A Proactive DOS Filter Mechanism for Delay Tolerant Networks

Godwin Ansa, Haitham Cruickshank, and Zhili Sun

Centre for Communications Systems Research, University of Surrey, GU2 7XH, England
{g.ansa,h.cruickshank,z.sun}@surrey.ac.uk

Abstract. Denial of Service (DOS) attacks are a major threat faced by all types of networks. The effect of DOS in a delay tolerant network (DTN) is even more aggravated due to the scarcity of resources. Perpetrators of DOS attacks in DTN-like environments look beyond the objective of rendering a target node useless. The aim of an attacker is to cause a network-wide degradation of resources, service and performance. This can easily be achieved by exhausting node or link resources and partitioning the network. In this paper we seek to provide a proactive approach in making the DTN authentication process robust against DOS. Our aim is to make security protocols which provide mandatory DTN security services resilient to DOS attacks. The overall objective is to make it hard to launch a DOS attack and ensure the availability of DTN services. A DTN-cookie mechanism has been proposed to quickly identify and filter out illegitimate traffic.

Keywords: Denial of service, attacker, delay tolerant network, resource exhaustion, DTN-cookie.

1 Introduction

Delay tolerant networking is fast becoming an area of great research interest. Where there is no direct link between a source and a destination, a node in one region can pass a message to another node in a remote region using store-and-forward message switching technique. Store-and-forward message switching technique or asynchronous message passing [1] as illustrated in Fig. 1 requires that the integrity of a message is verified by an intermediate node before it is forwarded.

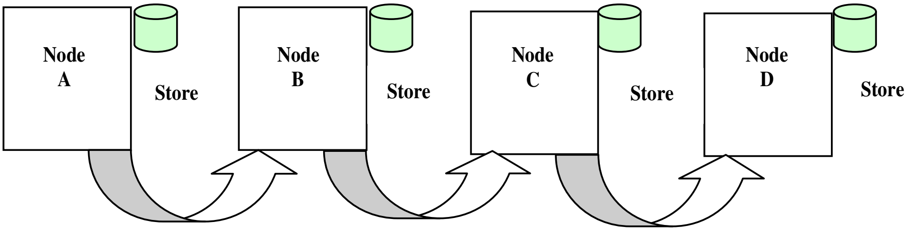

Fig. 1. DTN Store-and-Forward Message Switching

G. Giambene and C. Sacchi (Eds.): PSATS 2011, LNICST 71, pp. 213–226, 2011.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2011

In Wireless Sensor Networks (WSNs), messages are forwarded to a destination through intermediate nodes which act as routers. Sensor nodes are resource-constrained in memory, CPU cycles, battery power, and bandwidth [2]. It is therefore expedient that DTN-enabled sensor nodes forward or take into custody only bundles that are authentic and still within their useful lifetime.

Denial of service attacks poses a major threat to availability because it prevents an entity or network from fulfilling its functions by disabling or degrading the services it provides [2], [3]. A classification of DOS attacks by the Computer Emergency Readiness Team (CERT) groups DOS into three main categories [4]: Destruction or alteration of configuration information, physical destruction or alteration of network components and consumption of scarce, limited, or non renewable resources. Our main focus is to protect the scarce resources of the DTN which include communication contact time, battery power, bandwidth, CPU cycles, disk space and memory from exhaustion.

We present a DTN scenario with two Wireless Sensor Networks and a third network where the sensor data are processed. Each network has a security gateway which acts as an interconnection point to facilitate inter-regional communications. The three networks are bridged using a satellite as relay node. Within each region opportunistic message forwarding or short-range radio communication such as Bluetooth can be used. Data mules can be deployed to collect sensitive sensor data in a more scheduled manner.

Each sensor network is depicted as shown in Fig. 2, is divided into X number of domains by the security gateway. This is done during the initial set-up phase and subsequently during periodic refreshments of the regional secret key and nonce values. Each domain comprises a Group Head (GH) node, one or more security-aware nodes and numerous sensor nodes.

The remainder of the paper is structured as follows. Section 2 gives a brief overview of related work on DOS defence mechanisms in different networks. The network threats, design objectives, networking and security requirements are presented in section 3. Section 4 provides design details of the DOS-filter mechanism for both intra-regional and inter-regional scenarios. It also enumerates the design assumptions, and describes in-depth the use of loose time synchronization of the security gateways. A detailed evaluation of the proposed mechanism is carried out in section 5. Section 6 concludes the paper and gives a summary of our contribution towards DOS resilience in DTN.

2 Related Work

A clever attacker can exploit the strong security of a system or network to launch a protocol-based DOS attack through flooding and resource exhaustion. A security protocol is prone to this type of DOS attack if the server commits memory or computational resources during the client authentication process. In terrestrial networks, a number of solutions have been proposed to tackle this problem. An initial work by Dwork and Noar [5] in tackling the junk mail problem proposed the use of cryptographic puzzles where a sender is required to compute a puzzle for every message sent. The cost of this technique is negligible for normal users when compared to mass mailers. The client puzzle idea was extended to connection depletion attacks such as the TCP SYN flood attack by Juels and Brainard [6].

Another technique which combats protocol-based DOS is the Internet Security Association and Key Management Protocol (ISAKMP) defined by IPSec and derived from the PHOTURIS protocol. It is an anti-clogging technique where a client is required to return a server generated cookie. See [4] and [7] for more information on the ISAKMP specification. Meadows [8] proposed a formal framework for network DOS. The idea is to gradually strengthen the authentication process as the protocol executes by introducing a weak authentication phase prior to signature verification. Leiwo et al. [9] suggest that allocation of server resources can only take place after client authentication, and that a client's workload must be greater than that of the server.

The aforementioned DOS mitigating solutions proposed for terrestrial systems are only suitable for low-delay well-connected networks but unsuitable for delay tolerant networks. Sensor nodes are resource-limited in battery power and computational capabilities and will not be able to solve the cryptographic puzzles; the ISAKMP cookie requires a number of message exchanges during client authentication which is infeasible in DTN. The round-trip delay, broadcast nature of the satellite channel and the wireless communication medium makes the scenario described in section 1 prone to eavesdropping, interception of cookie values and masquerade.

The authors of [10] define a header extension field with no related trailer field and three ciphersuites for the specification. In their definition, a cookie value can be a long random number whose length is determined by the implementation. They assert that longer cookies are stronger and harder to guess but consume more bandwidth.

3 Threat Analysis, Design Objectives, Networking and Security Requirements

This section provides a detailed threat analysis for the DTN scenario described in section 1. It outlines the design objectives and the networking and security requirements

3.1 Threat Analysis

The DTN scenario presented in section 1 is prone to eavesdropping due to the wireless communication medium and the broadcast nature of the satellite channel. The depicted scenario is also susceptible to bundle content modification, masquerade, and denial of service attack. The Bundle Security Protocol (BSP) specification [11] states that bundles have to be validated at a security-aware node for authenticity and integrity. The BSP defines four security blocks for this purpose, for more details see [11] and [12]. To protect the scarce resources of the DTN it is mandatory that bundles are BAB-protected and validated. The aim is to ensure that network resources are used solely for forwarding authentic bundles with valid lifetimes.

A clever attacker can exploit this requirement to flood the network with small-sized BAB-protected bundles or flood a target node directly. Since the BABs on the bundles are fake, a security-aware node will waste its resources (CPU and battery) trying to verify the bundles. Victim nodes can become congestion points and legitimate bundles with no access to an alternative next hop node might be dropped if they expire on transit.

3.2 Security Objectives

Our primary objective is to make DTN security protocols resilient to DOS attacks launched through resource exhaustion by ensuring that the authentication process is robust and light-weight. The DOS filter mechanism should detect and discard malicious traffic as early as possible, detect and discard bundles whose headers have been modified, ensure that unauthorized entities do not gain control of the DTN infrastructure.

3.3 Networking and Security Requirements

It is imperative that a DOS filter mechanism for DTN should be able to withstand significant node mobility, run efficiently on resource-limited nodes like sensors and be resilient to delays which can be in the order of minutes, hours or days. The mechanism should support varying data rates and withstand changes in contact times. It should also be able to operate efficiently in the absence of an end-to-end path between source and destination.

In terms of security requirements, we restrict security processing to computationally capable nodes. To ensure freshness, we use nonce and timestamps to thwart the replay of old and expired bundles. Every bundle is checked for integrity to prevent bundle content modification during transit. Bundles are authenticated to ensure that they originate from legitimate sources and are still within their useful lifespan.

4 Design of a DOS Filter Mechanism for DTN Environments

A DOS filter mechanism should have a detection, classification and response element to be highly effective [3]. As design requirements, the DTN-cookie generation process must be simple and fast; the DTN-cookie value must be random and hard to forge, discourage Transport Layer Security (TLS) style of negotiation or handshake, the verification of the DTN-cookie should provide a weak authentication phase which is light-weight. Strong authentication can only take place if the weak authentication phase is successful; bundles that fail the weak authentication should be silently dropped.

Each sensor network is divided into X number of domains by the security gateway. This is done during the initial set-up phase and subsequently during periodic refreshments of the regional secret key and nonce values. Each domain comprises a Group Head (GH) node, one or more security-aware nodes and numerous sensor nodes. A GH node makes decisions on behalf of other nodes within its domain. It determines the Network Threat Level (NTL). The GH node and security-aware nodes act as data aggregation points for sensed data to ease management and coordination. These are more powerful in terms of computational and storage capabilities and act as security-sources and security-destinations for less-capable or IDless sensor nodes.

Bundle size within the DTN is fixed at 64KB for ease of processing. A node can only interact with the security gateway and other nodes within the same region. Inter-regional communication is gateway-to-gateway via the satellite whose pass is scheduled and can be predicted. The satellite acts only as a relay node for inter-regional communications. We define a new Administrative (AD) bundle called Alert for the dissemination of security information within each DTN region. An AD bundle has an expedited Class of Service (CoS) with priority of 1, while a data (D) bundle has a priority of 2 or 3 equivalent to Normal or Bulk CoS respectively.

The primary objective is to make DTN security protocols resilient to DOS attacks launched through resource exhaustion by ensuring that the authentication process is securely robust and light-weight.

4.1 Design Assumptions

- a node has bounded resources that could possibly be exhausted by a clever attacker
- the computational resource of the attacker is very large
- the attacker can compute efficiently pseudo-random functions and MACs in record time but does not possess the network secrets
- the attacker is assumed to have the ability to replay, modify, transmit, and receive bundles.
- the attacker has the ability to execute the protocol
- the attacker does not take over a legitimate node and in the process steal keying material and sensed data
- in inter-regional communications, the attacker is assumed to be a rogue router with enormous processing and sending capabilities and can predict the pass/schedule of the satellite
- trust is established during initial registration of a security- aware node with the security gateway
- the protocol is between two communicating entities
- a bi-directional communication asymmetry between a pair of nodes is assumed

A secure Key Management mechanism is required for the distribution of nonce seeds and cryptographic secret keys.

4.2 Intra-regional DOS Mitigation

Fig. 2 depicts the intra-regional DOS scenario in a DTN wireless sensor network which shows sensor nodes, Group Heads (GHs), security-aware (SA) nodes, an attacker and a security gateway. All in-bound and out-bound traffic must pass through the security gateway.

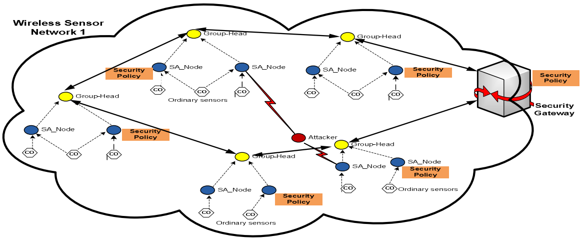

Fig. 2. Intra-Regional DOS Scenario

A generic DTN bundle is made up of the primary block and the payload block. Additional blocks such as the BAB, PIB and PCB can be added to provide security to the traffic. This is depicted in Fig. 3 and more details on security blocks can be found in [11] and [12]. To provide DOS-resilience in DTN, we propose a new security extension block called a DTN-COOKIE which adds a weak authentication phase and has less computational overheads at the security-aware nodes.

Primary Block					Security Blocks				Payload Block		Security Result		
TS	LS	Src_EID	Dst_EID	CoS	RSA with SHA256	DTN-COOKIE	BAB	PIB	ESB	Data	H (M)	privK (H (M))	COOKIE

Fig. 3. A DTN Bundle with Security Extension Blocks

- TS: the timestamp which is a concatenation of a bundle creation time and a monotonically increasing sequence number which is unique for every new bundle from a Source Endpoint Identifier (EID)
- LS: the bundle lifespan or expiration time. The LS value of a bundle can be in minutes, hours or days
- Src_EID: the Source EID of a bundle, and we assume that each EID is a singleton
- Dst_EID: the Destination EID i.e. the entity for which the bundle is destined
- RSAwithSHA256: represents the ciphersuite and gives an idea to what security blocks are in use
- M: the bundle payload
- H (M): h is the hash value derived by passing the payload M through the 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
- pubKXi: the public key of node Xi
- privKXi: the private key of node Xi

The BSP specification [11] provides minimal protection against DOS attacks. DTN nodes simply drop bundles that fail the authentication and access control checks. We have identified resource exhaustion as a simple means of launching DOS attacks and causing availability problems in DTN. For the intra-regional DOS scenario, we propose three variants of the DTN-cookie which can be dynamically chosen based on the perceived Network Threat Level (NTL).

$$\text{DTN-cookie} = h = H\,((\text{Timestamp} \mid \text{Src\_EID}_x) \mid \text{p-RNG (IV)}) \text{ - v1}$$

The Initialization Vector (IV) is known only to registered nodes of the region. The IV value is used to seed the pseudo-Random Number Generator (p-RNG); the result is a random long integer value. A concatenation of the timestamp and bundle source EID provides a unique bundle identifier. This unique bundle identifier is concatenated with the random long integer. It is then hashed using a one-way hash function H (SHA-256 algorithm) to produce a fixed-length hash value h. It is the hash value h that is appended to a bundle as DTN-cookie. The IV is changed periodically by the regional security gateway to ensure freshness.

$$\text{DTN-cookie} = h = H\,((\text{Timestamp} \mid \text{Src\_EID}_x) \text{ Xor p-RNG (IV)}) \text{ - v2}$$

The second variant of DTN-cookie (v2) is the result when we perform a bit-wise Exclusive-OR operation on $\text{Timestamp}|\text{Src\_EID}_x$ and p-RNG (IV) which results in the flipping of the bits. | is the concatenation operator, p-RNG (IV) is the same as in v1. This variant of DTN-cookie has more randomness and provides a stronger DOS solution.

$$\text{DTN-cookie} = \text{HMAC}\,((\text{Timestamp}|\text{Src\_EID}_x) \text{ Xor p-PNG (IV)}, K_{RS}) \text{ - v3}$$

The third variant of DTN-cookie (v3) is derived in the same way as v2 with SHA-256 as the underlying hash function. The only difference is that the result of the operation is hashed with a regional secret key K_{RS} to produce a fixed-length MAC which we append to every bundle. The mode of generation of the secret key, the use of p-RNG and bit-wise Exclusive-OR operation inputs more randomness to the DTN-cookie. The secrecy of the IV and the key makes the DTN-cookie hard to forge. These values are changed periodically by the security gateway to prevent compromise and ensure freshness.

When a bundle arrives at a security-aware node, the Bundle Protocol Agent (BPA) examines the bundle to determine if it is from a legitimate source and not expired, if it is a Data, or Alert bundle. Next the BPA tries to determine the perceived Network Threat Level (NTL) associated with the bundle. It does this by looking at the DTN-COOKIE Block. The DTN-COOKIE Block contains the NTL indicator (where Low = 1, Mild = 2, Severe = 3). Based on the NTL indicator, the BPA is able to choose which ciphersuite to use to verify the DTN-cookie. The DTN-COOKIE Block has a trailer block with the security result of the DTN-cookie computation as payload. The BPA can also use the Class of Service parameter to deduce the type of bundle it is dealing with by looking into the CoS field in the bundle primary block.

The perceived Network Threat Level (NTL) is determined by the GH node of the affected domain in a localized fashion. Every security-aware node keeps a Node Misbehaviour List (NML). If a SA node records three failed authentication entries against a node within the timeframes shown in Table 1, a Misbehaviour Alert notification is sent to the GH node in the affected domain. The Misbehaviour Alert bundle carries a DTN-cookie that matches the SA's perceived NTL. Entries on the NML beyond 120 minutes are flushed to create space and save memory.

Table 1. Network Threat Level and Associated DTN-Cookie Variants

Network Threat Level (NTL)	NTL Classification	Timeframe (minutes)	DTN-cookie Type
NTL 1	Low	51 - 120	DTN-cookie (v1)
NTL 2	Mild	31- 50	DTN-cookie (v2)
NTL 3	Severe	0 - 30	DTN-cookie (v3)

Table 1 shows NTL values and the associated DTN-cookie variants. This helps security-aware nodes to determine which DTN-cookie variant to use. Bundles arriving with lower NTL values will be processed as long as the node's EID is legitimate. The NTL threshold value reverts back to LOW if there are no Alert updates from a GH node within 24 hours. This is a measure designed to save power and make the DOS mechanism dynamic and adaptive to changing Network Threat Levels.

Egress filtering is enforced at security gateways to help ensure that no malicious or attack traffic leaves the region. The purpose is to prevent an attacker within the network from spoofing any source_EID in a bid to launch a DOS attack. To achieve this, the source_EID must belong to a valid node within the region. The egress filtering policy at security gateways requires all out-bound bundles to have a BAB. Also rate limiting techniques can be used to police the network interface at the security gateway to prevent an attacker from flooding it with bogus bundles. A step-by-step process of providing DOS-resilience within a DTN region is shown in Fig. 4.

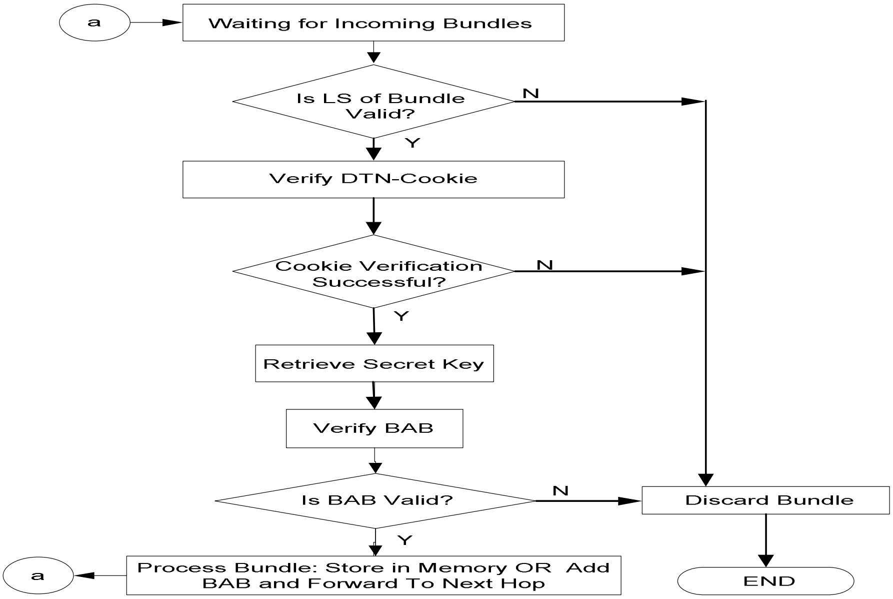
a
Waiting for Incoming Bundles
Is LS of Bundle Valid?
N
Y
Verify DTN-Cookie
Cookie Verification Successful?
N
Y
Retrieve Secret Key
Verify BAB
Is BAB Valid?
N
Discard Bundle
Y
a
Process Bundle: Store in Memory OR Add BAB and Forward To Next Hop
END

Fig. 4. Intra-Regional Dos Mitigation Flow Diagram

4.3 Inter-regional DOS Mitigation

Protecting inter-regional communications against DOS attacks is very vital to the survivability of a DTN and guarantees the availability of its services. A detailed description of the scenario is given in section 1. The satellite segment of the proposed solution is there to support inter-regional communications between two or more remote or isolated regions. In this scenario, the satellite is only a relay node to provide connectivity [4]. Large round trip times (RTT), large bandwidth delay product, burst errors on coded satellite links and variable RTT have impact that affect transport layer and application performance [4], [15].

The high altitude of the satellite provides a large terrestrial coverage which allows security gateways to send and receive messages to and from the satellite. Satellites are not affected by limitations such as line of sight range of ground-based nodes [15]. For more details on the advantages of satellite communications see [15]. The proposed solution for DOS mitigation in this scenario is similar to that for the intra-regional scenario. We assume that the security gateways are workstations with enormous storage, CPU processing and power capabilities. We use v3 as the proposed DTN-cookie to provide weak authentication in this scenario.

Ingress filtering is enforced at the security gateways to ensure that before a bundle is processed, it must originate from a legitimate and known source. Ingress filtering is a “good neighbour” policy based on mutual cooperation among gateways to thwart source address spoofing DOS attacks. An attacker may decide to spoof the source_EID of a legitimate gateway in order to mount a DOS attack. Such attacks will be thwarted during the verification of the DTN-cookie since the attacker does not know the secret of the network. The same steps shown in Fig. 4 apply to inter-regional communications with subtle differences. Before a bundle is processed any further, the gateway will have to verify the DTN-cookie. The composition of the DTN-cookie used for inter-regional communications is defined below where K_S is the inter-regional secret key.

$$\text{DTN-cookie} = \text{HMAC}\ ((\text{Timestamp}|\text{Src\_EID}_x)\ \text{Xor p-RNG (IV)},\ K_S)$$

The DTN-cookie is derived in the same way as specified for the intra-regional scenario. The only difference is that in the inter-regional scenario, we use variable nonce values in different timeslots to seed the p-RNG. This solution requires the security gateways to be loosely time-synchronized. Communication time is divided into timeslots of 2 hours interval with each timeslot having its associated nonce value such that (0-2) : S_1, (2-4) : S_2, ...:..., (20-22) : S_{11}, (22-24) : S_{12} where (0-2), (2-4), ..., (20-22) and (22-24) are the timeslots and S_1, S_2..., S_{11}, S_{12} represent the associated nonce values. Section 4.3.1 provides a detailed description of the process. The ingress filtering policy requires all in-bound bundles to have a BAB, PIB and PCB block. The BAB should be used for integrity and sender-side authentication.

We define the BAB to be a digital signature using asymmetric ciphersuite. Neighbour authentication in inter-regional communications does not make sense since the gateways belong to different regions [12]. Inter-regional communications should be gateway-to-gateway given the limited power budget of wireless mobile nodes. This is to enhance the survivability of the network and prevent mobile rogue routers from keeping the satellite busy with fake or mal-formed bundles during each pass of the satellite. Payload encryption is required in gateway-to-gateway communications for both maintenance traffic and data bundles. Also rate limiting techniques can be used to police the network interfaces at the security gateways to guard against flooding attacks.

4.3.1 Time-Synchronization of Security Gateways

Time-synchronization is an important aspect to be considered when designing a mechanism to provide DOS-resilience in DTN. In section 5.2 of [4], the authors elaborate on the use of timestamps and the need for time synchronization. We assume initial pre-shared symmetric keys between SGW_{HQ}, SGW_{WSN1} and SGW_{WSN2}. The security gateways have a common view of time (say UTC) irrespective of their time zones. Also, the p-RNG functions at the security gateways have a uniform initial seed value (S_o). Communications among the gateways is initiated by the SGW_{HQ} by sending two different S_1 nonce values to SGW_{WSN1} and $SGW_{WSN2.}$ The nonce is the bundle payload and is encrypted using the public key of SGW_{WSN1} and SGW_{WSN2}. The SGW_{HQ} signs the bundles using its private key, calculates the DTN-cookie, appends it to the bundles and sends to the gateways.

At the WSN-SGWs, the timestamp and sender EID are retrieved from the bundle and based on the pre-shared symmetric keys (K_S) between the SGW_{HQ}, SGW_{WSN1} and SGW_{WSN2}. The DTN-cookie is computed and compared to that on the received bundle. The bundle is silently dropped if the DTN-cookie verification is unsuccessful. On the other hand, if the verification is successful we proceed to test the integrity of the BAB (digital signature). Each SGW_{WSN} uses the public key of the SGW_{HQ} to verify the signature. If the signature verification fails the bundle is dropped because its content is considered modified on transit. If the verification of signature is successful, we proceed to decrypt the payload. Each SGW_{WSN} uses its private key to decrypt the payload which is the new reference nonce for communications. Attackers within the satellite's coverage are able to see every communication since the satellite uses a broadcast channel. To prevent eavesdropping of the nonce value, we encrypt the payload. We define a bound for the generation of nonce values as follows: $0 < S_i< 999999$ where i is a positive integer. If the S_i value generated is greater than the defined upper-bound, the entire seed generation process is started all over again.

A security gateway with data to send first chooses a random number within the pre-defined bound which it uses as seed to the p-RNG function. The result of this operation is a nonce which it sends to a destination gateway. This is done following the steps described earlier above. The gateways remain synchronized using previous nonce values as seed to the p-RNG function. Where initial nonce equals S_o, S_1 = p-RNG (S_o), S_2 = p-RNG (S_1), S_3 = p-RNG (S_2) and so on thereby forming a hash-chain of nonce values.

$$\text{DTN-cookie=HMAC ((Timestamp|EID}_{HQ\text{-}SGW}\text{) Xor p-RNG (}S_i\text{), }K_S\text{)}$$

One important property of one-way hash chains is that intermediate values can be recomputed using subsequent values in the chain. Bundles that arrive after their timeslot can still be processed if they are not expired. Also SGW_{WSN1} and SGW_{WSN2} can communicate with each other via the satellite and can remain synchronized by following the steps as described.

5 Analysis and Evaluation of Design

We have critically analyzed the solutions proposed for the terrestrial Internet and other networks and found them unsuitable for DTN. Our design minimizes the number of roundtrips required during entity authentication by discouraging TLS-like handshake. The use of NTL indicators and their associated DTN-cookies in the intra-regional scenario makes the proposed solution dynamic, energy-efficient, and provides DOS-resilience in a localized fashion. Our solution for the inter-regional scenario uses variable nonce values based on a hash-chain of previous nonce values. The security gateways are assumed to be powerful workstations that are loosely time-synchronized with enormous processing, computational, and storage capabilities.

The proposed design is light-weight since the process of generating the DTN-cookie is simple and fast and requires less CPU processing cycles and power. Due to the limited storage and computation capacities of sensor nodes, we use symmetric cryptography for our proposals. The DTN-cookie variants use simple cryptographic primitives such as hash functions and MACs because they require less computational requirements. Symmetric cryptography and hash functions are four orders of magnitude faster than public-key cryptography and digital signatures. The DTN-cookie variants are random and hard to forge because a cryptographically secure pseudo-random number generator in conjunction with fixed or variable seed values are used to generate the nonce. The fixed/variable seed values and the secret keys (K_S, K_{RS}) are prerequisites for computing a valid nonce and DTN-cookie which are unknown to the attacker.

A unique feature of the DTN-cookie 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 DTN-cookie verification.

The design is similar to fail-stop protocols described in the work of Gong and Syverson [13], but different because it introduces a weak authentication phase prior to strong authentication. The design also follows a proposed framework by Meadows [8] where a server gradually gains assurance of the client's intentions at every step during protocol execution. By providing a weak authentication phase, the design is able to quickly identify and discard bogus bundles from an attacker.

The v1 and v2 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 [14]. The hash operation produces a fixed-length DTN-cookie which saves memory, CPU processing and provides integrity. 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) [14]. The v1 and v2 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 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 [14]. 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.

Attacks which aim at substituting or tampering with the bundle payload are thwarted during the signature verification phase which is triggered if the weak authentication succeeds. The v3 variant of DTN-cookie proposed for the inter-regional scenario is light-weight, random, hard to forge and is a much stronger mechanism. The computation of a one-way hash chain of nonce values is lightweight. Hash-chain based authentication requires time-synchronization at granularities which might require special hardware [16]. Our design proposes the security gateways to be loosely-time synchronized and we assume that storage at the security gateways is large.

Table 2. Comparison of DTN-Cookie Variants

DTN-cookie Type	Complexity	Robustness	Processing	Energy Efficiency	Resilience (security)
DTN-cookie (v1)	Low	Yes	Low	Yes	$2^{n/2} = 128$
DTN-cookie (v2)	Medium	Yes	Medium	Yes	$2^{n/2} = 128$
DTN-cookie (v3)	High	Yes	High	Yes	$2^k, 2^{n/2} = 128$

Table 2 compares the three DTN-cookie variants in terms of complexity, robustness of the mechanisms, processing requirements, power-saving and resilience to attacks. The v1 variant is the least complex while v3 is the most complex of the three variants. The three mechanisms have been designed to be adaptive to the prevailing NTL and any bundle whose DTN-cookie fails to authenticate is dropped. This makes the proposed solution very robust.

In terms of processing needs v3 is more computationally demanding than v1 and v2. HMAC has a higher energy cost than hash functions and can be as high as 96%. The three variants (v1, v2 and v3) are still more energy efficient in terms of verification efficiency when compared to digital signatures and public-key cryptography. Since DTN is an overlay network, we transfer all security processing to wireless mobile nodes (sinks) and adopt the concept of security clusters, domains and a hierarchical based model to conserve battery power.

Apart from the cryptographic properties of the proposed solution, a number of operational security measures have also been proposed. Setting the bundle size to a reasonable uniform length of 64KB is adequate for the application scenario, the bundles are easy to verify, the network is protected from memory exhaustion and waiting times are drastically reduced. Egress and ingress filtering rules at the security gateways help ensure that only bundles with legitimate EIDs and valid lifetimes are processed while illegitimate bundles are dropped.

6 Conclusions

Network DOS is a threat which can degrade network performance and the availability of DTN services. Implementing strong security does not imply that a network is attack-proof; instead it exposes it to resource exhaustion which degrades performance at resource-constrained nodes. It is therefore not advisable to use strong cryptographic algorithms for these nodes with limited resources. A more efficient approach is to begin with weak authentication which is more efficient and light-weight and gradually progress to stronger authentication mechanisms.

In this paper, we have identified resource exhaustion as a simple means by which an attacker can launch DOS in DTN. We have proposed the use of DTN-cookies for both scenarios and based on our evaluation, the proposed solution is considered lightweight, efficient, hard to forge, and incurs less overheads in terms of computation, communication, power, and bandwidth. The proposed solution is highly random since inputs such as the timestamp, nonce, and symmetric secret keys are generated through very random processes. In summary, the proposed mechanism can proactively filter out attack bundles and make the DTN resilient to DOS attacks.

As future work, we will focus on the compromised nodes problem and how to identify, isolate and mitigate their effects in the network. We will implement these designs through simulations and emulation using the ONE Simulator and DTN2 Reference Implementation (RI) respectively.

References

1. Bindra, H., Sangal, A.: Considerations and Open Issues in Delay Tolerant Network's (DTNs) Security. Wireless Sensor Network Scientific Research Journal, 635–648 (2010)
2. Raymond, D.R., Midkiff, S.F.: Denial-of-Service in Wireless Sensor Networks: Attacks and Defences. IEEE Pervasive Computing 7(1), 74–81 (2008)
3. Loukas, G., Öke, G.: Protection Against Denial of Service Attacks - A Survey. The Computer Journal 53, 1020–1037 (2010)
4. Ansa, G., Johnson, E., Cruickshank, H., Sun, Z.: Mitigating Denial of Service Attacks in Delay-and Disruption-Tolerant Networks. In: Sithamparanathan, K., Marchese, M., Ruggieri, M., Bisio, I. (eds.) Psats 2010. LNICST, vol. 43, pp. 221–234. Springer, Heidelberg (2010)
5. Dwork, C., Naor, M.: Pricing via Processing or Combating Junk Mails. Springer, Heidelberg (1998)
6. Juels, A., Brainard, J.: Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. In: Proc. Network and Distributed Systems Security Symposium, pp. 151–165 (1999)
7. Maughan, G., Schertler, M., Schneider, M., Turner, J.: Internet Security Association and Key Management Protocol (ISAKMP), RFC 2408 (1998)
8. Meadows, C.: A Formal Framework and Evaluation Method for Network Denial of Service. In: Proc. IEEE Computer Security Foundations Workshop (1999)
9. Leiwo, J., Aura, T., Nikander, P.: Towards Network Denial of Service Resistant Protocols. In: Proc. IFIP TC11 Conference Proceedings, vol. 175, pp. 301–310 (2000)
10. Farrell, S., Ramadas, M., Burleigh, S.: RFC5327: Licklider Transmission Protocol – Security Extensions Network Working Group (2008)

11. Symington, S., Farrell, S., Weiss, H., Lovell, P.: Bundle Security Protocol Specification, Draft-irtf-dtnrg-bundle-security-17 (2010)
12. Ivancic, W.D.: Security Analysis of DTN Architecture and Bundle Protocol Specification for Space-Based Networks. In: IEEE Aerospace Conference, Big Sky Montana (2010)
13. Gong, L., Syverson, P.: Fail-stop Protocols: An Approach to Designing Secure Protocols. In: Proc. of IFIP DCCA-5, Illinois (1995)
14. Krawczyk, H., Bellare, M., Canetti, R.: HMAC: Keyed-Hashing for Message Authentication. In: Crypto 1996, pp. 1–15 (1996)
15. Sterbenz, J. P.G., et al.: Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions. In: Proceedings of the 1st ACM Workshop on Wireless Security WISE 2002 (2002)
16. Yang, H., Luo, H., Ye, F., Zhang, L.: Security in Mobile Ad hoc Networks: Challenges and Solutions. IEEE Wireless Communications 11(1), 38–47 (2004)