Journal of Computer Sciences and Applications, 2017, Vol. 5, No. 2, 50-63
Available online at http://pubs.sciepub.com/jcsa/5/2/2
© Science and Education Publishing
DOI:10.12691/jcsa-5-2-2

A Security Scheme to Mitigate Denial of Service Attacks in Delay Tolerant Networks

Godwin Ansa[1,*], Haitham Cruickshank[2], Zhili Sun[2], Mazin Alshamrani[3]

[1]Department of Computer Science, Akwa Ibom State University, Mkpat Enin, Nigeria
[2]Institute of Communications Systems, University of Surrey, Guildford, United Kingdom
[3]Studies and Decision Support Center, Department of Planning and Development, Ministry of Haj, Saudi Arabia
*Corresponding author: godwinansa@aksu.edu.ng, godwin_unique@yahoo.com

Abstract Denial of Service (DoS) attacks are a major network security threat which affects both wired and wireless networks. The effect of DoS attacks is even more damaging in Delay Tolerant Networks (DTNs) due to their unique features and network characteristics. DTN is vulnerable to resource exhaustion and flooding DoS attacks. Several DoS mitigating schemes for wired and wireless networks have been investigated and most of them have been found to be highly interactive requiring several protocol rounds, resource-consuming, complex, assume persistent connectivity and hence not suitable for DTN. To mitigate the impact of resource exhaustion and flooding attacks in DTN, we propose a security scheme which integrates ingress filtering, rate limiting and light-weight authentication security mechanisms to monitor, detect and filter attack traffic. We propose three variants of light-weight bundle authenticators called DTNCookies. To make the proposed DTNCookies random and hard to forge, we exploit the assumption that DTN nodes are loosely time-synchronized to generate different nonce values in different timeslots for the computation and verification of our proposed DTNCookies. The results demonstrate the efficiency and effectiveness of the proposed scheme to detect and drop attack traffic. The simulation results also show good performance for the proposed scheme in terms of energy and bandwidth efficiency, high delivery ratio and low latency.

Keywords: *denial of service, DTNCookie, flooding, resource exhaustion*

Cite This Article: Godwin Ansa, Haitham Cruickshank, Zhili Sun, and Mazin Alshamrani, “A Security Scheme to Mitigate Denial of Service Attacks in Delay Tolerant Networks.” *Journal of Computer Sciences and Applications*, vol. 5, no. 2 (2017): 50-63. doi: 10.12691/jcsa-5-2-2.

1. Introduction

In today’s world there are a variety of network deployments some in very remote regions of the world with very extreme conditions which make communications difficult or near impossible. These networks are referred to as “Challenged” networks because they do not conform to the existing Internet protocol semantics. The success of the Internet is largely due to its ability to interconnect communication devices globally using a homogeneous set of protocols, called the Transmission Control Protocol/Internet Protocol (TCP/IP) suite. The present Internet architecture is built on the assumption that there is a continuous bi-directional link between a communicating source and destination. The delay in sending and receiving packets is relatively small, data rates between two communicating entities are symmetric, and the rate of packet loss and error is low [1].

DTN is an overlay network on top of a number of diverse regional networks such as Mobile Ad hoc Networks (MANETs), Wireless Sensor Networks (WSNs), the Interplanetary Internet, Satellite Networks and the Internet. The DTN overlay provides interoperability across these varying network characteristics to provide a service that works regardless of the difficult conditions of the underlying networks. DTN is characterized by limited bandwidth, long queuing delays, low data rates, delivery latency, intermittent connectivity due to frequent disruptions, and scarcity of resources such as battery power, CPU processing cycles, bandwidth and memory. It uses the carry-store-and-forward message switching technique and the inherent mobility of nodes to overcome these constraints and deliver bundles to a destination. DTN introduces a new protocol layer**,** *the Bundle Layer*, which sits on top of the transport layer.

In [2] DTN is defined as an Overlay architecture which introduces a new protocol layer above existing protocol stacks of other heterogeneous networks where gateway functionality help in the interconnection of these disjoint networks. Communication impairments are overcome using replication, parallel forwarding and error correction techniques. DTN as a networking concept and architecture was proposed by the Internet Engineering Task Force (IETF) with pioneering work on the Interplanetary Network (IPN) [3]. DTN has gained popularity over the years with several research in areas such as the Interplanetary Networks (IPNs) for space and satellite communication [3], Airborne Networks (ANs) [4], Delay-Tolerant Sensor Networks [5], Vehicular Ad hoc Networks [6], Underwater Networks (UWNs) [7] and Pocket-Switched Networks (PSNs) [8]. A number of research works have been carried out and are still on-going in DTN routing [9], congestion control/buffer management [10], convergence layer design [11], application layer design [12], and flow control [13] but very little on DTN security.

In communication networks, there are key components that provide critical services such as monitoring or query access points, routers, gateways, cryptographic key managers, and network uplinks. This infrastructure can come under serious DoS attacks when an attacker sends a large number of requests which engage any of these key components in computationally intensive authentication protocol. Therefore, protecting the DTN infrastructure and controlling access to the network is critically important. Providing security to challenged networks like DTN requires new techniques. This is due to the wireless multi-hop communication which makes the channel open to attacks, lack of infrastructure, changing network topology due to mobility, intermittent connectivity and limited power budget of participating nodes. Disruptions are caused by limited communication range, sparse density of nodes, attacks and noise. Due to its unique characteristics, DTN is vulnerable to packet injection, flooding, modification attacks, eavesdropping, and unauthorized access/use of its scarce resources. Standard security protocols like Transport Layer Security (TLS) [14] and Internet Protocol Security (IPSec) [15] require more than one protocol round to exchange cryptographic materials and agree on the cryptographic ciphersuites (algorithms). The round-trip delay of these traditional protocols to establish secure connections make them not suitable for DTNs since message transfer is opportunistic. To encourage large-scale deployment and use, DTN must guarantee secure and reliable communications. In designing protocols to secure a DTN, such designs have to be very efficient and light-weight to guarantee and prolong the life of the network.

We look into the aspect of service availability which is one critical requirement for computer and communication networks. Availability guarantees that requests of authorized entities are satisfied in a timely manner. The aim of DoS attacks is to prevent a network from fulfilling its functions by disabling, degrading and making network services unavailable to legitimate users. In a DTN, network capacity is scarce and connectivity is infrequent. The DTN security architecture includes a hop-by-hop mechanism to provide authentication and integrity to protect the network from unwanted traffic. The security architecture also supports end-to-end data integrity and confidentiality. However, DoS attacks are still an open problem in DTN research. In this paper we propose a comprehensive defense scheme against flooding and resource exhaustion DoS attacks. In the proposed scheme, a gateway uses ingress filtering to detect attack bundles with randomly spoofed source addresses. To prevent flooding attacks we incorporate a rate limiting mechanism to the defense scheme, each traffic flow is monitored and gateways are assigned different thresholds. Traffic flows exceeding the set thresholds are blocked for a set period. Attack bundles with spoofed gateway addresses are detected and dropped during the verification of our proposed DTNCookie. The aim of the proposed scheme is to mitigate the effects of flooding and resource exhaustion DoS attacks and ensure the availability of DTN resources (i.e. communication contact time (link), battery, memory and CPU processing cycles) to legitimate users.

Denial of service attacks have been studied extensively in traditional terrestrial networks like the Internet. A number of solutions have been proposed in tackling flooding and resource exhaustion attacks [16,17,18,19,20]. These works cannot be easily extended to DTN due to its architecture and network characteristics. For example nodes in the terrestrial

Internet are fixed and well-connected and can support several message exchanges for connection establishment. In DTN, connectivity is achieved when nodes come in communication range with each other through the inherent mobility of the nodes. Most of the existing works in DTN research focus on routing and the dissemination of data with little emphasis on security. Security is one of the major challenges impeding the rapid and large-scale deployment of DTN. Security threats in DTN such as Resource exhaustion attacks [21], flooding of bogus messages [22], bundle dropping [22], routing table corruption [22], counterfeiting acknowledgments for bundle delivery [22] have been addressed in earlier works on DTN security. Other identified attacks include blackhole [23] and wormhole attacks [24]. Lee et al [25] proposed a queuing mechanism to combat flooding attacks on probalistic DTN routing algorithms. Choo et al. [26] investigate the robustness of DTN routing without the use of an authentication mechanism. An authentication scheme which uses Identity-based Cryptography (IBC) [27,28,29] is possble in DTN.

2. System Model and Design Goals

Figure 1 (a) and Figure 1 (b) show the intra-region and inter-region scenarios respectively. The focus of this work is on the inter-region communications in terrestrial DTNs.

In our adopted scenario, we opt for a more general DTN and focus on providing DoS resilience in the inter-region scenario. The hosts in this scenario are message custodians which we refer to as gateways. Regional gateways are fixed and act as inter-connection points. Mobile sinks such as data mules [31,32,33] are examples of mobile gateways. Inter-region communication is enabled by data mules which visit the regional gateways to deliver bundles destined for a particular region and collect messages that are destined for other regions. Examples of data mules include satellite, car, bus, train and aeroplane as shown in Figure 1 (b). The gateways have a wide communication range and good reception capabilities and communicate using high-speed links such as WiFi. An end-to-end path is not always guaranteed so messages are routed in a scheduled manner. Figure 1 (a) shows the topology of each region depicted in Figure 1 (b) in great detail. The gateways are modeled as stationary. The data mule uses the Map-based model and moves at a uniform speed between 105-118 km/hr and pauses on reaching a region for a period between 0 and 5 seconds. A legitimate gateway generates 1 bundle per minute and randomly selects a destination gateway. Each generated bundle is 1.5 Megabytes in size. Communication between the gateways and the data mule is bi-directional with a transmission speed of 54Mbps. The communication range of each gateway is 300 meters.

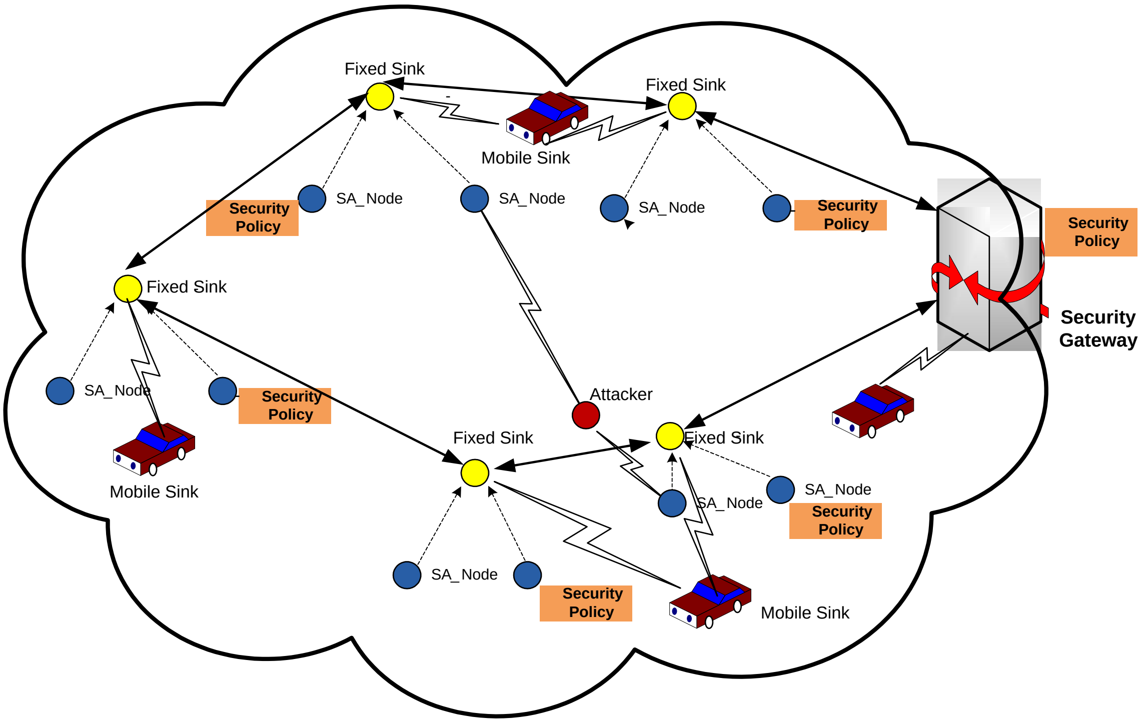

Figure 1(a). Intra-region scenario: A DTN region

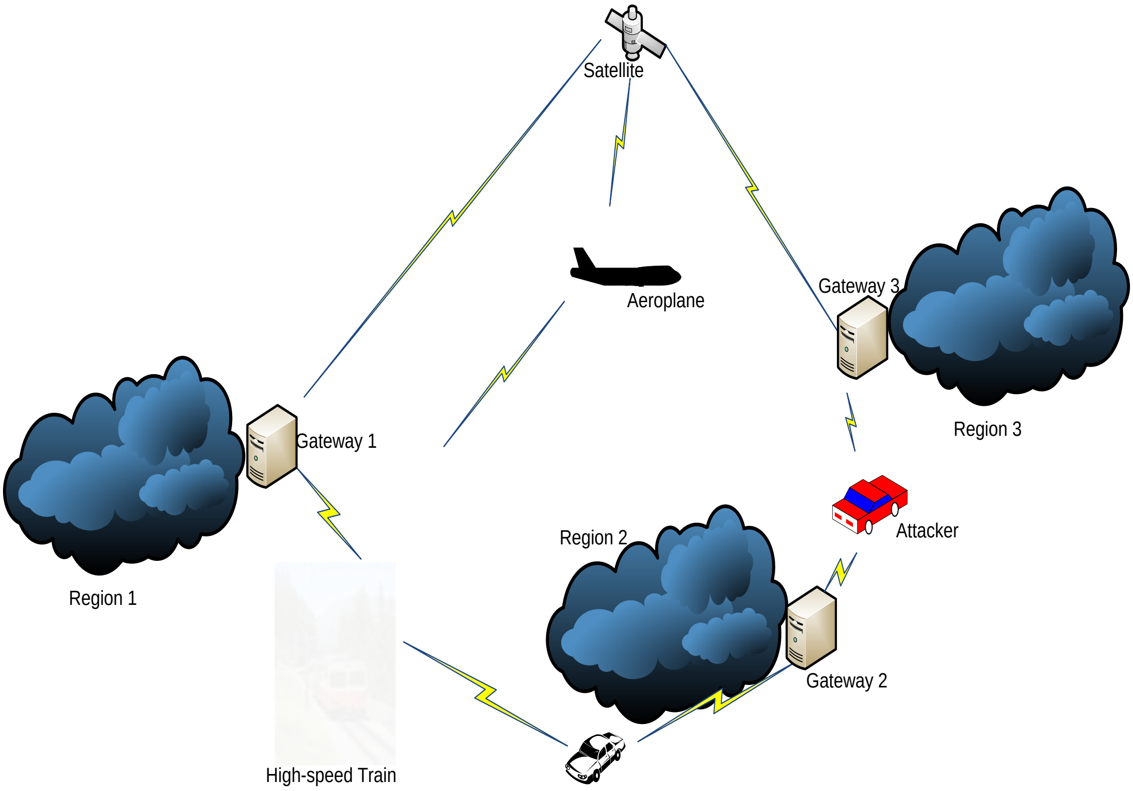

Figure 1(b). Inter-region scenario: DTN regions connected via gateways and data mules

Protecting a system against DoS attacks involves the three cycles of preparation, detection and reaction [30]. In the ***preparation phase*** actions such as over provisioning of capacity, security policy creation, selection of good security protocols, the monitoring of on-going operations (packet rates, CPU and memory utilization) to distinguish between normal and abnormal behaviour. The ***detection phase*** is quite critical and important because the ability to detect attacks directly affects how the system reacts to such attacks and minimizes the possibility of damage [30]. The detection phase should be automatic and the response to DoS attack swift. Late detection of a DoS attack leads to the degradation of availability of critical services.

The ***reaction phase*** involves the characterization and mitigation of attacks. During the *characterization phase*, the victim classifies the incoming traffic in order to determine if an attack is on-going. The classification helps the victim to distinguish between normal traffic (sent by legitimate nodes) and attack traffic (sent by malicious nodes) [30]. A good characterization will lead to a proper understanding of the nature of attack. In the *mitigation phase*, the victim uses the knowledge in the attack characterization sub-phase to deploy appropriate defenses to defuse the attack.

2.1. Attack Model

The goal of the attacker is to inject bundles or flood the network with bogus bundles in order to gain unauthorized access to DTN resources. This causes a depletion in the energy of DTN nodes. The attacker is mobile and can replay, transmit, and modify bundles. We assume that the attacker is able to perform localized flooding during a connection opportunity since most nodes in the DTN are unavailable most of the time and there is no direct path from a source to a destination. The attacker has the ability to generate a large volume of bundles to overwhelm the victim node. The attacker also exploits the mobility pattern to attack all nodes within its communication range. Alternatively, the attacker can permanently be within the range of one node in the network and cannot compromise DTN nodes.

2.2. Design Goals and Assumptions

The scheme should have a number of properties to be considered efficient and suitable for DTN.

- Portability: simple to deploy and it should be compatible with a number of devices and routing protocols.
- Effectiveness: the proposed scheme should be effective in identifying and discarding attack traffic quickly.
- Security: the scheme should be resilient, light-weight and robust against a number of attacks and not be a target to new threats.
- Efficiency: the proposed scheme should be efficient and not generate additional traffic thus increasing the load during periods of attack. The scheme should also improve the performance of the security service and minimize both computational and communication overhead.
- Authenticity: all relayed bundles must be authenticated to prevent the misuse of the DTN infrastructure.

We assume that security policies and cryptographic materials (such as keys and Initialization Vectors (IV)) have been securely distributed. We assume that an Offline Security Manager (OSM) exist that handles the generation and distribution of cryptographic credentials during the initialization phase of the system. Key revocation is out of scope of this work.

3. The Proposed DOS Mitigation Scheme

The Bundle Security Protocol (BSP) specification [34] provides minimal protection against DoS attacks. DTN nodes simply drop bundles that fail the authentication and access control checks. This in itself is vulnerable to new security threats such as resource exhaustion attacks. An attacker simply sends a large volume of bundles to a target node. The victim node will be kept busy verifying bogus signatures thereby wasting its resources (CPU processing cycles and battery). Legitimate bundles will be denied access to the victim node or dropped due to congestion or time-to-live expiry. The primary goal of any flood-based DoS mitigation mechanism is to restrict the volume of malicious traffic during an attack. Mitigating such attacks will consume resources at security-aware nodes or gateways and may require a number of filters to classify attack flows.

To guarantee the availability of connections to legitimate traffic during a flooding DoS attack we propose a comprehensive DoS-resilient scheme against flooding and resource exhaustion DoS attacks. The design of the proposed scheme combines rate limiting techniques, ingress filtering (for gateways) and light-weight bundle authentication. We propose three DTNCookie variants to provide light-weight authentication for the intra-region and inter-region DoS scenarios. Figure 3 shows a generic DTN bundle with additional security blocks such as the Bundle Authentication Block (BAB), Payload Integrity Block (PIB), Payload Confidentiality Block (PCB) and DTNCookie block. These security blocks are used to protect certain fields on the bundle to prevent modification attacks, replay attacks, eavesdropping and resource exhaustion DoS attacks.

Table 1 provides a description of bundle fields shown in Figure 2.

Table 1. Bundle fields and their meanings

Symbol	Description
TS	The bundle timestamp
LS	Life time of the bundle
SrcEID	Source End-point Identifier
DstEID	Destination End-point Identifier
CoS	Class-of-Service: normal, expedited, bulk
RSA-SHA256	Cipher suite for digital signature
NTL	Network Threat Level Indicator
BAB	Bundle Authentication Block
PIB	Payload Integrity Block
PCB	Payload Confidentiality Block
ESB	Extension Security Block
M	Message payload
H (M)	Hash of the message payload
privK(H(M)	Digital signature
Cookie	DTNCookie Block

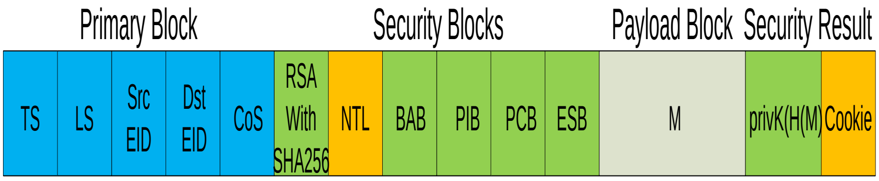

Figure 2. A generic bundle with security blocks

The proposed DTNCookie variants are as follows:

$$\text{DTNCookie1} = \text{H}\Big(\big((\text{TS} \mid \text{SrcEID})\big|\text{LS}\big|\text{CoS} \mid \text{NTL}\big) \mid \text{p} - \text{RNG}(\text{IV})\Big) \quad (1)$$

$$\text{DTNCookie2} = \text{H}\big(\big((\text{TS} \mid \text{SrcEID})\big|\text{LS}\big|\text{CoS} \mid \text{NTL}\big)\,\text{xorp} - \text{RNG}(\text{IV})\big) \quad (2)$$

$$\text{DTNCookie3} = \text{Hmac}\big(\big((\text{TS} \mid \text{SrcEID})\big|\text{LS}\big|\text{CoS} \mid \text{NTL}\big)\,\text{xor}\ \text{p} - \text{RNG}(\text{IV})\big),\ \text{K}_{\text{RS}}) \quad (3)$$

The proposed light-weight (DTNCookies) are derived from the fields which are specific to each bundle. DTNCookie1 is derived when an Initialization Vector IV known to only legitimate and registered nodes is used as seed to the pseudo-Random Number Generator (p-RNG). The resulting value is a big integer which is used as nonce. A concatenation of the source addresses (Src EID) and the timestamp enables our mechanism to uniquely identify each bundle. The nonce is further concatenated with the unique bundle identifier (Timestamp+Src EID) and the result hashed using SHA-256 represented here by H. The fixed length hash *h* becomes the light-weight DTNCookie which we append to every bundle.

DTNCookie2 is derived in the same way. The difference is the replacement of concatenation with exclusive-OR (Xor) operation. The Xor operation produces a bit flip which inputs more randomness into the DTNCookie generation process. In the same vein, DTNCookie3 variant is generated in the same way as the second. The difference is that the result of the operation is hashed with a regional secret key K_{RS} using SHA-256 to produce a fixed-length Message Authentication Code (MAC). The MAC becomes the DTNCookie which we append to every bundle. The use of a pseudo-random number generator (p-RNG), bitwise exclusive OR, the secrecy in the mode of key generation and its length makes the three DTNCookie variants random and secure. The DTNCookies are hard to forge due to the secrecy associated with the secret key and the initialization vector (IV). The IV or seed is changed periodically to ensure freshness and prevent compromise.

For the intra-region scenario shown in Figure 1 (a), any of DTNCookie1 or DTNCookie2 can be used as the light-weight bundle authenticator. Similarly, for the inter-region scenario shown in Figure 1 (b), DTNCookie3 is used as the lightweight bundle authenticator. DTNCookie3 though light-weight, is computationally more expensive than DTNCookie1 and DTNCookie2 since a gateway is assumed to be computationally more capable than a node within a region. Secondly, DTNCookie3 is derived based on the assumption that DTN gateways are loosely time-synchronized. Finally, the key fetch operation cost during the computation and verification of DTNCookie3 is negligible for DTN gateways compared to nodes within regions.

Our proposed scheme is based on the approach of analyzing the source addresses of bundles and other specific bundle fields (blocks) in order to correctly distinguish between legitimate and illegitimate traffic. Figure 3 shows a block diagram of a regional gateway with two interfaces *in* and *out*. Bundles that traverse the gateway pass through the ingress filter to test if the bundle originates from a trusted and legitimate gateway. The rate limiting filter counts the bundles per traffic flow to ensure that each flow does not exceed a set threshold. The rate limiter helps to dampen the effects of a flooding DoS attack. The light-weight authentication filter ensures that only legitimate bundles are allowed to use DTN resources. Bundles that do not meet the requirements set in the security policies are discarded and the node address logged to help the gateway make informed decisions in the future. In Figure 1, each security-aware node within a region has two filters (rate limiting filter and light-weight authentication filter) as opposed to three by a gateway.

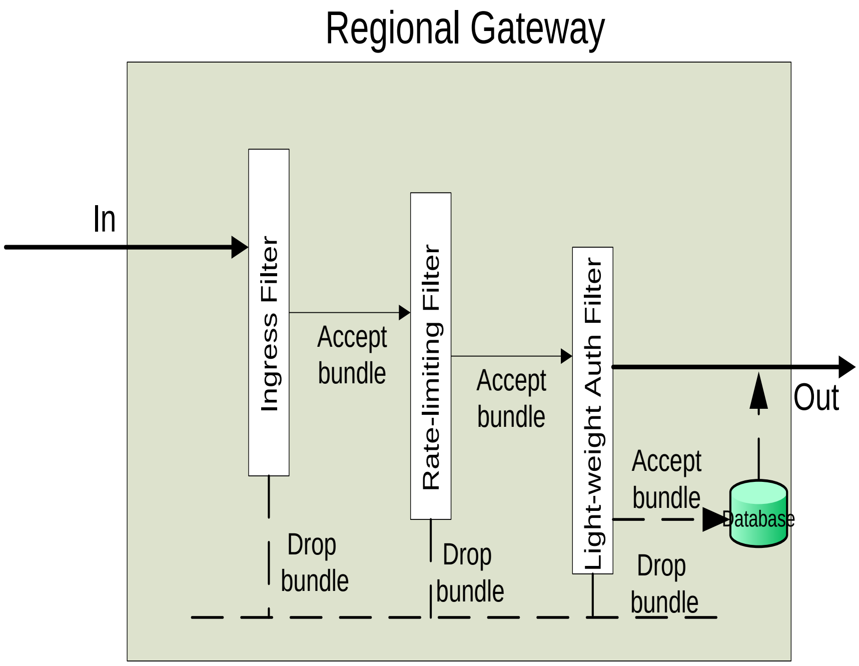

Figure 3. Gateway block diagram with DoS filters

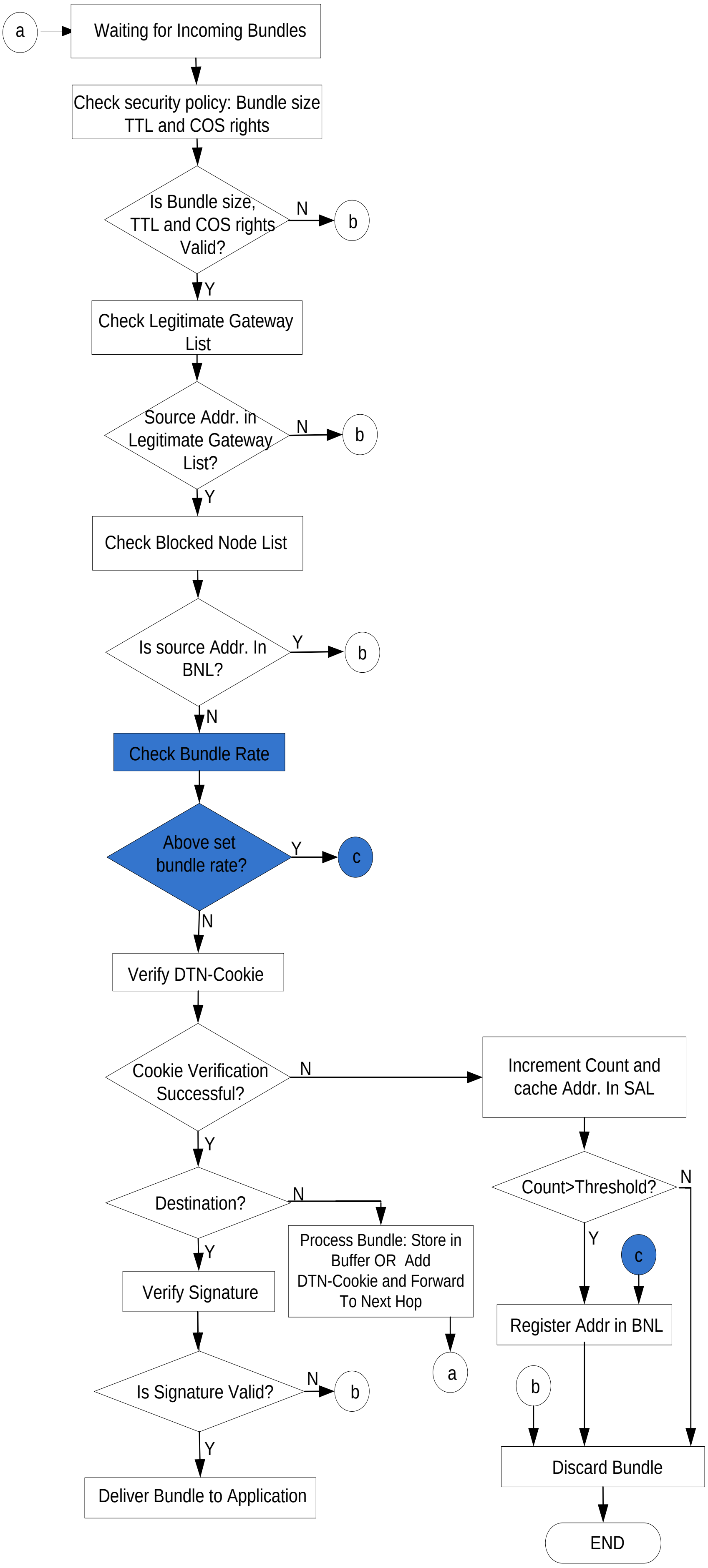
a
Waiting for Incoming Bundles
Check security policy: Bundle size TTL and COS rights
Is Bundle size, TTL and COS rights Valid?
N
b
Y
Check Legitimate Gateway List
Source Addr. in Legitimate Gateway List?
N
b
Y
Check Blocked Node List
Is source Addr. In BNL?
Y
b
N
Check Bundle Rate
Above set bundle rate?
Y
c
N
Verify DTN-Cookie
Cookie Verification Successful?
N
Increment Count and cache Addr. In SAL
Count>Threshold?
N
Y
c
Register Addr in BNL
Y
Destination?
N
Process Bundle: Store in Buffer OR Add DTN-Cookie and Forward To Next Hop
a
Y
Verify Signature
Is Signature Valid?
N
b
Y
Deliver Bundle to Application
b
Discard Bundle
END

Figure 4. Flood-based DoS attack mitigation for DTN gateways

Figure 4 shows the flow of processes and steps taken by a gateway to thwart DoS attacks mounted through flooding. We incorporate a rate limiting mechanism to help identify malicious nodes that send bundles at a rate they are not authorized to. Any node identified as exceeding its authorized bundle sending rate is penalized. When a bundle arrives at a gateway, the bundle size, time-to-live and Class-of-Service rights of the user are checked for validity. Bundles with sizes that do not meet the requirement specified in the security policy are discarded. Similarly, bundles with expired TTL and incorrect CoS are dropped. If the bundle size, TTL and CoS rights are valid, the gateway checks the source address on the bundle against the Legitimate Gateway List (LGL). Bundles whose source addresses are not in the LGL are considered spoofed and are discarded. If the source address matches a gateway address in LGL, that gateway address is checked against the Blocked Node List (BNL). The BNL contains addresses of blocked gateways that are considered spoofed because they failed the DTNCookie verification or exceeded the set bundle rate threshold.

If a source address matches an address in BNL, the bundle is discarded otherwise the bundle rate associated with the particular traffic flow is checked against a set bundle threshold. If a gateway exceeds the bundle rate threshold, its address is logged in the BNL and subsequent bundles originating from such blocked gateways are silently dropped until the configured block period expires. This helps to free up bandwidth for legitimate traffic and improve network performance. However, if the bundle rate of a particular traffic flow is within the bounds of the set threshold, the gateway proceeds to verify the DTNCookie. To verify the DTNCookie, the gateway uses the timestamp to choose the correct seed for generating the unique nonce for computing the DTNCookie. The computed DTNCookie must match the received DTNCookie. Any mismatch will mean that the bundle is spoofed and must be dropped. The source addresses of bundles that fail the DTNCookie verification are logged in the Spoofed Address List (SAL). Source addresses in the SAL that exceed a COUNT threshold are automatically logged in BNL. Bundles originating from such spoofed gateways addresses are silently dropped until the configured block period expires.

The gateway proceeds to verify the digital signature protecting the bundle payload provided the DTNCookie verification is successful and the gateway is the bundle destination. If the signature verification is successful, the bundle is delivered to the application. If the gateway is not the bundle destination, the bundle is either stored in the gateway’s buffer or a new DTNCookie is computed, appended to the bundle and then forwarded to the next hop. In the event where signature verification is unsuccessful, the bundle is discarded and processed no further.

Table 2 illustrates our use of variable seeds in the computation of different nonce values in different timeslots. The reason for this is to guarantee that the generated DTNCookies are random and hard to forge. Time-synchronization is an important aspect we considered during the design of the proposed scheme. We assume initial pre-shared symmetric keys between the gateways (Gateway1, Gateway 2 and Gateway 3). The security gateways have a common view of time [say Coordinated Universal Time (UTC)] irrespective of their time zones. Also, the p-RNG functions at the security gateways have a uniform initial seed value (N_o).

Communications among the gateways is initiated by the DTN owner or Administrator (say Gateway 2) by sending two different seed values to Gateway 1 and Gateway 3. The seed is the bundle payload and is encrypted using the public key of Gateway 1 and Gateway 3. Gateway 2 signs the bundles using its private key, calculates the DTNCookie, appends it to the bundles and sends to the gateways. At the gateways, the timestamp and sender EID are retrieved from the bundle and based on the pre-shared symmetric keys (K_S) between the Gateway 1, Gateway 2 and Gateway 3. The DTNCookie is computed and compared to that on the received bundle. The bundle is silently dropped if the DTNCookie verification is unsuccessful. On the other hand, if the verification is successful we proceed to test the integrity of the digital signature. Each gateway uses the public key of Gateway 2 to verify the signature. If the signature verification fails the bundle is dropped because its content is considered modified on transit.

If the signature verification is successful, we proceed to decrypt the payload and each gateway uses its private key to decrypt the payload which is the new reference seed for the generation of nonce values. Attackers within the data mules’ coverage are able to see every communication if the channel is a broadcast channel like in satellite communication. To prevent eavesdropping of the seed, we encrypt the payload. Bundles that arrive after their creation timeslot can still be processed if they are not expired. When a bundle arrives at a node, the bundle agent retrieves the bundle timestamp. The bundle agent uses the timestamp to determine the seed it should use for the generation of nonce for bundle verification.

Table 2. Generation and use of variable nonce values in different timeslots

Timeslot (seconds)	Nonce Variables	Description
0 - 7200	N_0	The nonce N_0 is a 256-bit random BIGInteger value derived when a reference seed S_0 is input into a pseudo Random Number Generator (pRNG).
7200 - 14400	N_1	The nonce N_1 is a 256-bit random BIGInteger value derived using the seed S_1 where $S_1 = S_0 +$ [counter] when seed S_1 is input into a pseudo Random Number Generator (pRNG).
14400 - 21000	N_2	The nonce N_2 is derived using the seed S_2 where $S_2 = S_1 +$ [counter] and derived the same way as $N_{1.}$
21000 - 28200	N_3	The nonce N_3 is derived using the seed S_3 where $S_3 = S_2 +$ [counter] and derived the same way as $N_{2.}$
28200 - 36000	N_4	The nonce N_4 is derived using the seed S_4 where $S_4 = S_3 +$ [counter] and derived the same way as $N_{3.}$
36000 - 43200	N_5	The nonce N_5 is derived using the seed S_5 where $S_5 = S_4 +$ [counter] and derived the same way as N_4.

3.1. Security Analysis of the Proposed Scheme

The proposed scheme requires a single bundle exchange to achieve bundle authentication. The scheme uses symmetric cryptography and hash functions which are four orders of magnitude faster than public-key cryptography and digital signatures. The computational requirements of hash functions and MACs are low compared to digital signatures. A unique feature of the DTNCookie is the concatenation of the timestamp and source_EID to produce a unique bundle identifier useful for thwarting replay attacks and preventing old or expired bundles from circulating the network. Any attempt to change the timestamp field will invalidate the bundle during the verification of the DTNCookie. Our proposed DTNCookie protects the primary block and security block fields against modification attacks. The integrity and confidentiality of the payload is protected using a digital signature and encryption respectively.

The use of a cryptographically secure random number generator and variable nonce values in different timeslots make the proposed DTNCookies random and hard to forge**.** The first and second DTNCookie variants use SHA-256 as hash function. SHA-256 is a 256-bit hash function which uses 32-bit words and provides 128 bits of security against collision attacks [24]. The hash operation produces a fixed-length DTNCookie which saves memory, CPU processing and provides integrity to bundle fields. As a requirement, H can be applied to a block of data of any size, and it is relatively easy to compute H(x) for any x. For any given value h it is computationally infeasible to find $y \neq x$ such that $H(y) = H(x)$ (weak collision resistance). It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$ (strong collision resistance) [35]. The first and second DTNCookie variants have all these properties in-built.

The third DTNCookie variant uses HMAC, a mechanism for message authentication and uses SHA-256 and a secret key. The cryptographic strength of HMAC is dependent on the properties of the underlying hash function and the bit length of the key. On average, an attack will require $2^{(k-1)}$ attempts on a k-bit key. The amount of effort needed for a brute-force attack on a MAC algorithm can be expressed as min $(2^k, 2^n)$. The key and MAC lengths should satisfy the relationship min $(k, n) \geq N$, where N is in the range of 128 bits [24]. The irreversibility property of the one-way hash function and the secrecy of the symmetric keys ($K_{S,}$ K_{RS}), makes the proposed DTNCookie hard to forge.

4. Performance Evaluation and Simulation Results

Providing security in DTN imposes both bandwidth utilization costs and computational cost on DTN nodes. The amount of bandwidth consumed and the amount of computation required depends on how parameters are encoded and the cryptographic algorithms in use [36]. In this section, we conduct the performance evaluation on the proposed DoS defence scheme.

4.1. Simulation Setup and Evaluation Metrics

We implement our comprehensive DoS mitigation scheme on the Opportunistic Network Environment (ONE) simulator [37] and evaluate its performance. In our simulations, 5 super nodes are uniformly deployed in a 4500 meters by 3400 meters area as gateways. These super nodes are defined as static gateways and are visited periodically by a mobile gateway which acts as data mule to provide connectivity to the 5 defined regions. The gateways have a transmission speed of 54Mbps and transmission range of 300 meters. The data mule travels at a speed of between 105-118km/hr and pauses when it arrives at a region to collect and deliver bundles for 0 to 5 seconds.

In the first scenario, we vary the number of attackers from 10 to 50 to see the effect of increased number of attackers on bundle delivery ratio, average latency, overhead ratio and energy consumption. An attacker generates one bundle of size 1 Megabytes every 5 seconds. In the second scenario, we vary the number of bundles generated by an attacker from 1 attack bundle per second to 1 attack bundle every 20 seconds. The purpose is to evaluate the effect of high bundle rate used in bandwidth DoS attacks on network performance. Some details of the simulation parameters are shown in Table 2. We implement our mechanism using the Spray and Wait routing protocol [38]. The simulation parameters and the network metrics used in the evaluation of the proposed DoS defence scheme as defined in the ONE simulator are shown in Table 3.

Table 3. Simulation Parameters

Simulation Duration	43200 s
Number of Gateways	6
Speed of Gateway	Stationary
Transmission Range	300 meters
Initial Energy	4×10^9 mJ
Message Size	1.5 Megabytes
Message TTL	300 minutes
Message Generation Interval	60 s
Routing Protocol	Spray and Wait
Number of Forwarding Copies	5 Copies
Buffer Size	50 Megabytes

$$DeliveryRatio = \frac{(1.0*\text{number of bundles Delivered})}{\text{number of bundles Created}}$$

$$Overhead = \frac{\left(1.0*\left(\begin{array}{l}\text{number of bundles Relayed}\\ -\text{number of bundles Delivered}\end{array}\right)\right)}{\text{number of bundles Delivered}}$$

$$Average\ Latency = \frac{\text{LatencyB1} + \text{LatencyB2} + \ldots + \text{LatencyBN}}{\text{number of bundles Delivered}}$$

Where B1, B2…BN represent bundles 1 to N.

4.2. Simulation Results

4.2.1. Scenario 1: Effect of Increasing Number of Attackers

We first test the robustness of the proposed DoS mitigation mechanism in the presence of increased number of attackers. We vary the number of attackers from 10 to 50 and examine how this affects network performance.

Figure 5 shows the effect when the number of attackers increases from 10 to 50. In a network of 10 attackers, when rate limiting is applied to incoming traffic flows, the delivery ratio for legitimate bundles is 90.69%. The delivery ratio for legitimate bundles declines to 68.61% when the number of attackers increases to 50. This result shows that rate-limiting as a flood mitigation technique performs poorly during high bandwidth DoS attacks which involve very high bundle rates. Combining rate limiting and RSA-1024 security achieves a delivery ratio of 31.53%. However when we activate our proposed flood mitigation mechanism, an average delivery ratio of 99.25% is achieved even when 50 attackers are present in the network.

Figure 6 shows the average latency experienced by legitimate bundles as attackers increase from 10 to 50. Any increase in the number of attackers directly equates to an increase in attack intensity and volume. Using rate limiting as the only mitigation mechanism, legitimate bundles experience an average latency of 198 seconds with 10 attackers and 214 seconds with 50 attackers respectively. Average latency increases as more attackers join the network because the rate limiting mechanism allows a percentage of attack bundles to pass through. This causes the network to become congested and legitimate bundles are dropped if buffers are full. Using RSA-1024 in combination with rate limiting as mitigation mechanism, the average latency per legitimate bundle is 104 seconds. The reason for the low average latency for RSA-1024 is because less than 32% of legitimate bundles are delivered as shown in Figure 5 and average latency is calculated based on number of bundles delivered. When our proposed mechanism is used, the average latency experienced by a legitimate bundle is 191 seconds as attackers increase from 10 to 50. This represents 1.06% of a bundle’s TLL.

Overhead ratio is an assessment of bandwidth efficiency. Figure 7 shows the Overhead ratio which is dependent on the number of relayed and delivered bundles. The overhead ratio when rate limiting is used as the sole mechanism to mitigate flood-based DoS attacks is 106.16 on average because a large number of bundles are relayed but not delivered as a result of network congestion. RSA -1024 with rate limiting has an overhead ratio of 10.04 because less than 32% of the total relayed bundles are delivered. Our proposed mechanism has the lowest overhead ratio of 3.19 which is largely due to the high delivery ratio recorded by our mechanism as shown in Figure 5.

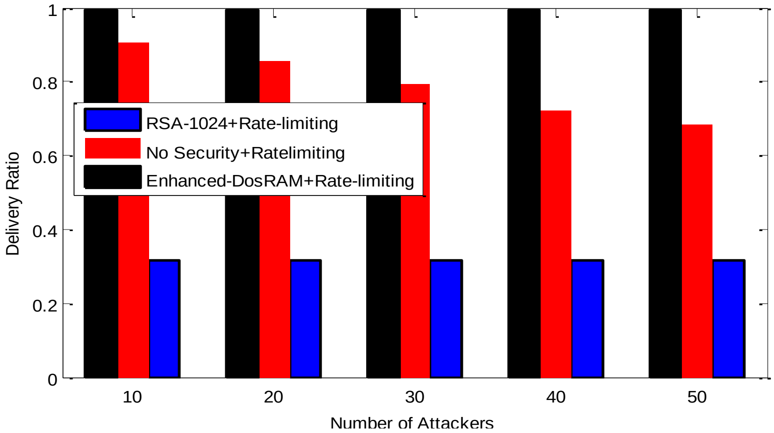

Figure 5. Delivery ratio with increased number of attackers

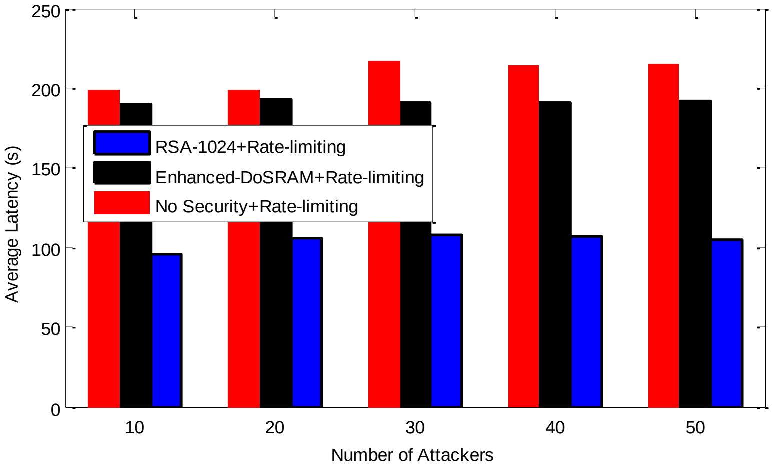

Figure 6. Average latency with increased number of attackers

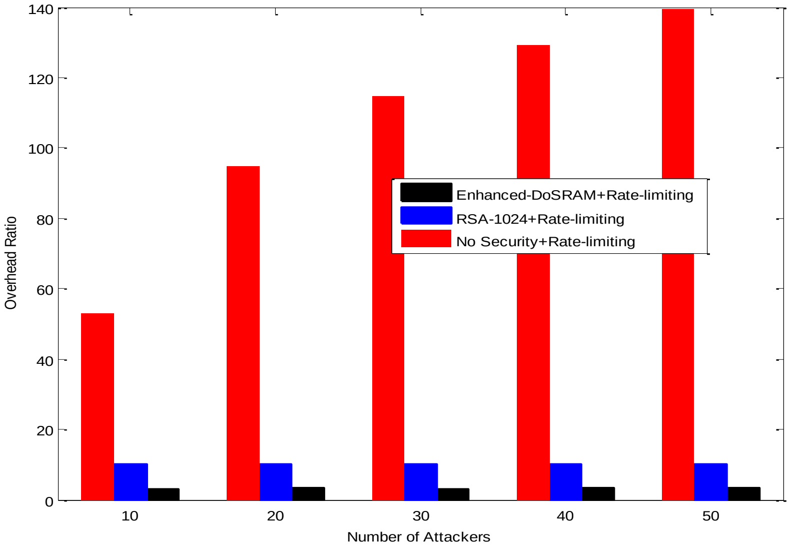

Figure 7. Overhead ratio with increased number of attackers

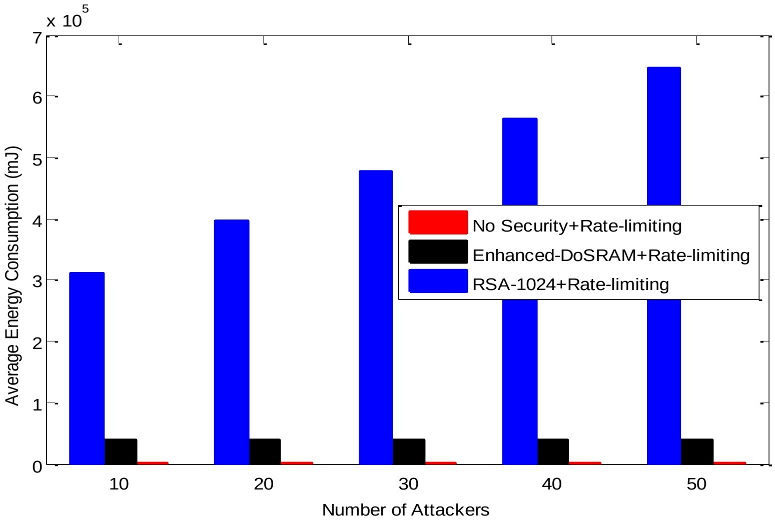

Figure 8. Energy consumption with increased number of attackers

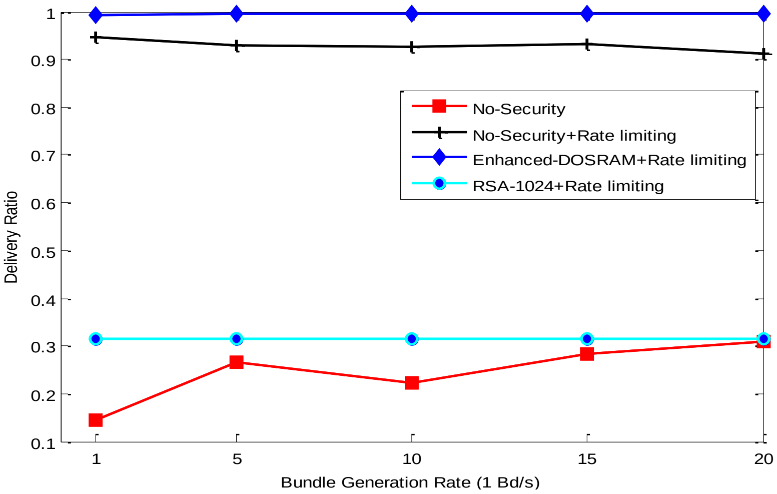

Figure 9. Impact of network load on delivery ratio

Figure 8 shows that the average energy consumption is 1238 mJ when rate limiting is used as the only security mechanism against flooding DoS attacks. This represents 0.00003% of a gateway's total energy reserve. Combining RSA-1024 with rate limiting consumes 312078 mJ of energy per gateway when 10 attackers are present in the network. Energy consumption continues to rise with an increase in the number of attackers. A gateway expends 645674 mJ of power with 50 attackers in the network. However, when our proposed flood mitigation mechanism is activated, the average energy consumed per gateway is 39465 mJ which represents 0.001% of a gateway's energy reserve. The energy consumption of our proposed mechanism is less than the energy consumption when RSA-1024 digital signature is used. This is because the energy required for ingress filtering, table lookups, list searches and DTN-Cookie computation and verification is negligible.

4.2.1. Scenario 2: Impact of Traffic Load on Network Performance

The intensity of a DoS attack is measured based on the percentage of legitimate bundles lost due to excessive traffic load. An attack intensity of 70% means that only 30% of legitimate traffic will be delivered. We set the bundle rate threshold for each gateway to 600. In this scenario, we are interested in measuring the impact of different bundle sending frequencies on the overall network performance. We adjust the bundle generation rate for the attacker from 1 bundle per second to 1 bundle every 20 seconds. Again we measure the network performance using four metrics: delivery ratio, average latency, overhead ratio and average energy consumption.

Figure 9 shows the effect of different bundle generation rates on delivery ratio. High bundle generation rates imply increased number of messages which in turn result in lower delivery ratios. This is evident when no form of security mechanism is adopted and the bundle generation rate is high. The delivery ratio for legitimate bundles when an attacker generates 1 bundle per second is 14.44%. As the attack intensity reduces to 1 bundle every 5 seconds, the delivery ratio for legitimate bundles rises to 26.53%. Delivery ratio for legitimate bundle drops to 22.22% when the attacker generates 1 bundle every 10 seconds. Although not shown in Figure 9, simulation results show a 99.79% delivery ratio for attack bundles. From the graph, it is clear that delivery ratio for legitimate bundles start rising gradually as the bundle generation rate becomes less frequent. When incoming bundles are rate limited, delivery ratio rises to an average of 93% because any attacker that exceeds the set bundle generation threshold of 600 bundles per flow is blocked. Using RSA-1024 with rate limiting as a DoS defence mechanism against flooding achieves a delivery ratio of 31.53%. However when we switch ON our proposed mechanism, a delivery ratio of 99.31% is achieved with 1 attack bundle generated per second. The delivery ratio rises to 99.44% as the attack intensity eases with 1 attack bundle generated every 5, 10 or 20 seconds.

Latency is influenced by many factors such as congestion, node processing, and queuing delays. Multi-hop routing can also introduce delay as bundles move from source to destination. Figure 10 shows that when no security mechanism is adopted to mitigate flooding, average latency rises as the intensity of the bundle generation rate reduces from 1 bundle per second to 1 bundle every 20 seconds. Average latency is calculated as the cumulative sum of latencies experienced by legitimate bundles delivered divided by the number of delivered bundles. This supports the result shown in Figure 9 which shows delivery ratio rising with a reduction in attack intensity. When we rate limit the incoming traffic, average latency is 194 seconds because a large percentage of legitimate bundles are delivered to destination. When RSA-1024 and rate limiting are used as mitigation mechanism, the average latency is 99.6 seconds when 1 attack bundle is generated per second, and 105 seconds on average when 1 attack bundle is generated every 5, 10, 15 or 20 seconds. The reason for the low average latency is because less than 32% of legitimate bundles are delivered as shown in Figure 9. When we apply our proposed mechanism, the average latency is 190 seconds which is equivalent to 1.06% of a bundle's TLL.

Bandwidth is one of the scarce resources in DTN which we attempt to protect using our proposed scheme. Overhead ratio is an assessment of bandwidth efficiency and an important metric used in the evaluation of our proposed scheme. In Figure 11 with no security mechanism activated, overhead ratio when 1 attack bundle is generated per second is 14.64 and rises sharply with 1 attack bundle generated every 5, 10, 15 or 20 seconds. This is because a large number of both legitimate and attack bundles are relayed but few legitimate bundles are delivered to destination due to congestion. When the rate limiting mechanism is activated, the overhead ratio remains high but less than with no security mechanism in place. Using RSA-1024 with rate-limiting, the overhead ratio is 10.04 on average and is fairly constant throughout the duration of the simulation because illegitimate bundles are dropped at the first hop. However since RSA digital signatures are used, the gateways are computationally occupied with the verification of bogus signatures from the attacker and legitimate bundles are dropped if their TTL expire. It is clear that our proposed flood mitigation mechanism is more bandwidth efficient since it has the lowest average overhead ratio of 3.18. This is evident in the high performance of our proposed scheme in terms of delivery ratio as shown in Figure 9. Despite the variation in bundle generation rate, overhead ratio remains fairly constant when our mechanism is activated. The proposed DTN-Cookies is based on hop-by-hop authentication where an attack bundle is dropped close to the source of attack. This is in contrast to end-to-end authentication which allows an attack bundle to traverse the network wasting scarce network resources before it is dropped at the destination.

Energy consumption is an important consideration in DTN security design because most of the nodes are low-power devices. During bundle authentication, both computational and communication energy (transmit and receive) are consumed. Figure 12 shows that the difference in energy consumption when no form of security is adopted and when rate limiting is used as a flood mitigating mechanism is quite negligible (1245 mJ and 1240 mJ respectively). This is because no cryptographic operations are involved and the amount of energy required for table lookups is low. When RSA-1024 is used in combination with rate limiting, the energy consumption per gateway rises to 270468 mJ which represents 0.0068% of a gateway's total battery power. Our proposed mechanism incurs an energy cost of 39464.4 mJ per gateway when activated which is equivalent to 0.00099% of a gateway's total battery power which is less when compared to the energy consumption of RSA-1024 digital signatures. The graphs for No Security mechanism and when Rate limiting is used as a security mechanism appear to overlap in Figure 12.

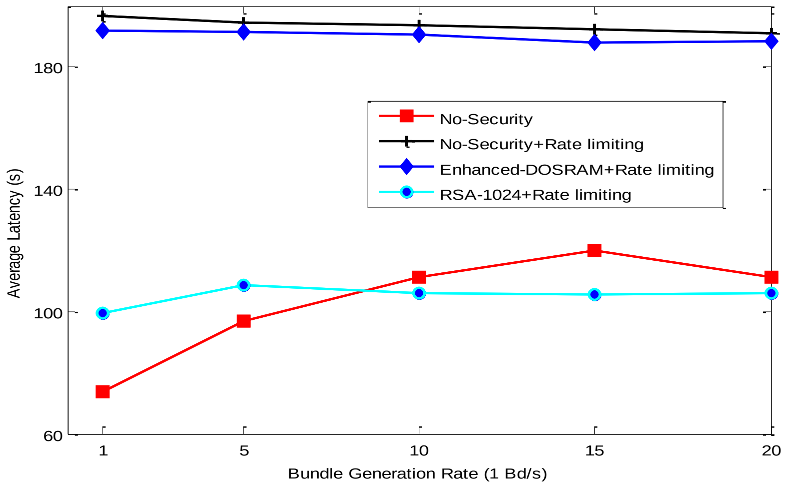

Figure 10. Impact of network load on average latency

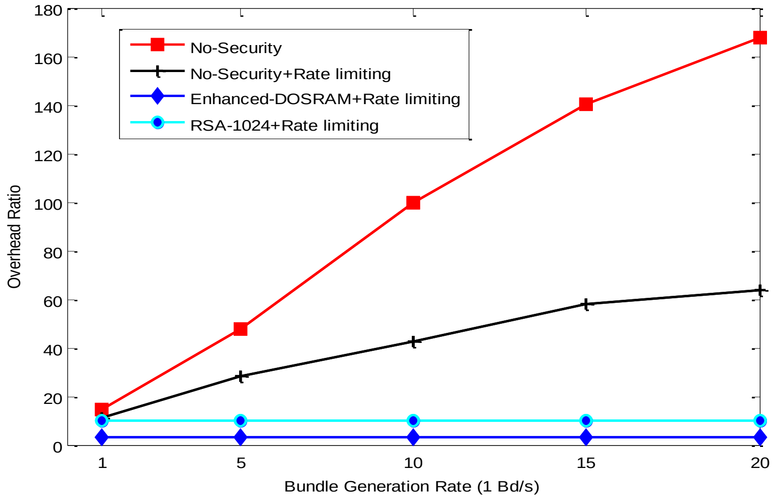

Figure 11. Impact of network load on overhead ratio

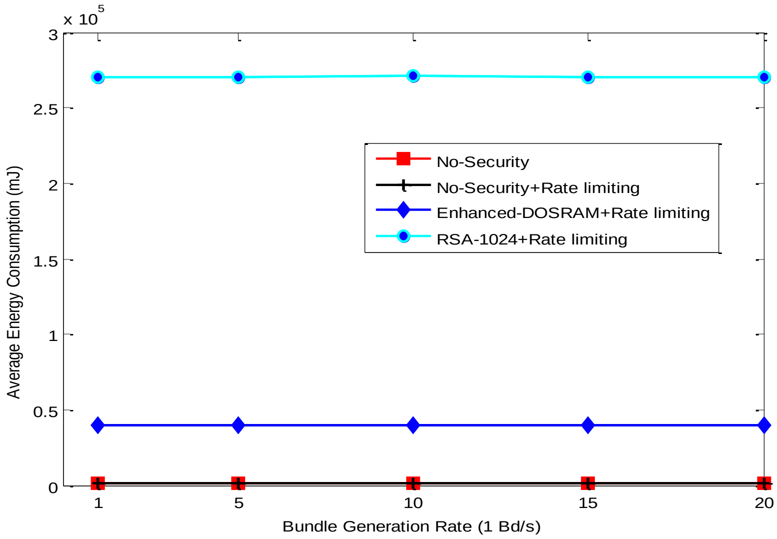

Figure 12. Impact of network load on energy consumption

5. Conclusion

DoS attacks are a threat to network availability in DTN. The use of strong and complex security algorithms against DoS attacks exposes DTN nodes to a new threat called resource exhaustion attack. This type of attack depletes the scarce resources of the network and degrades performance. In this paper, an attack model which uses high traffic volume through flooding was described. We have proposed a scheme which provides DoS resilience against resource exhaustion and flood-based DoS attacks in the inter-region scenario. The proposed scheme is a light-weight bundle authentication mechanism which uses DTN-Cookies, ingress filtering and rate limiting techniques to detect and drop attack traffic.

The proposed scheme is dynamic because each gateway runs the algorithm and independently decides when to trigger the rate limiting filter. Different nodes are assigned different threshold values based on their capability and role in the network. We adopted the use of standard spray-and-wait routing in our simulations to boost performance while controlling the amount of traffic. The aim of this scheme is two-fold: to identify and stop authorized nodes (with cryptographic credentials) and unauthorized nodes from sending bundles at a rate which they are not authorized to.

Secondly, the proposed scheme protects the scarce resources of a DTN which include battery power, memory, bandwidth and CPU processing cycle against wasteful depletion. The results show that the proposed DTN-Cookies accurately detect DoS traffic and outperform RSA-1024 digital signatures in terms of energy and bandwidth efficiency. The results from the two case scenarios described in this paper show significant improvements in delivery ratio, average latency, overhead ratio and energy consumption. We have also shown through the results, that the proposed DoS defence scheme is scalable as the number of attackers and attack bundle rate increases

References

[1] Ansa, G., Johnson, E., Cruickshank, H., and Sun, Z, "Mitigating Denial of Service Attacks in Delay-an-Disruption Tolerant Networks," *in Personal Satellite Services Conference*, vol. 43, Rome, 2010, pp. 221-234.

[2] Khabbaz, M. J., Assi, C. M. and Fawaz, W. F, "Disruption Tolerant Networking: A Comprehensive Survey of Recent Developments and Challenges," *IEEE Communications Surveys and Tutorials*, 2011.

[3] Caini, C., Cruickshank, H., Farrel, S. and Marchese, M. "Delay- and Disruption-Tolerant-Networking (DTN): An Alternative Solution for Future Satellite Networking Applications," *IEEE Proceedings*, vol.99, no. 11, pp. 1980-1987, 2011.

[4] Jonson, T., Pezeshki, J., Choa, V., Smith, K.. and Fazio, J. "Application of Delay Tolerant Networking (DTN) in Airborne Networks," in *Military Communications Conference*, San Diego, California, USA, 2008.

[5] Ehsan, S. et al., "Design and Analysis of Delay-Tolerant Sensor Networks for Monitoring and Tracking Free-Roaming Animals," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 1220-1227, March 2012.

[6] Pereira, P. et al., "From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1-17, September 2011.

[7] Small, T. and Haas, Z.J, "The Shared Wireless Infostation Model: A New Ad hoc Networking Paradigm (Or Where There is a Whale, There is a Way)," in *ACM MobiHoc'03*, Annapolis, Maryland, USA, 2003.

[8] Hui, P. et al., "Pocket-Stiched Networks and Human Mobility in Conference Environments," in *ACM SIGCOMM Workshop on Delay Tolerant Networking*, Philadelphia, Pennsylvania, USA, 2005.

[9] Zhang, Z.,"Routing in Intermittently Connected Mobile Ad hoc Networks and Delay Tolerant Networks: Overview and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 1, pp. 24-37, 2006.

[10] Fall, K., "A Delay-Tolerant Network Architecture for Challenged Internets," in *ACM SIGCOMM Conference on* Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003.

[11] Wood, L., Eddy, W., Ivancic, W., McKim, J., and Jackson, C, "Saratoga: A Delay-Tolerant Networking Convergence Layer for Efficient Link Utilization," in *Satellite and Space Communications*, Oston Maryland, USA, 2007.

[12] Wood, L. and Holliday, P, "Using Http for Delivery in Delay/Disruption Tolerant Networks," Network Working Group, *draft-wood-dtnrg-http-dtn-delivery-07*, USA, 2011.

[13] De Rango, F., Tropea, M,. Laratta, G., and Marano, S., "Hop-by-hop Local Flow Control Over InterPlanetary Networks Based on DTN Architecture," *in IEEE International Conference on Communications*, Glasgow, Scotland, 2008.

[14] Dierks, T. and Rescorla, E., "The Transport Layer Security Protocol," Version 1.2, *RFC 5246*, Network Working Group, 2008.

[15] Kent, S. and Seo, K., "Security Architecture for the Internet Protocol" RFC 4301, Network Working Group, 2005.

[16] Dwork, C. and Noar, M., "*Pricing via Processing or Combating Junk Mails"* Springer, Heidelberg, 1998.

[17] Juels, A. and Brainard, J., "Client Puzzles: A Cryptographic Countermeasure against Connection Depletion Attacks," *in Proc. Network and Distributed Systems Security Symposium*, pp. 151-165, 1999.

[18] Maughan, G., Schnertler, M., Schneider, M., and Turner, J., "Internet Security Association Key Management Protocol (ISAKMP)," *RFC 2408*, 1998.

[19] Onen, M. and Molva, R., "Denial of Service Prevention in Satellite Networks," *IEEE International Conference on Communications*, vol. 7, pp. 4387-4391, 2004.

[20] Meadows, C.,"A Formal Framework and Evaluation Method for Network Denial of Service," *in Proc. IEEE Computer Security Foundations Workshop*, 1999.

[21] Symington, S., Farrell, S., Weiss, H., and Lovell, P., "Bundle Security Overview," *Internet Draft*, 2008.

[22] Burgess, J., Bissias, G., Corner, M., and Levine, B., "Surviving Attacks on Disruption-Tolerant Networks without Authentication," *in Proc. ACM MOBIHOC*, 2007.

[23] Li, F., Wu, J., and Srinivasan, A., "Thwarting Blackhole Attacks in Disruption-Tolerant Networks using Encounter Tickets," *in Proc. IEEE INFOCOM*, 2009.

[24] Ren, Y., Chuah, M., Yang, J., and Chen, Y., "Detecting Wormhole Attacks in Delay-Tolerant Networks," *in Proc. IEEE Wireless Communications*, 2010.

[25] Lee, F., Goh, W., and Yeo, C., "A Queuing Mechanism to Alleviate Flooding Attacks in Probabilistic Delay Tolerant Networks," *in Proc. AICT*, 2010.

[26] Choo, F., Chan, M., and Chang, E., "Robustness of DTN against Routing Attacks," *in Proc. COMSNETS*, 2010.

[27] Asokan, N., Kostiainen, K., Ginzboorg, P., Ott, J., and Luo, C.,"Applicability of Identity-based Cryptography for Disruption-Tolerant Networking," *in Proc. ACM MOBIOPP*, 2007.

[28] Seth, A. and Keshav, S. "Practical Security for Disconnected Nodes," *in Proc. of IEEE ICIP Workshop on Secure Network Protocols*, 2008.

[29] Kate, A., Zaverucha, G.M., and Hengartner, U., "Anonymity and Security in Delay Tolerant Networks," *3rd International Conference on Security and Privacy in Communications Networks*, pp. 504-513, 2007.

[30] Householder, A., Manion, A., Pesante, L., Weaver, G.M, "Managing the Threat of Denial of Service Attacks," Version 10.0, CERT Coordination Centre, Carnegie Mellon University, Oct., 2001.

[31] Demmer, M. et al., "Implementing Delay Tolerant Networking," Intel Corporation Berkeley, Technical Report TRB-TR-04-020, 2004.

[32] Jea, D., Somasundra, A.A., and Srivastava, M.B, "Multiple Controlled Mobile Elements (Data mules) for Data Collection in Sensor Networks," *in 7th IEEE Conference on Distributed Computing in Sensor Systems*, pp. 244-257, 2005.

[33] Shah, R.C., Roy, S., Jain, S., and Brunette, W., *"Data mules: Modelling and Analysis of Three-Tier Architecture for Sparse Sensor Networks,"* Elsevier Ad Hoc Networks Journal, vol. 1, pp. 215-233, Sept., 2003.

[34] Symington, S., Farrell, S., Weiss, H., and Lovell, P., "Bundle Security Protocol Specification," *Draft-irft-dtnrg-bundle- security-17*, 2010.

[35] Krawczyk, H., Bellare, M., and Canetti, R., "HMAC: Keyed Hashing for Message Authentication," *in Crypto 1996*, 1996, pp. 1-15.

[36] Bindra, H.S. and Sangal, A.L, *"Considerations and Open Issues in Delay Tolerant Networks (DTNs) Security,"* Wireless Sensor Network Scientific Research Journal, pp. 635–648, Aug., 2010.

[37] Keränen, A., "Opportunistic Network Environment Simulator," Helsinki University of Technology, Department of Communications and Networking, Finland, Special Assignment Report, 2008.

[38] Spyropoulos, T., Psounis, K., and Raghavendra, C.S, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," *in ACM SIGCOMM Workshop on Delay Tolerant Networking*, New York, USA, 2005.