Combating Intra-Region DoS Attacks in Delay Tolerant Networks using Energy-Efficient Mechanisms

Godwin Ansa[1,*], Haitham Cruickshank[2], Zhili Sun[2] and Feda Alshahwan[3]

[1]Department of Computer Science, Akwa Ibom State University,
Ikot Akpaden, Mkpat Enin, Akwa Ibom State, Nigeria

[2]Institute for Communication Systems (ICS), Faculty of Engineering and Physical Science,
University of Surrey, Guildford, GU2 7XH, United Kingdom

[3]Public Authority for Applied Education and Training, College of Technological Studies,
Electronics Engineering Department, Computer Section, Kuwait University, Kuwait

[*]*Corresponding author's email: godwinansa {AT] aksu.edu.ng*

ABSTRACT— ***Denial of Service (DoS) attacks have been a major threat in the Internet and in other emerging networks including Delay Tolerant Networks (DTNs). A DTN is characterized by limited bandwidth, long queuing delays, low data rate, low power and intermittent connectivity. Most of the proposed DoS mitigation schemes for wired and wireless networks are highly interactive requiring several protocol rounds. They are also resource consuming, complex and assume intermittent connectivity. These features make the applicability of proposed schemes unsuitable in a DTN scenario. An attacker can exploit the DTN message forwarding mechanism to inject fake bundles into the network. The attacker's overall objective is to deplete node and link resources such as CPU processing cycles, battery power, memory and bandwidth. In this paper, we propose a proactive DoS-Resilient Authentication Mechanism (DoSRAM). The proposed mechanism uses three message authenticator variants called DTN-Cookies to minimize computational and communication costs. The proposed mechanism has been verified through simulations using the Opportunistic Network Environment (ONE) simulator. Results show that DoSRAM outperforms solutions which are based on RSA-Digital Signatures in terms of throughput, energy and bandwidth efficiency. DoSRAM can accurately detect and filter out DoS traffic.***

Keywords— Security, Resource exhaustion, DTN-Cookie, Denial of Service Attack.

1. INTRODUCTION

Many systems especially most mobile, sensor and hand-held devices are constrained by the need to conserve power. Most of these systems are totally destroyed due to a complete drain in their battery power. Such systems will be unable to provide service if a large proportion of nodes suffer from total battery depletion which can lead to network partitions. The cost of recovery can be very significant [1]. In these systems and in many applications, power is a critical and limited resource and its conservation is a priority [2]. Providing security in a DTN may impose additional costs in terms of bandwidth utilization and computational costs on the nodes. In Internet Protocol Security (IPSec) [3], bogus traffic injected into the network is carried all the way to the security destination and this consumes a lot of resources [4], [5].

Attackers are motivated to launch a resource-exhaustion type of DOS attack on a DTN because most DTN nodes are power-constrained. In a multi-hop DTN routing scenario, bundles are checked for authenticity and integrity at intermediate nodes. An attacker can exploit the message forwarding mechanism and the authentication/access control checks by injecting bogus bundles into the network. The aim is to deplete the scarce resources of the DTN which include battery power, bandwidth, CPU cycles, communication contact time and memory. Without authentication, unsuspecting honest nodes will replicate and forward fake bundles from the attacker. If allowed to go unchecked, the network will become congested and network resources and services will be denied to legitimate users. Extra traffic from rogue nodes may pose serious threats to the operation and survivability of the DTN due to its resource constraints. In DTN, unauthorized access and use of resources is viewed seriously and is highly discouraged.

Most access control methods use strong security and resource intensive authentication. Strong security implies a more secure system but degrades the performance of devices with limited resources and introduces new threats such as resource exhaustion. Public Key Cryptography (PKC) involves computationally-expensive operations such as modular exponentiation. Verifying digital signatures which are based on PKC hop-by-hop at intermediate nodes takes time and consumes resources. This leads to an increase in bundle propagation time and network congestion. The situation becomes worse when multi-copy routing/forwarding is adopted to boast bundle delivery ratio. The per-bundle addition in terms of size of the digital signatures and their corresponding certificates is large. For instance 1024-bit RSA is in the order of hundreds of bytes (128 bytes) in extra transmission cost per packet per hop. In this paper, we will focus on how to make security services like authentication efficient and robust against DoS attacks.

The Delay Tolerant Network Research Group (DTNRG) has adopted the Bundle Security Protocol (BSP) [6] specification to address two important security challenges in DTN communications. The absence of authenticity of transmitted bundles and a lack of authorization for nodes to access and utilize DTN resources. The BSP specification uses the Bundle Authentication Block (BAB) or the Payload Integrity Block (PIB) to achieve bundle authentication and node authorization by adding a digital signature to each bundle. The security source node signs the bundle with its private key to produce a bundle-specific digital signature. Intermediate nodes and the destination node can verify the authenticity of the sender, the integrity of the message and the Class of Service (CoS) rights of the sender using their public keys [6].

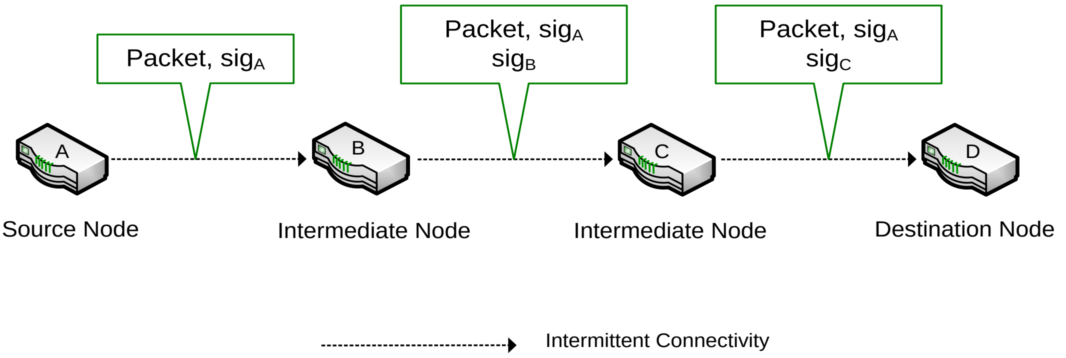

Figure 1: DTN Hop-by-Hop Authentication

A generic DTN bundle is made up of the primary block and the payload block as shown in Figure 3. The BSP specification [6] provides additional blocks which can be added to a bundle to provide a security service. For more details on the BAB, PIB, PCB and ESB security blocks, see section 3.3.1 and [6]. The BAB and the PIB are security blocks related to bundle authentication. The BAB provides protection for the bundle on a hop-by-hop basis, while the PIB protects the bundle end-to-end. In Figure 1, a source node A signs a bundle (packet) and appends its signature sig_A (PIB) to the bundle and sends it to an intermediate node B. Node B checks the sender's identity and Class of Service (CoS) rights for authenticity by verifying the signature. Node B generates its own signature sig_B (BAB) and appends it to the bundle and forwards to intermediate node C. Node C verifies Node B's signature, sig_B and replaces it with its own signature. This continues until the bundle arrives at the destination.

Implementing strong and resource-intensive security like digital signatures means more than having a very secure system or network. For resource-constrained DTN environments, such strong security will lead to a decline in device performance and resource exhaustion. Our objective is to achieve great efficiency during bundle authentication, by minimizing the computational and communication cost associated with security by using very light-weight mechanisms. In this paper, we propose a proactive DoS-Resilient Authentication Mechanism (DoSRAM). The scheme uses three variants of light-weight DTN-Cookies to detect and react quickly to illegitimate traffic that can lead to resource exhaustion DoS attacks. Our aim is to minimize the impact of security processing on energy constrained DTN nodes, keep computational and communication costs low, reduce latency, improve packet delivery ratio and prolong the life of the network.

2. SYSTEM/ATTACKER MODEL AND DESIGN ASSUMPTIONS

In this section we describe the system model, the attacker model, design assumptions, design goals and the networking and security requirements.

2.1 System Model

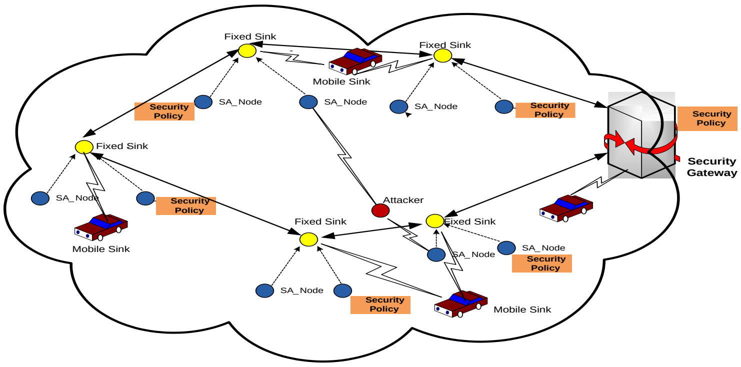

Figure 2: A DTN Region

2.2 Attacker Model

We assume the attacker can inject bogus bundles into the network to keep legitimate DTN nodes busy. This will drain their limited resources and introduce network latency. We assume that the computational power of the attacker is large and that the attacker can compute crypto-functions and Message Authentication Codes (MACs) with great speed and efficiency. The attacker can modify bundles but does not have knowledge of network secrets such as keys and nonce values. The attacker can or cannot spoof source addresses of legitimate nodes. We also assume that the attacker is mobile and is not capable of compromising legitimate nodes.

2.3 Design Assumptions

We assume that an Offline Security Manager (OSM) exist during the initialization of the system to handle key generation, distribution of secret credentials and security policies. We assume that a large number of nodes in the DTN are resource-constrained. We assume that gateways are stationary while malicious nodes are mobile. Key revocation is out of scope of this work. The security goals of the proposed DoS defence mechanism are as follows:

- Prevent otherwise authorized nodes/applications from sending bundles at a class-of-service which they lack permission
- Discard promptly bundles that are damaged or modified in transit
- The proposed DoS resilience mechanism should not be a target of new attacks
- Ensure that all relayed bundles are authenticated in order to filter and drop bogus traffic injected by the attacker i.e. promptly detect and discard unauthorized traffic
- The security mechanism should not increase network load during attack periods by generating additional traffic
- The proposed scheme should improve the performance of security service and network efficiency, improve bundle delivery ratio and minimize computational and communication costs in resource-constrained DTNs

2.4 Networking and Security Requirements

In order to counter the attacks listed in section 2.2, our design has to fulfil a number of networking and security requirements:

2.4.1 Networking Requirements

- It is important that the DoS resilience mechanism should be able to withstand significant node mobility
- The mechanism should be able to run efficiently on resource-limited nodes and be resilient to delays in the order of minutes, hours or days
- The mechanism should support varying data rates and withstand changes in contact times.
- In the absence of an end-to-end path between source and destination, the mechanism should be able to operate efficiently

2.4.2 Security Requirements

- Restrict security processing to capable and security-aware nodes
- To ensure freshness, we use nonce and timestamps to thwart replay attacks and drop expired bundles
- Every bundle is checked for integrity to prevent bundle content modification during transit
- Every bundle is authenticated to ensure they originate from legitimate sources.

3. THE PROPOSED SECURITY SCHEME

Figure 3 shows a generic DTN bundle with additional security blocks (BAB, PIB and PCB) added to provide security to the bundle. To provide DoS resilience, we propose a new security extension block called DTN-Cookie block. This block introduces light-weight message authentication mechanism which reduces the computational overhead associated with digital signature verification.

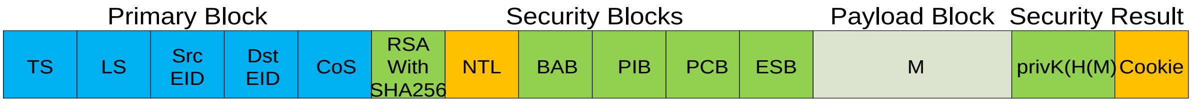

Figure 3: A DTN Bundle with Appended DTN-Cookie Block

We provide a definition of each bundle field shown in Figure 3 as follows:

- **TS:** the timestamp is a concatenation of a bundle creation time and a monotonically increasing sequence number. The TS is unique for every new bundle for source End-point Identifier (EID).
- **LS:** the bundle life span or expiry time of a bundle which can be in minutes, hours or days. In this work, LS and TTL (Time-To-Live) are used interchangeably.
- **Src_EID:** the Source End-point Identifier or Source address is the originator of the bundle, we assume that each EID is a singleton.
- **Dst_EID:** the destination EID is the entity or node for which the bundle is destined
- **CoS:** the sender's Class of Service rights is used to assign priority to certain class of traffic. A bundle can have an expedited, normal or bulk CoS rights
- **RSAwithSHA256:** represents the cipher suite for the digital signature algorithm
- **NTL:** represents the Network Threat Level indicator which a node uses to determine which cipher suite to use during DTN-Cookie verification.
- **BAB:** the Bundle Authentication Block, assures the authenticity of the bundle in a hop-by-hop basis
- **PIB:** the Payload Integrity Block, assures the authenticity and integrity of the bundle in an end-to-end basis
- **PCB:** the Payload Confidentiality Block, indicates that some parts of the bundle has been encrypted at the source
- **ESB:** Extension Security Block, indicates that there are additional blocks after the payload block to provide security to the bundle
- **M:** the bundle payload
- **H (M):** h is the hash value derived by passing the bundle payload M through the hash function H. H is a cryptographic hash function such as MD5, SHA1 or SHA256. We will be using SHA256 as the underlying hash function to the signature and MAC algorithms
- **privK(H(M)):** the bundle-specific digital signature
- **Cookie:** the DTN-Cookie, is derived from a combination of bundle fields and a 256-bit long random nonce as input to a one-way hash function.
- **pubKxi:** the public key of node X_i
- **privKxi:** the private key of node X_i

The BSP specification [7] 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 and 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.

3.1 DTN-Cookie Design

To prevent the attack described section 3, we propose three DTN-Cookie variants for the intra-region scenario which can be applied based on the perceived Network Threat Level (NTL). One critical feature of our design is the requirement to reduce the computational cost associated with bundle authentication. We take advantage of the unique blocks (fields) of the DTN bundle in the composition of our light-weight DTN-Cookie mechanism. The present BSP specification provides HMAC-SHA1 and RSA-SHA-256 digital signature as BAB for bundle authentication. HMAC-SHA1 and RSA-SHA-256 use symmetric and asymmetric cryptography respectively. Our proposed DTN-Cookies are based on cryptographic one-way hash functions with no need for cryptographic keys with the exception of variant 3. Fetching the symmetric key or the private/public key pairs from the key store during the computation and verification of the BSP BAB imposes energy consumption cost on security-aware nodes. The design of the first and second DTN-Cookie variants eliminates the key fetch operation. The DTN-Cookie is designed to provide resilience against modification and resource exhaustion attacks. It protects the integrity of the primary block fields and security blocks associated with the DoS-resilience mechanism. The payload block is protected against modification and eavesdropping attacks using PIB (digital signatures) and PCB (data encryption) in an end-to-end fashion.

$$DTN\text{–}Cookie = H((\langle TS|Src_{EID}\rangle|LS|CoS|NTL)|p\text{–}RNG(IV)) \qquad (1)$$

The Initialization Vector *(IV)* is known to registered nodes of the region. The IV is used to seed the pseudo-random number (p-RNG) generator. The output from the p-RNG is a 256-bit random long integer value of type BigInteger which is optimized for speed. This BigInteger value is the random nonce. A concatenation of the timestamp (*TS)* and the bundle source address (*Src_EID*) provides a unique bundle identifier. The unique bundle identifier is concatenated with the bundle life span (*LS* i.e. *TTL*: time-to-live), Class of Service (*CoS*), Network Threat Level (*NTL*) and the random nonce. The resulting output is hashed with **H** (the SHA-256 algorithm) to produce a fixed length hash *h* which is appended to a bundle as DTN-Cookie. The *IV* is changed periodically by the regional security gateway to ensure freshness.

$$DTN\text{–}Cookie = H((\langle TS|Src_{EID}\rangle|LS|CoS|NTL)\ XOR\ p\text{–}RNG(IV)) \qquad (2)$$

The difference between the first and second DTN-Cookie variant is the XOR operation. The XOR operation introduces bit flip which produces a stronger mechanism with a high degree of randomness and makes forgery more difficult.

$$DTN\text{–}Cookie = HMAC((\langle TS|Src_{EID}\rangle|LS|CoS|NTL)\ XOR\ p\text{–}RNG(IV), K_{RS}) \qquad (3)$$

The third DTN-Cookie variant uses keyed Message Authentication Code (HMAC) to provide DoS resilience and uses SHA-256 as the underlying hash function. The result of the operation is hashed with a regional secret key K_{RS} to produce a fixed length MAC which is appended to every bundle. The secrecy of the key K_{RS} and the *IV* makes the DTN-Cookie hard to forge. To prevent compromise, K_{RS} and the *IV* are periodically changed by the security gateway. We will test our proposed mechanisms in two case scenarios: In the first scenario, we assume that the attacker does not have the capability to spoof the source addresses of other nodes and does not hide his identity. In the second scenario, we assume that the attacker has the ability to spoof the source addresses of other nodes and can hide his identity.

3.1.1 Non-spoofing Scenario

Figure 4 is a flowchart which shows the steps a node takes to authenticate a bundle. In this first scenario, we assume that the attacker does not spoof the address of other nodes and does not hide his address. When a bundle arrives at a DTN node, the sender's address is checked against a Node Isolation Log (NIL). The NIL is used to store the addresses of all blacklisted or blocked nodes. All bundles from a blocked node are discarded without any form of processing. All nodes serve as Policy Enforcement Points (PEP) and are required to enforce their own configurable security policies [8]. In general, policies are defined as *Event-Condition-Action (ECA)* clauses, where on event(s) E, if condition (C) is (are) true, then action(s) A is (are) executed. To prevent the injection of bogus bundles, the receiving node proceeds to ensure that the bundle size is within the limits allowed by the application, the time-to-live (TTL) is not set into the future and that the bundle was sent using the authorised class-of-service (COS) rights. If any of these checks fail, the bundle is discarded and processed no further.

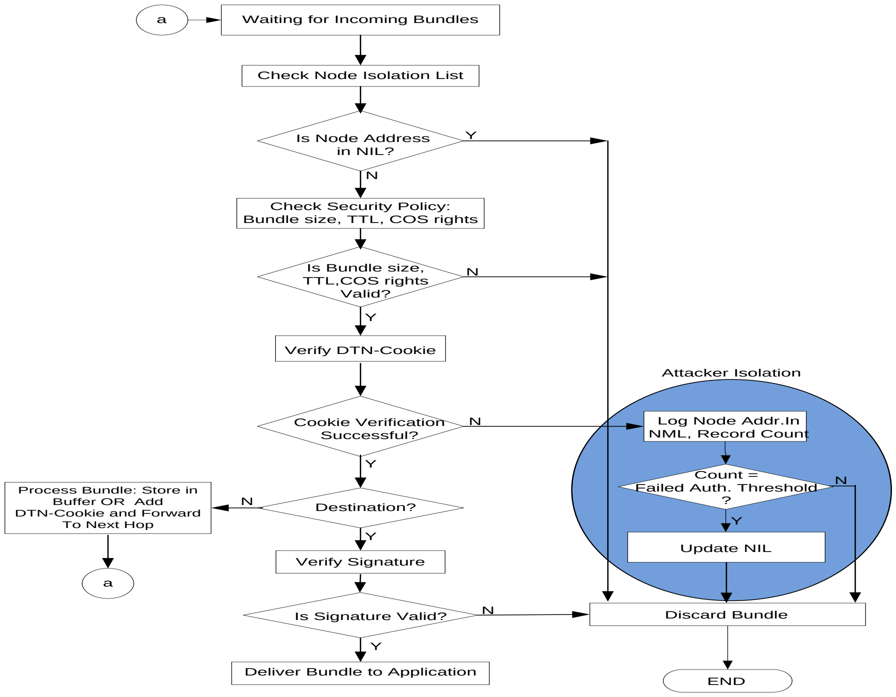
a
Waiting for Incoming Bundles
Check Node Isolation List
Is Node Address in NIL?
Y
N
Check Security Policy: Bundle size, TTL, COS rights
Is Bundle size, TTL,COS rights Valid?
N
Y
Verify DTN-Cookie
Attacker Isolation
Cookie Verification Successful?
N
Log Node Addr.In NML, Record Count
Count = Failed Auth. Threshold ?
N
Y
Update NIL
Y
Process Bundle: Store in Buffer OR Add DTN-Cookie and Forward To Next Hop
N
Destination?
Y
a
Verify Signature
Is Signature Valid?
N
Discard Bundle
Y
Deliver Bundle to Application
END

Figure 4: DoS Resilient Mechanism against Resource Exhaustion Attacks

The next step is to check the authenticity of the bundle by verifying the DTN-Cookie. A node's address is logged in the Node Misbehaviour Log (NML) if bundle authentication fails. A node is blocked if it has a configurable number of failed authentication attempts recorded against it in the NML. Conversely, if DTN-Cookie verification is successful, and the processing node is an intermediate node, the bundle is stored in the buffer until a contact opportunity arises. On the other hand if the processing node is the bundle destination, signature verification is triggered. The bundle is discarded if signature verification fails otherwise the bundle is delivered to the application.

3.2 Evaluation of the Design

In this section, we evaluate the strength and resilience of our proposed scheme against attacks. The proposed scheme requires a single bundle exchange to achieve bundle authentication and 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. In our design, every bundle has a timestamp embedded in it which is used to provide an accurate record of the bundle creation time and act as a freshness identifier. The concatenation of the timestamp and source_EID is a unique bundle identifier which provides a strong feature for thwarting replay attacks. Any attempt to modify the timestamp is detected during DTN-cookie verification. The payload is protected using a digital signature and can be encrypted for confidentiality. The BSP specification recommends RSA-1024 as the de facto digital signature algorithm for DTN. Recent studies show that Elliptic Curve Cryptography (ECC) is suitable for resource-constrained devices like sensors. The Elliptic Curve Digital Signature Algorithm ECDSA-160 supports 160-bit keys and provides the same level of security as RSA-1024 [9]. ECDSA-160 has a smaller key and signature size which makes it more energy efficient and results in bandwidth, memory and computational savings than RSA-1024 [9], [10], [11].

The use of a cryptographically secure random number generator and secret nonce values make the proposed mechanisms random and hard to forge**.** The first and second DTN-cookie 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 [12]. The hash operation produces a fixed-length DTN-cookie 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). Finally it is computationally infeasible to find any pair (x, y) such that H(x) = H(y) (strong collision resistance) [12]. The first and second DTN-cookie variants have all these properties in-built.

The third DTN-cookie 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 [12]. The irreversibility property of the one-way hash function and the secrecy of the symmetric keys (K_S, K_{RS}), makes the proposed DTN-cookie hard to forge.

3.3 Simulation Results and Performance Evaluation

We implement DoSRAM on the Opportunistic Network Environment (ONE) simulator [13] and evaluate its performance. ONE is an agent-based discrete event simulation engine and a number of simulation modules are updated by the simulation engine at each simulation step. The simulator models node movement, inter-node contacts, routing and message handling. Results are collected and analysed through visualization, reports and post-processing tools. Node movement is implemented by synthetic movement models or through existing movement traces. Connectivity between nodes is based on node location, communication range and bit-rate. Messages are generated through event generators or external events and unicast having a single source and destination host inside the simulation world. The Simulator is written in Java and the basic agents are nodes which model mobile endpoints and are capable of forwarding messages using a store-carry-and forward approach. Each node belongs to a group which is assigned a set of capabilities and a node inherits the capabilities of the group to which it belongs. Capabilities such as the node buffer size, message size, transmission range, transmission speed etc. are set in the Configuration file. More complex capabilities such as movement and routing are configured using specialized modules which models a particular behaviour for that capability.

For energy simulations, each node in the simulation world is assigned a fixed amount of energy. A node's energy profile is depleted when it transmits or receives messages, scans for the presence of other nodes during the discovery process and performs security processing. More details on the ONE simulator can be found in [13], [14]. Figure 5 shows the internal modules of the ONE simulator and how they link and interact with each other. One major limitation of the ONE simulator is that it does not implement any form of security. We have linked the ONE to cryptographic primitives using the Java Security Architecture and extended the simulator in order to implement our proposed security schemes.

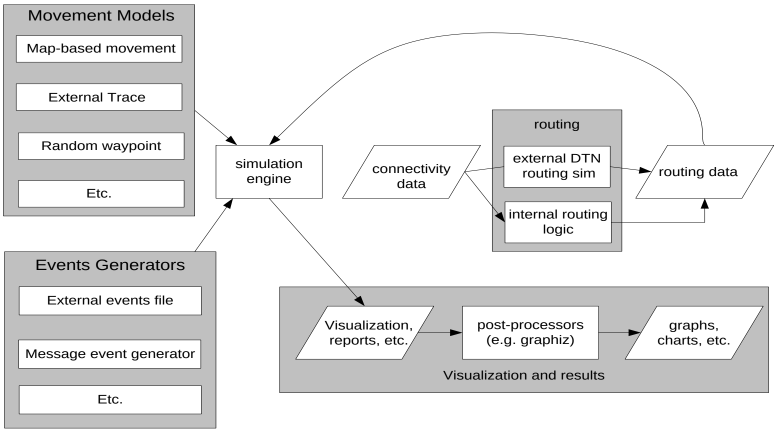

Figure 5: Overview of the ONE Simulator Environment [13]

In our simulations 100 nodes are uniformly deployed in a 4500 meters by 3400 meters area. Each node has a transmission range of 100m and travel at a speed of between 18 to 36 km/hr to conform to the selected scenario of vehicular networks. We vary the number of attackers from 10 to 50 to see what effect it has on bundle delivery ratio, average latency and energy efficiency of nodes. The attacker generates one bundle every 5 seconds, moves with a speed between 18 to 36 km/hr and uses the Random Way Point (RWP) Mobility model. We implement DoSRAM using the Spray and Wait routing protocol [15]. This scheme can also be implemented on other DTN routing protocols such as Epidemic [16]. To demonstrate the efficiency of DoSRAM, we compare it to the Bundle Security Protocol authentication mechanism based on RSA-1024 digital signatures. We use the second variant of our proposed DTN-Cookies for the simulations. Details of the simulation parameters are shown in the Table 1 where the node speed represents the speed of a car in an urban area. The message size is set at 64kB to prevent an attacker from arbitrarily sending large bundles to fill available buffer space. Spray and Wait routing protocol supports bundle replication. To prevent flooding we limit the number of forwarding copies to 2 to enhance bandwidth efficiency.

Table 1: Simulation parameters

Simulation Duration	43200 s (12hrs)
Number of Nodes	100
Speed of Nodes	18-36 km/hr
Transmission Range	100m
Mobility Model	Map-Based Mobility Model
Message Size	64kB
Message TTL	300 minutes
Message Generation Interval	60 ~ 120 (90) s
Routing Protocol	Spray and Wait
Number of Forwarding Copies	2 Copies
Buffer Size	5MB

3.3.2 Simulation Results

In this section, we use simulations to evaluate the performance of DoSRAM. One of the prime purposes of a DoS attack by a malicious node is to reduce the number of legitimate bundles delivered to destinations.

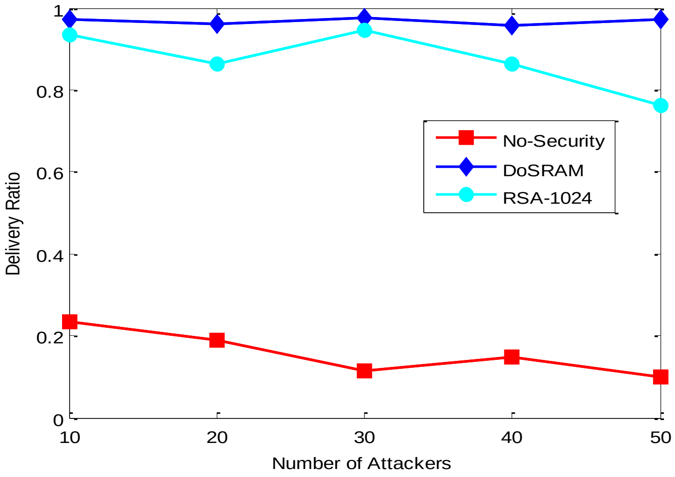

Figure 6: Effects of Increasing Number of Attackers on Delivery Ratio

In the ONE simulator, delivery ratio is defined as follows:

$$Delivery\ Ratio = \frac{number\ of\ bundles\ delivered}{number\ of\ bundles\ created\ by\ legitimate\ nodes}$$

Figure 6 shows the effect on bundle delivery ratio when the number of attackers increases. In the presence of mobile attackers with no DoS defence mechanism in place, the average number of bundles delivered is 76 compared to 585 bundles generated by honest nodes. This represents 15.7% of the overall bundles generated by honest nodes. The low delivery ratio is as a result of attack traffic using up resources such as bandwidth, buffers and memory. In this instance, legitimate traffic is dropped if their time-to-live expires or the buffers of nodes become full. The BSP RSA-1024 digital signature mechanism gives an average delivery ratio of 87.3% representing 424 out of 585 bundles delivered to destination. When we activate DoSRAM, the bundle delivery ratio rises to 96.8% (469 bundles) which represents 81.1% increase when compared when no security mechanism is used and 9.5% increase when compared to when BSP RSA-1024 digital signature is used. DoSRAM performs better because attack traffic is filtered and discarded freeing up resources for legitimate traffic (flows). The solution (DoSRAM) only drops legitimate traffic if their time-to-live expires

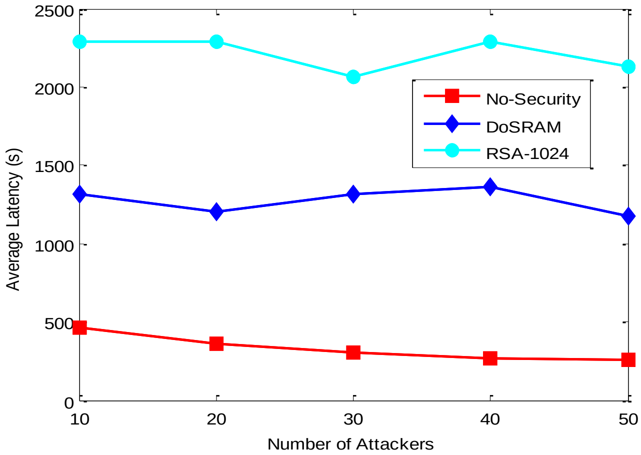

Figure 7: Effects of Increasing Number of Attackers on Average Latency

Delay is introduced as bundles traverse the network in a multi-hop routing scenario. There are a number of factors such as transmission, propagation, processing and queuing delays which affect latency in the network. In traditional end-to-end communications, latency is defined as follows:

$$Latency = transmission\ delay + propagation\ delay\ + processing\ delay + queuing\ delay$$

Transmission and propagation delays are influenced by the transmission medium. The capability of the DTN node has a direct influence on the processing and queuing delays. Each bundle has a time-to-live (TTL) of 300 minutes (18000 seconds). Security processing at DTN nodes introduces congestion which has a major role to play in affecting the latency levels in the network. Figure 7 shows the average latency experienced by each bundle. Without any security mechanism, the average latency experienced by a bundle is 125 seconds as the number of attackers increase from 10 to 50. This represents 0.69% of a bundle’s total TTL. When BSP RSA-1024 digital signature is used as the mitigation mechanism, the average latency per bundle is 2214 seconds which represents 12.3% of a bundle’s TTL. The high average latency is as a result of the resource-intensive operations associated with the computation and verification of RSA digital signatures. DoSRAM imposes a minimal amount of latency due to security processing at intermediate nodes. The average latency experienced by a bundle from source to destination is 1272 seconds which represents just 7.1% of the TTL of each bundle. Thus DoSRAM introduces a much lower latency when compared to RSA-1024 digital signature but higher latency than when no security mechanism is used.

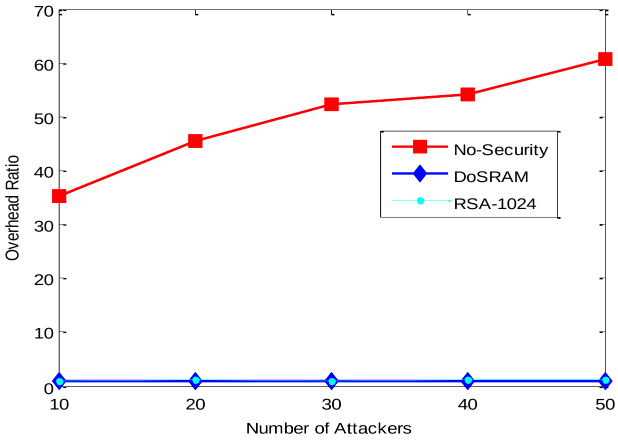

Figure 8: Overhead Ratio with Increasing Number of Attackers

$$\boldsymbol{Overhead\ Ratio} = \frac{\boldsymbol{(number\ of\ bundles\ relayed - number\ of\ bundles\ delivered)}}{\boldsymbol{number\ of\ bundles\ delivered}}$$

Overhead ratio is dependent on the number of relayed bundles and number of bundles delivered to destination. In Figure 8 with no security mechanism, the Overhead ratio keeps increasing as the number of attackers increase. The average overhead ratio is 49.5 (approximately 50) because only a small fraction of relayed bundles are actually delivered to destination. It should be noted that an attacker generates one bundle every 5 seconds and none of these are filtered from the network. This makes it less likely for legitimate bundles to be delivered to destination. When mitigations schemes are activated, overhead ratio drops drastically. Figure 8 shows that DoSRAM has a lower overhead ratio of 1.00556 when compared to BSP RSA-1024 (1.10386). This is because DoSRAM has a higher delivery ratio as shown in Figure 6. The graph showing the overhead for DoSRAM and RSA-1024 appear to merge because the difference in overhead ratio is just 0.1016 as shown in Figure 8. When DoSRAM is activated, it reduces the overhead ratio by as much as 48.49% compared to the instance when no security mechanism is used.

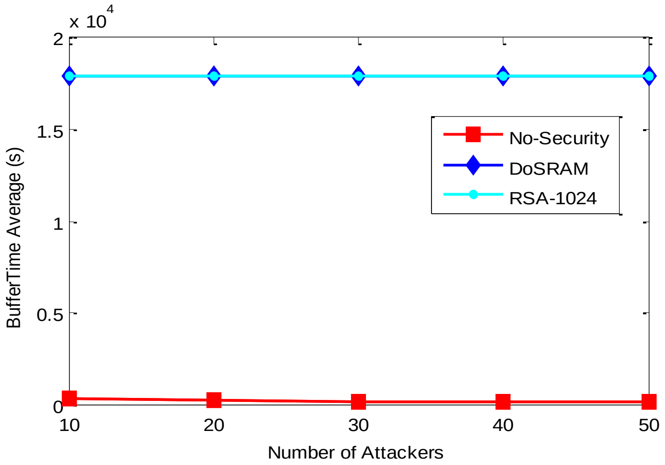

Figure 9: BufferTime Average with Increasing Number of Attackers

The BufferTime Average is a metric which shows the average time a bundle spends in the buffer as it traverses the network from source to destination. Buffer availability enhances the efficiency of the DTN carry-store-and-forward mechanism. Figure 9 shows that with no security mechanism the average buffer time of a bundle from a legitimate source is 174 seconds. This is as a result of the network being saturated with fake bundles from the attacker. A DTN node drops bundles when its buffer become full. This affects the number of bundles from honest nodes that are stored in buffers for onward forwarding to next hop towards the destination. With RSA-1024, the buffertime average is 17960 seconds which is the same as DoSRAM. DoSRAM filters and drops all attack bundles, increasing the buffertime average to 17960 seconds. Bundles from legitimate nodes experience extended buffertime average values thus increasing their chances of being delivered to destination.

Energy consumption is a major concern for most mobile and resource-constrained nodes. The battery life of nodes has a direct impact on the lifespan of a DTN network. A certain amount of energy is expended every time a node sends or receives a message. The amount of energy consumed by a security function for a given microprocessor is determined by the processor, power consumption, the processor clock cycle frequency and the number of clocks needed by the processor to compute the security function. The cryptographic algorithm and the efficiency of the software implementation determine the number of clock cycles necessary to perform the security function [17]. For symmetric cryptographic functions, the energy cost at both transmitter and receiver of a processed message are relatively equal. Asymmetric public-key cryptographic algorithms have different energy cost associated with a processed message at the receiver and transmitter. For the energy simulations 10 nodes are uniformly deployed in a 4500 meters by 3400 meters area and all configuration parameters in Table 1 remain the same. To obtain the average energy consumed by each node, we consider transmission energy, receive energy, computational energy and scan energy in our simulations. Every node has an initial energy of 2 x 10^6 mJ and we set transmit and receive energy to 0.0596 and 0.0286 mJ/s respectively. A node scans for other nodes once every 60 seconds and expends 0.0140mJ of energy each time. Each DTN-Cookie computation and verification requires 0.0059 mJ/s of energy while each RSA-1024 *Sign* and *Verify* operation requires 304 mJ/s and 11.9mJ/s respectively [9].

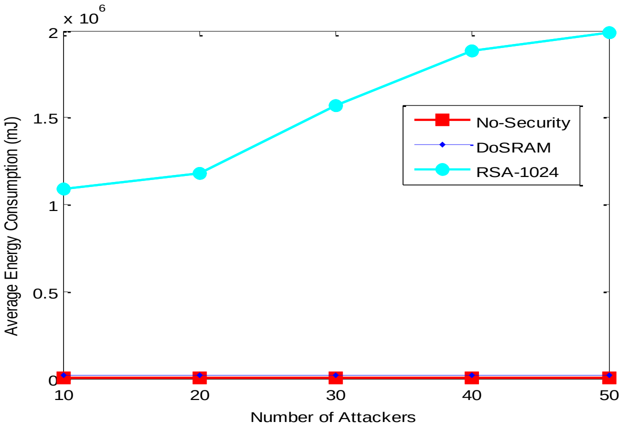

Figure 10: Average Energy Consumption with Increasing Number of Attackers

The result in Figure 10 shows the energy consumed in the propagation and authentication of bundles and how the increase in number of attackers affects the energy consumption of nodes. The average energy consumed by honest nodes with no security mechanism is 1274mJ which represents 0.06% of the total battery power. When Public-Key Cryptography (BSP RSA-1024) is used as security mechanism, the energy consumption rate rises sharply with an increase in number of attackers. With 50 attackers, the average energy expended by a node is 1.990419 x 10^6 mJ which represents 99.5% of a node's total battery power. When our mechanism (DoSRAM) is activated, the average energy consumption per node is 16557mJ which represents 0.83% of the total battery power. The average energy consumed by a node when DoSRAM is used is slightly higher than when no security mechanism is used. The energy consumption for DoSRAM is much lower than when Public-Key Cryptography (RSA-1024 digital signature) is used for bundle authentication. The result in Figure 10 clearly demonstrates the effectiveness of DoSRAM in tackling resource exhaustion attacks which targets the battery power of nodes when a security service like authentication is applied.

3.4 Spoofing Scenario

In the second scenario, we assume that the attacker can spoof source addresses. In our design as shown in Figure 11, we do not block (perform filtering) node addresses to avoid memory exhaustion and masquerade attacks. The only cost we will bear is in computational cost.

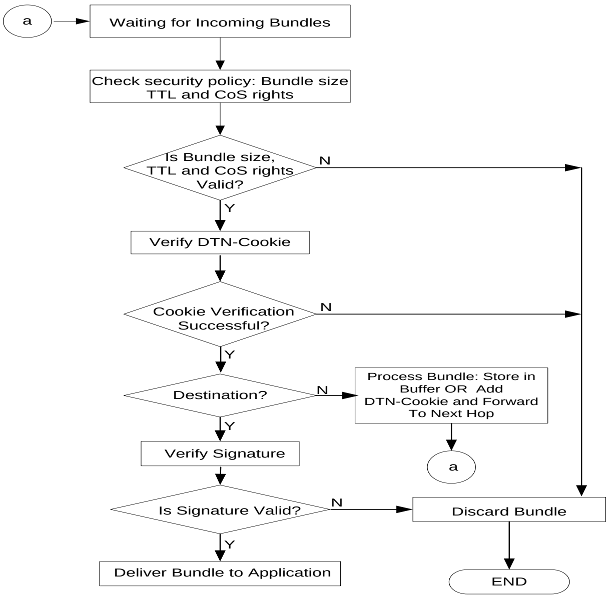
a
Waiting for Incoming Bundles
Check security policy: Bundle size TTL and CoS rights
Is Bundle size, TTL and CoS rights Valid?
N
Y
Verify DTN-Cookie
Cookie Verification Successful?
N
Y
Destination?
N
Process Bundle: Store in Buffer OR Add DTN-Cookie and Forward To Next Hop
a
Y
Verify Signature
Is Signature Valid?
N
Discard Bundle
Y
Deliver Bundle to Application
END

Figure 11: A DoS Resilient Mechanism against Resource Exhaustion and Spoofing Attacks

This is negligible as the result in Figure 14 shows which is primarily due to the light-weight mechanisms we have proposed. In Figure 11, when a bundle arrives at a DTN node, the receiving node checks to ensure that the bundle size is within the limits allowed by the application, the time-to-live (TTL) is not set into the future and that the bundle was sent using the authorised Class of Service (CoS) rights. If any of these checks fail, the bundle is discarded and processed no further. This is to prevent the injection of bogus bundles into the network. The next step is to check the authenticity of the bundle by verifying the DTN-Cookie. If the verification of DTN-Cookie fails, the bundle is dropped and processed no further. Conversely, if DTN-Cookie verification is successful, and the processing node is an intermediate node, the bundle is stored in the buffer until a contact opportunity arises. The bundle can be forwarded to the next hop (DTN node) or destination. On the other hand if the processing node is the bundle destination, signature verification is triggered. The bundle is discarded if signature verification fails otherwise the bundle is delivered to the application.

In the simulations, 10 nodes are uniformly deployed in a 4500 meters by 3400 meters area. We vary the number attackers from 10 to 50 to see how this affects delivery ratio, average latency, energy consumption and network overhead ratio. The purpose is to show the resilience and scalability of the proposed solution when there are more attackers than legitimate nodes in the network. Apart from the number of nodes, all other simulation parameters remain same as in Table 1. We use the second variant of DTN-Cookie for the simulations.

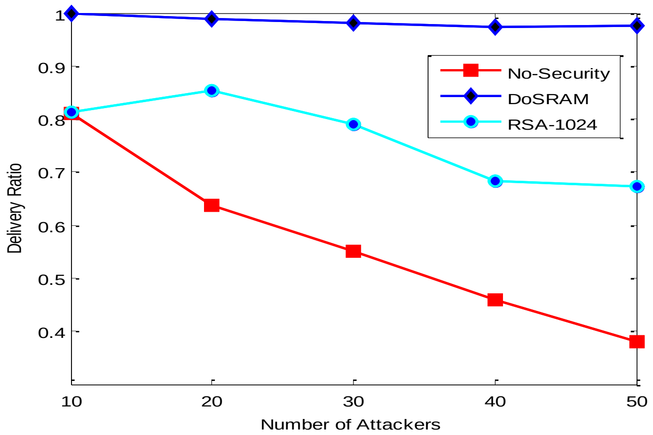

Figure 12: Increasing Number of Attackers and its Effects on Delivery Ratio

One of the prime purposes of a DoS attack is to disrupt or reduce network functionality. Network throughput is an important metric which we consider in our simulations. Figure 12 shows the effect of bundle injection DoS attack on delivery ratio. With an increasing number of mobile attackers and no DoS defence mechanism, the average number of bundles delivered is 275 compared to 484 bundles generated by honest nodes. This represents 56.82% of the overall bundles generated by honest nodes. It is evident that delivery ratio declines shapely with an increase in number of attackers in the network. Attack traffic uses up resources such as bandwidth, buffers, CPU processing cycles and memory. The BSP RSA-1024 digital signature mechanism gives an average delivery ratio of 76.24%. This means that out of the 484 bundles generated, 369 bundles were delivered to destination. Legitimate traffic is dropped when the time-to-live on the bundle expires as a result of processing delays and congestion. When we activate DoSRAM, there is a significant improvement in bundle delivery ratio to 98.3% (476 bundles) which represents a 41.5% increase when compared to when no security is used. DoSRAM performs better than RSA-1024 digital signatures with an overall improvement of 22.1% in bundle delivery ratio.

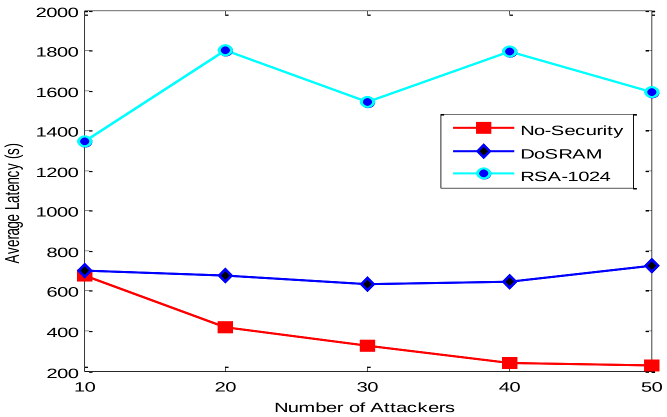

Figure 13: Increasing Number of Attackers and its Effects on Latency

As stated earlier, a number of factors such as transmission, propagation, security processing and queuing delays affect the latency in a system. Figure 13 shows that BSP RSA-1024 has the highest latency value when compared to the instance where DoSRAM or no security mechanism is adopted. Without any security mechanism, the average latency experienced by a legitimate bundle is 377.6 seconds which is 2.10 % of a bundle's TTL. The main reason we compare the latency value to the TTL of a bundle is because of the TTL determines how long a bundle can remain in the network. High average latencies reduce the chances for a bundle to be delivered to destination. With RSA-1024, the average latency is 1614.8 seconds which is 8.97% of a bundle's TTL. This can be attributed to the complex modular arithmetic associated with Public-Key Cryptography. The average latency when DoSRAM is adopted as security mechanism is 675.2 seconds. This represents 3.75% of a bundle's TTL. This result is significant in the sense that DoSRAM protects the network against DoS attacks and at the same time reduces the latency levels by more than half when compared to BSP RSA-1024.

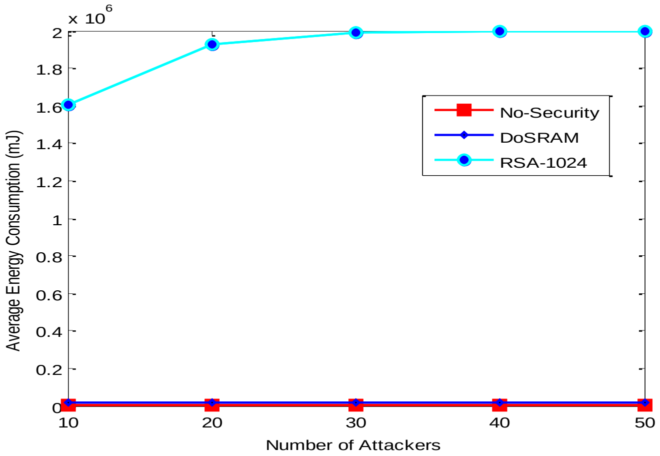

Figure 14: Energy Consumption with Increasing Number of Attackers

Figure 14 shows the effects of an increase in number of attackers on the energy consumption of nodes. The average energy consumed by honest nodes when no security mechanism is adopted is 1258mJ which represents 0.0629% of the total battery power. Energy consumption rises sharply when RSA-1024 digital signature is used as security mechanism. The average energy expended is 1.902955 x 10^6 mJ which represents 95.15% of a node's total battery power. When a node expends all its battery power, that node is effectively dead and cannot send or receive bundles. When DoSRAM is used, the average energy consumption is 17124mJ, which represents 0.8562% of a node's total battery power. The amount of energy consumed by a node when DoSRAM is used as security mechanism is slightly higher than when no security mechanism is used. Also, the energy consumption for DoSRAM is much lower than RSA-1024 digital signature. In the first scenario, the average energy consumption when DoSRAM is activated is 16569 mJ which represents 0.82845% of the total battery power of a node. This is 0.03% or 600 mJ lower than the average energy consumed by a node in the second scenario when DoSRAM is activated. This difference is as a result of the filtering (blocking) mechanism used in the first scenario. In the first scenario, a node is blocked for 20000 seconds if it has three failed authentication counts against it in the Node Misbehaviour Log (NML). Figure 14 clearly demonstrates effectiveness and efficiency of DoSRAM in both scenarios to protect DTN nodes against resource exhaustion DoS attacks.

4 CONCLUSION

From the literature, resource exhaustion is one form of DoS attack which DTNs are vulnerable. To protect the hop-by-hop bundle authentication service against this attack, a DoS-Resilient Authentication Mechanism (DoSRAM) has been proposed. Two different scenarios were considered: In the first scenario, we assumed that the attacker cannot spoof source addresses. A threshold is set for the number of failed authentication attempts which we assume is distributed as part of the security policy. The traffic flow for each node is monitored, failed authentication attempts are logged in the Node Misbehaviour Log (NML) and nodes which exceed the set failed authentication threshold are logged in the Node Isolation Log (NIL) for a set period of time. In the second scenario we assumed that the attacker can spoof addresses and we simply discard bundles that do not authenticate.

In terms of energy efficiency, the proposed DTN-Cookie is approximately 94% more energy efficient than RSA-1024 digital signature as the simulation results in Figure 10 and 14 show. Computationally, no key fetch operation is required if DTN-Cookie1 or DTN-Cookie2 is used as light-weight bundle authenticators. DTN nodes must store their public/private key pairs and public keys of other nodes in their key stores when using RSA to authenticate bundles. The proposed mechanism does not require DTN nodes to store keys in memory except when DTN-Cookie3 is used as the light weight bundle authenticator. Memory savings is achieved by dropping all attack bundles at the first point of contact during bundle verification. This makes more memory available to store legitimate bundles as the results in Figure 7 and 12 show. In the first scenario, the proposed mechanism allows 9.5% more bundles be delivered to destination than RSA-1024. In the second scenario where there are more attackers than legitimate nodes, the delivery ratio is 22.1% higher when the DoSRAM is used compared to RSA-1024 digital signature.

In both scenarios, the results presented show significant improvements in delivery ratio, buffer time average, average latency and energy consumption of DoSRAM over schemes which use RSA-1024 digital signatures for bundle authentication. In addition, the results for the second scenario show the robustness of DoSRAM in maintaining a high delivery ratio even when the number of attackers is more than the number of legitimate nodes. In terms of scalability, the results from both scenarios show that the proposed mechanism can support a large number of nodes. The results show a degradation of performance for schemes that use RSA digital signature as the network size increases.

5 REFERENCES

[1] K. Fall, “A Message-Switched Architecture for Challenged Internets,” Technical Report IRB-TR-02-010, Intel Research, Berkeley California, USA, 2002.

[2] P. Ferguson, “Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing,” IETF Network Working Group, RFC 2267, 1998.

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

[4] T. Killalea, “Recommended Internet Service Provider Security Services and Procedures,” 2000.

[5] The International Telegraph and Telephone Consultative Committee (CCITT), “Security Architecture for Open Systems Interconnection for CCITT Applications,” 1991.

[6] S. Symington, S. Farrell, H. Weiss and P. Lovell, “Bundle Security Protocol Specification,” Network Research Group, Draft-irft-dtnrg-bundle-security-17, 2010.

[7] K. Fall and S. Farrell, “DTN: An Architectural Retrospective,” IEEE Journal on Selected Areas in Communications, vol. 26, no. 5, pp. 828 - 836, June 2005.

[8] T. Small and Z.J. Haas, “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.

[9] A. S. Wander, N. Gura, H. Eberle, V. Gupta and S.C. Shantz, “ Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks,” in *3rd IEEE International Conference on Pervasive Computing and Communications*, 2005.

[10] O. Arazi, H. Qi and D. Rose, “A Public Key Cryptographic Method for Denial of Service Mitigation in Wireless Sensor Networks,” in *4th Annual IEEE Communications Conference on Sensor, Mesh and Ad hoc Communications and Networks*, San Diego, CA , 2007.

[11] K. Ren, S. Yu, W. Lou and Y. Zhang, “Multi-user Broadcast Authentication in Wireless Sensor Networks,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 223 - 232 , October 2009.

[12] M. Belware et al., “Keying Hash Functions for Message Authentication,” in *Advances in Cryptology-CRYPTO'96*, 1996.

[13] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE Simulator for DTN Protocol Evaluation,” in *2nd International Conference on Simulation Tools and Techniques(SIMUTools'2009)*, Rome, Italy, 2009.

[14] TKK/COMNET, “Project page of the ONE Simulator,” 2008. [Online] Available: http://www.netlab.tkk.fi/tutkimus/dtn/theone. [Accessed 12 May 2011].

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

[16] A. Lindgren, A. Doria and O. Schelén, Probabilistic Routing in Intermittently Connected Networks, Vols. 3126 239-254, Lecture Notes of Computer Science, 2004.

[17] H. Jun, M.H. Ammar and E.W. Zegura, “Power Management in Delay Tolerant Networks: a framework and Knowledge-Based Mechanism,” in 2nd IEEE Communications Society Conference on Sensor and Ad hoc Communications and Networks, 2005.