be detected at TTV, because the Trust tag contains the sufficient fields to characterize the requestor behavior. If the incoming requestors' request inter-arrival rate is minimum and at the same time if the number of the requestor is huge, it is easier to predict the incoming requestor that the session ID involves a flash crowd event. Moreover, the current traffic rate would also follow a legitimate behavior for any requestor involving a flash crowd event. Since each requestor involved in a flash crowd follows the legitimate behavior, the flash crowd is considered legitimate traffic behavior.

4.2.4. Aggressive legitimates behavior

Aggressive legitimates are the requestors who follows the legitimate behavior by holding the sufficient trust credit points. This shows that mutual trust has been established and identity initiation was successful. But the legitimate requestor cannot be assured as a long-lasting trusted legitimate requestor. In order to suffice an aggressive legitimate behavior, the trust tag supports momentary trust computation. If the current traffic rate or request inter-arrival rate exceeds the legitimate behavior for any active session holding requestor, the legitimate requestor is considered an aggressive legitimate requestor, also known as resource hunger requestor.

4.2.5. Botnets behavior

Botnets are the robotic networks setup to compromise the DC. It keeps injecting the spurious requests towards DC to create overload and to thrash down the performance of DC. Usually this kind of an attack randomly shifts the IP address and overloads DC. These kinds of attackers simply launch high rate traffic towards DC with the aim to bring down the DC performance. These botnet machines can be easily detected because they fail at MTI authentication which is intimated to TIRD

and a firewall. So, botnets are filtered at TIRD which is one of the highest rate overload conditions.

4.2.6. Trusted legitimates behavior

Trusted legitimates are the requestors who always follow the most optimal traffic behavior. This type of requestors not only follow the current legitimate traffic behavior and good trust credit points, but also follow the former session legitimacy which allows maintaining better historical behavior. The Trusted Legitimates successfully pass the identity initiation and establish mutual trust and considerably maintain better momentary trust by not attempting to overload DC.

5. Experimentation and performance evaluation

5.1. Experimental setup

Jeyanthi and Iyengar [8, 9] used OPNET as a simulator to test the cloud computing environment. Jeyanthi et al. [10] experimented DDoS in cloud computing. We tested our proposed mechanism as a simulation experiment in OPNET Modeller [11, 12]. The experiments are performed in a campus network where DC requesters are grouped in three subnets and each subnet has got 100 workstations, 100 attackers and 200 legitimate clients requesting for application-specific requests at each subnet. In this way we created the attacker and a legitimate profile and other devices, which would be needed to test our algorithm in an experiment. The traffic represents Internet and the group of spoof attackers is activated at varying time intervals. The attack profile is replicated to increase the attack strength to engage DC resources like bandwidth, CPU and memory. On the whole, our experiment has 600 clients and 300 attackers activated and de-activated at various time intervals. The experiment is carried out with different scenarios, namely the network with attackers where no detection mechanism is in place, and the network with attackers where the proposed Trilateral Trust mechanism is in place.

5.2. Performance evaluation

5.2.1. Overall network traffic rate

The overall network traffic rate is the statistic which is the average number of bytes forwarded per second from the email application and File Transfer Protocol (FTP) application clients towards the DC. Here the application clients include legitimates and also the attack traffic.

Fig. 4 shows the comparative result of the overall network traffic rate with the proposed Trilateral Trust Mechanism and without a Trilateral Trust Mechanism. Without the Trilateral Trust Mechanism, the traffic rate tends to be higher. As it was uncontrolled, the traffic rate reaches the maximum and continuously grows higher. In contrast, the traffic rate with a trilateral trust mechanism appears to be well controlled, as the overload conditions have been detected at earlier stages.

Thus, the network traffic rate seems to be resilient even at the time of overload. The traffic oscillation in Fig. 4 is due to the attacker exclusion and rejoining at several points of time. The traffic rate with trilateral trust proves the considerable difference and improved performance of the traffic reaching DC.

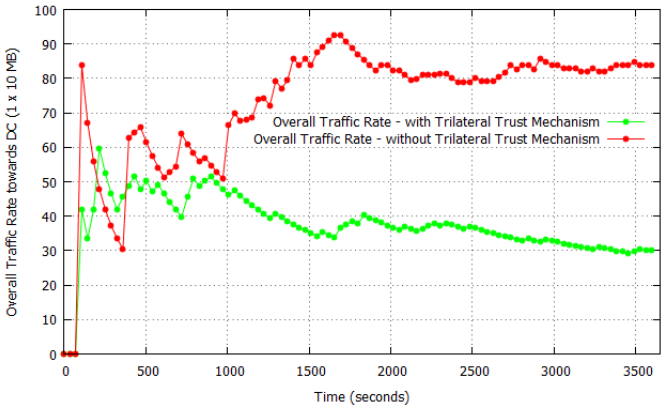

Fig. 4. Overall network traffic rate

5.2.2. TIRD filtration

TIRD filtration is the statistic measure indicating the number of requests being filtered due to the illegitimate and abnormal traffic conditions approaching DC. Fig. 5 shows the TIRD filtration of the proposed Trilateral trust mechanism. At the beginning, TIRD signals the overload traffic condition. After a considerable period of time (120 s approximately), the number of abnormal traffic was detected and filtered.

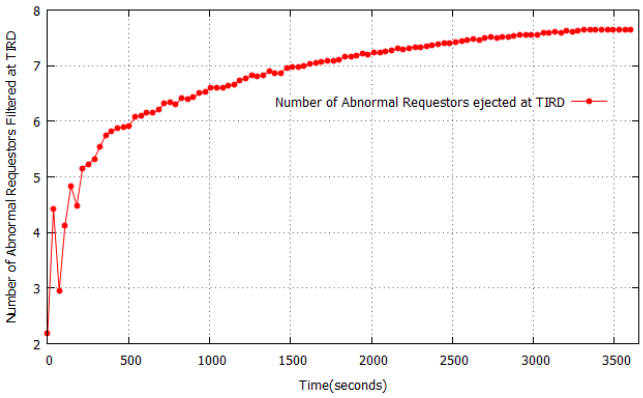

Fig. 5. TIRD filtration

After a prolonged period of time (900 s approximately), the requestors are continuously dropped almost exponentially. The requestors filtered seem to be heavier, because the filtered requestors attempt to enter the DC, but the abnormal requestors log has been monitored continuously and blocked from entry and DC resource usage. Thus, once the abnormal requestor is filtered, they cannot enter

again until their session expires because once the requestor is filtered, their detail is forwarded to TIRD and the firewall to prevent their entry which allows the new incoming requestor to be served.

5.2.3. MTI hold-off time

MTI hold-off time is the statistic measure showing the number of requestors being processed. This measure also helps in identifying the active sessions allocated for the requestors.

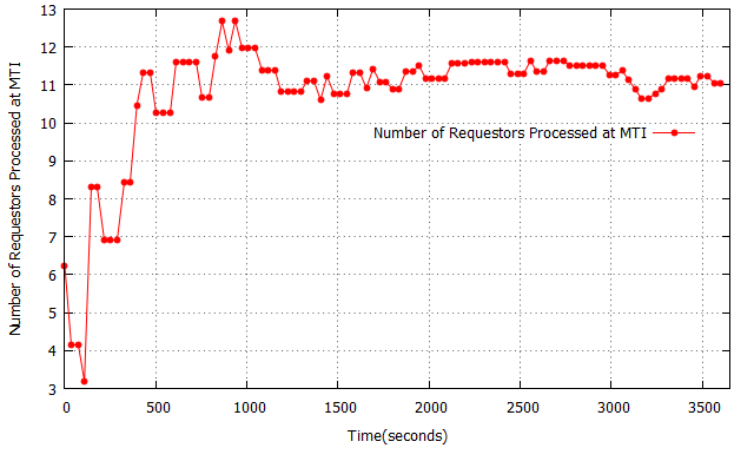

Fig 6. MTI hold-off time

Fig. 6 shows the MTI hold-off Time of the proposed TTM. At the beginning, the number of requestors processed was smaller in number because of no or less overload. Gradually the incoming traffic rate increases which can be seen in Fig. 4. Fig. 6 shows the number of requestors successfully bypassing TIRD, here the identity is initiated between MTI (DC end) and the requestor. Once the identity initiation fails, that particular requestor is again blocked at TIRD and the firewall. This aids to detect a spoof attack. The increase in the number of requestors processed at MTI shows the new requestor has been given the chance to be served by DC and the misbehaved requestor is blocked at TIRD. So, as the botnet and spoof overload is detected, the number of requestors processed at MTI increases.

5.2.4. Trilateral trust dynamic signaller response time

TT Dynamic signaller response time is the statistic unit measuring the time taken for TTV to efficiently acquire a Trust tag and to monitor the historical behavior.

Fig. 7 shows the TT Dynamic Signaller Response Time of the proposed Trilateral trust mechanism. Once the identity initiation is successful at MTI, at TTV, the Trust tag is acquired for the particular requestor. Though the number of requests was high, still the response time at TTV is gradually reducing and at some point of time it is almost zero which shows the improved performance.

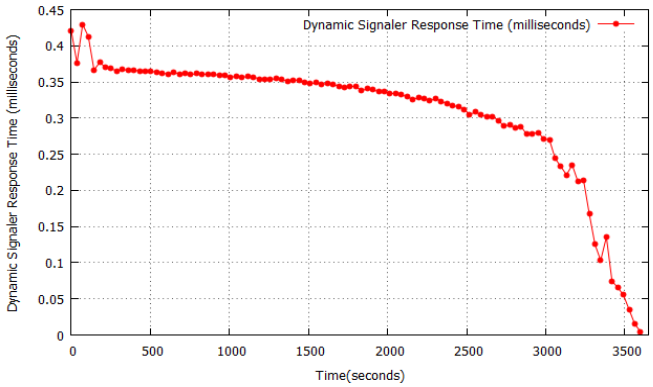

Fig. 7. Trilateral trust dynamic signaller response time

At a certain (120 s approximately) time, the overload surges severe which can be seen in Fig. 4, but the traffic reaching at MTI is the controlled traffic which is filtered and fed into MTI. From Figs 4 and 5 the overall traffic rate and the filtration rate can be observed, which indicates that further on TTV will have legitimate and controlled traffic. So, after a sustained period of time, the response time drops down. This shows improvement in performance at TTV.

5.2.5. Trust credits computation

Trust credits computation is the statistic measure of the time taken to process the requestor behavior based on the trust tag that belongs to an active session and the time taken to periodically update the requestor trust credit points while the active session is closed.

Fig. 8 shows the Trust Credits Computation of the proposed TTM. The trust credit points computation depends on TTV and acquiring the trust tags. The computation seems to be minimal initially, as the number of requestors in the network is low, but at (120 s approximately), the time spent in trust credit point computation increases due to the larger number of requestors processed at that particular time.

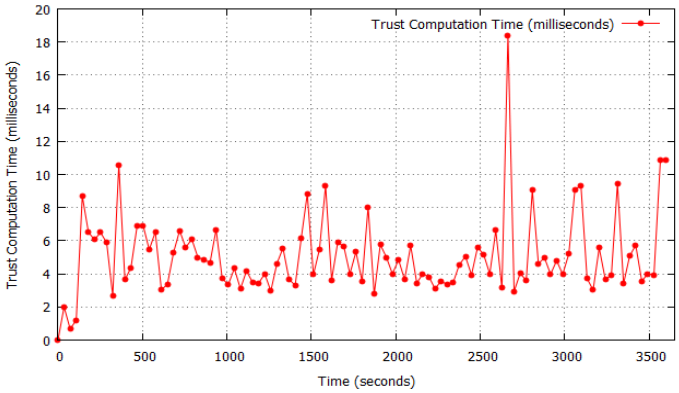

Fig. 8. Trust credits computation

Trust credit point computation processing time keeps oscillating as the legitimate and flash crowd closes the active session and the trust credit point is updated at the trust tag. The oscillation is also due to the aggressive legitimate detection and their trust credit points updated. Though Fig. 7 shows a better response in fetching the trust tag, but due to accepting new connections and closing some active legitimate connection requires trust credit computation and updating the appropriate trust tag of the requestor which shows some oscillating behavior in performance. This practice of updating the trust tag helps in quicker task processing time in the subsequent trust tag verification of a particular requestor.

5.2.6. Application response time

Application response time is the statistic unit which measures the time elapsed between sending a request and receiving the response packet. It is the measure of the time from when a client application sends a request to the server up to the time it receives a response packet. This time includes the signalling delays for the connection set-up.

Fig. 9 shows the comparative result of the Application response time with a proposed TTM and without a TTM. Without a TTM the response time at the time of DDoS increases exponentially which results in poor performance. Initially the response time with a TTM and without a TTM appears similar but once the attack traffic is detected at TIRD, considerable difference can be seen in the response time which proves the improved performance. The oscillation shows the added legitimate traffic to the network at a certain point of time. The response time shows a drastic gain in the response time and a quicker task processed at DC. The performance improvement shows almost 100% by reducing the response time much shorter even at the time of overload.

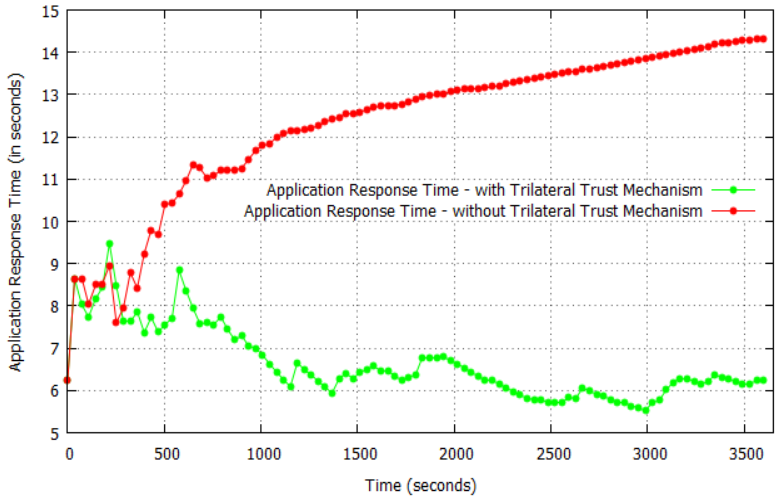

Fig. 9. Application response time

5.2.7. GoodPut

Goodput is the statistic measure of the rate at which the legitimate packets reach the destined DC. The increase in goodput assures the decrease in legitimates' packet loss and retransmission.

Fig. 10 shows the comparative result of goodput with a proposed TTM and without a TTM. Goodput increases gradually which shows that the attacker traffic still prevents the legitimate traffic. When a considerable amount of attack traffic has been detected, the goodput increases and constantly remains the same. This proves that legitimates are allowed to access DC even at the time of DDoS. Fig. 10 shows that the proposed Trilateral Trust mechanism performs around 6 times better improvement even at the time of DDoS. Though at the initial stage the goodput remains the same as without a trilateral trust, upon detecting and filtering the attack traffic at TIRD shows drastic improvement in goodput.

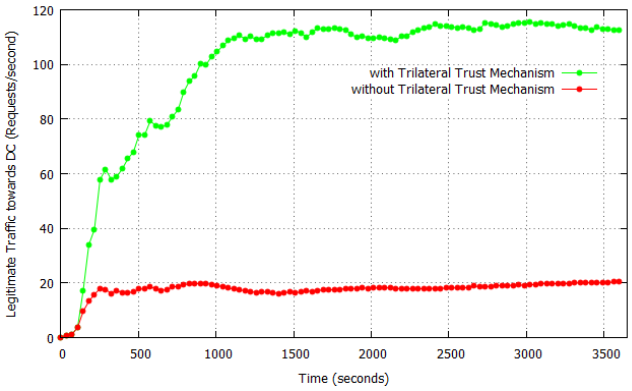

Fig. 10. Goodput

5.2.8. Behavior forecast

Behavior Forecast is the statistic tool measuring the number of illegitimate requestors being rejected at acertain point of time which enter the networks as new requestors. This rejection also includes the aggressive legitimates and other misbehaving characteristic of the requestors. Fig. 11 shows the behavior forecast of the proposed TTM. Here the behavior forecast is the behavior monitoring scheme at various levels. Initially, the abnormal requests were high which is due to DDoS attack initiation. Once identified, they will be blocked at TIRD which is shown in Fig. 5, which improves MTI hold-off time, which in turn again improves the Dynamic signaller response time.

When these improvements continue for a certain period of time, the application response time is much quicker even at the time of DDoS. Even if a group of attackers launches overload towards DC, they are blocked at TIRD and other new attackers initiation reaches MTI, and thus the behavior forecast gradually decreases because a part of the attack traffic has already been filtered at TIRD. The behavior forecast at all earlier levels like TIRD, MTI, and TTV proves the network resiliency with minimal abnormal traffic attempt to reach MTI as a new requestor.

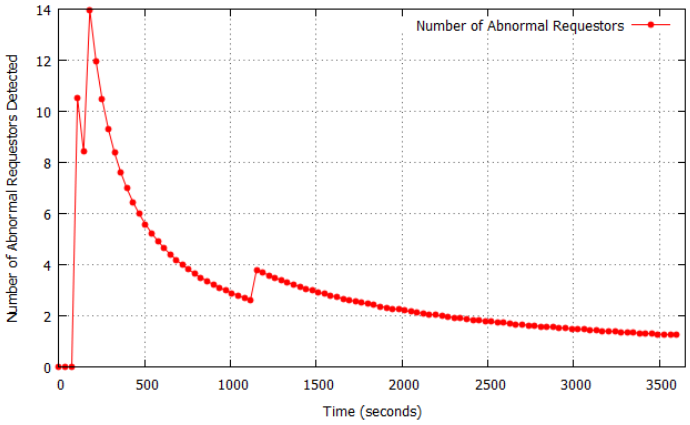

Fig. 11. Behavior forecast

5.3. Advantages of the proposed mechanism

5.3.1. Profit analysis

The cost is computed based on the data transmission and memory resident operations at each DC, based on an average sample that is a combination of attack traffic and legitimate traffic.

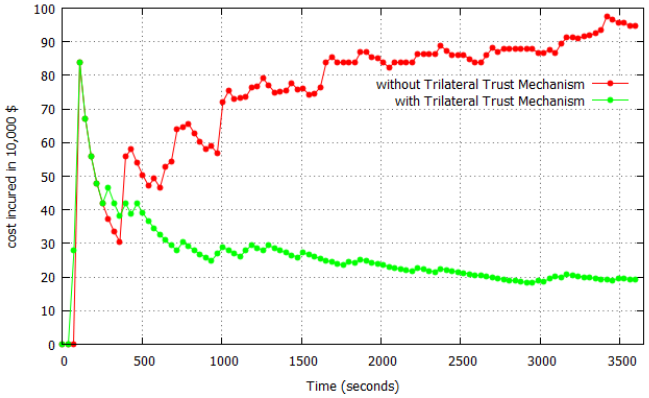

Fig. 12. Profit analysis

Let N = time in hours; CI_{BW} = Bandwidth cost; CI_{MEM} = RAM cost of each physical equipment; CI_{VM} = VM cost of each physical equipment and CI_{DS} = Data stored within a DC.

The total cost incurred at

$$DC = \sum_{i=1}^{N} \{CI_{BW} + CI_{MEM} + CI_{VM} + CI_{DS}\}. \quad (12)$$

Fig. 12 shows the huge cost incurred at the victim DC. When the proposed Trilateral Trust Mechanism is in place, the cost incurred at the DC improved the revenue, which results in resource protection and is used only for legitimates that in turn result in resource availability. The costs used are $ 0.1 per 1 Gb for any data

transmission at the DC and $ 0.05 per 1 s for any memory resident operations at the DC. The extreme difference in profit is due to detection of an attacker at their initiation and preventing their subsequent entry towards the DC.

This paves the way to improve the availability with an acceptable response time and goodput shown in Figs 9, 10. In addition to the improved detection efficacy, other benefits have been observed that would improve the choice of deployment. This also proves the advantageous feature of being resilient even at the time when DDoS attack scenario prevails in the network.

6. Conclusion and future work

DDoS attacks are very common attacks that exhaust the resources of the DC. Such an attack is easier to launch and difficult to be detected. Therefore, it is necessary to deploy the detection mechanism that identifies each requester and their incoming traffic rate to detect whether the incoming requester is a legitimate user or not.

The proposed Trilateral Trust mechanism contains several stages to monitor the traffic behavior like TIRD that acts as a traffic beacon signaller and MTI, which acts as an authentication protocol. Though authentication is essential, monitoring them momentarily with trust parameters is also important to detect the aggressive legitimates. Detecting the high rate attackers at an earlier stage allows better and resilient network performance. Deviating requestors will have differential treatment in the mechanism proposed. Since the proposed mechanism follows a trust protocol with zero trust approach, it is possible to prevent DDoS attackers up to some period of time. The proposed Trilateral Trust mechanism simulation results show improved performance.

The future work is not only to derive a hybrid trust mechanism but also to develop an enhanced hybrid approach to protect the DC resource from anti-viral, malware injection, source criticism and DNS amplification and reflection attacks. Since detecting the overload condition in a network is a crucial and critical task, the detection system should be light-weight. Hence, it is planned to deploy the perimeter-centric mechanism in addition to authentication and trust in an enhanced way. This would handle the overload conditions of newer scenarios at quite earlier levels to serve the legitimate users better than ever.

References

1. **https://www.cloudflare.com/under-attack** (accessed on 1 December 2014)
2. **http://www.thedailybeast.com/articles/2010/12/11/hackers-10-most-famous-attacks-worms-and-ddos-takedowns.html** (accessed on 1 December 2014)
3. A b d u l-R a h m a n, A., S. H a i l e s. Using Recommendations for Managing Trust in Distributed Systems. – In: Proc. of IEEE Malaysia International Conference on Communication (MICC'97), 1997.
4. L i u, B., Z. Y u a. Incorporating Social Networks and User Opinions for Collaborative Recommendation: Local Trust Network Based Method. – In: Proc. of the Workshop on Context-Aware Movie Recommendation, ACM, 2010, pp. 53-56.

5. Zacharia, G. Trust Management Through Reputation Mechanisms, Workshop in Deception, Fraud and Trust in Agent Societies. – In: Third International Conference on Autonomous Agents (Agents'99), ACM, 1999.
6. Eschenauer, L., V. D. Gligor, J. Bara. On Trust Establishment in Mobile Ad Hoc Networks. – Security Protocols Springer, 2004, pp. 47-66.
7. Iyengar, N. C. Sriman Narayana, Gopinath Ganapathy, P. C. Mogan Kumar, Ajith Abraham. A Multilevel Thrust Filtration Defending Mechanism Against Ddos Attacks in Cloud Computing Environment. – International Journal of Grid and Utility Computing, Vol. **5**, 2014, No 4, pp. 236-248.
8. Jeyanthi, N., N. C. S. N. Iyengar. Packet Resonance Strategy: A Spoof Attack Detection and Prevention Mechanism in Cloud Computing Environment. – International Journal of Communication Networks and Information Security, Vol. **4**, 2012, No 3, pp. 163-173.
9. **http://www.opnet.com/news/press_releases/pr-2010/OPNET-Introduces-Cloud-Readiness-Service-pr.html** (accessed on 1 December 2014)
10. Jeyanthi, N., N. C. S. N. Iyengar, P. C. Mogan Kumar, A. Kannammal. An Enhanced Entropy Approach to Detect and Prevent DDoS in Cloud Environment. – International Journal of Communication Networks and Information Security, Vol. **5**, 2013, No 2, pp. 110-119.
11. **http://www.opnet.com./services/brochures/OPNET_CloudReadiness.pdf** (accessed on 1 December 2014)
12. Jeyanthi, N., C. Iyengar. Escape-on-Sight: An Efficient and Scalable Mechanism for Escaping DDoS Attacks in Cloud Computing Environment. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 1, pp. 46-60.