Mitigating Denial of Service Attacks in Delay-and Disruption-Tolerant Networks

Godwin Ansa, Enyenihi Johnson, Haitham Cruickshank, and Zhili Sun

Centre for Communications Systems Research, University of Surrey
Guildford, Surrey, UK
{g.ansa,e.johnson,h.cruickshank,z.sun}@surrey.ac.uk

Abstract. There is a growing interest in providing communications to "Challenged" environments which have been hitherto isolated and disconnected due to the lack of communications infrastructure. These are regions which lie at the edge of the current Internet. Confidentiality, integrity and availability are the three major security requirements of any secured system or network. This paper presents our work on Denial of Service mitigation in Delay-and Disruption-Tolerant Networks. We propose three examples of a light-weight bundle authenticator (DTN-cookie) based on XOR and HMAC operations to thwart DoS attacks that lead to resource exhaustion.

Keywords: DTN, Denial of Service, Protocol, Security.

1 Introduction

The success of the Internet is largely due to its ability to interconnect communication devices across the world using a homogenous set of protocols, the Transmission Control Protocol/Internet Protocol (TCP/IP) suite. The present Internet is built on the assumption of availability of a continuous bidirectional link between source and destination which supports end-to-end communication, small and relatively consistent delay in sending packets and receiving the corresponding acknowledgement packets, data rates from source to destination and vice versa are symmetric, and low error rates in terms of packet loss or data corruption [1]. However, there are networks that do not conform to the above assumptions. These networks referred to as "Challenged" networks are characterized by limited bandwidth, host and router mobility, disconnection due to interference or limited battery power [2].

These highly heterogeneous networks unlike the Internet are prone to long and variable delays, high error rates, arbitrarily long periods of link disconnection and large bi-directional data-rate asymmetries [1]. To overcome these difficulties, the DTN architecture which is based on the initial work on the Interplanetary Internet [3, 4] was conceived. DTN uses a store-and-forward message-switching technique to isolate delay and move data along the communication path. DTN is a network of regional networks [1], forming an overlay architecture which operates above the existing protocol stacks found in other network architectures [5]. Its purpose is to support interoperability among underlying challenged regional networks [2]. At the DTN

K. Sithamparanathan et al. (Eds.): PSATS 2010, LNICST 43, pp. 221–234, 2010.

nodes, a new overlay layer called the "bundle layer" sits on top of the traditional transport layers to provide end-to-end data transfers among the DTN regions. The DTN infrastructure suffers from severe resource scarcity and this has prompted some restrictions to its use through some form of authentication and access control. The DTN security architecture supports hop-by-hop authentication and integrity checks. This is to ensure bundle content correctness before forwarding and is designed to authenticate DTN nodes as legitimate senders and receivers of bundles to each other [6, 7]. The end-to-end mechanism provides authentication for the user.

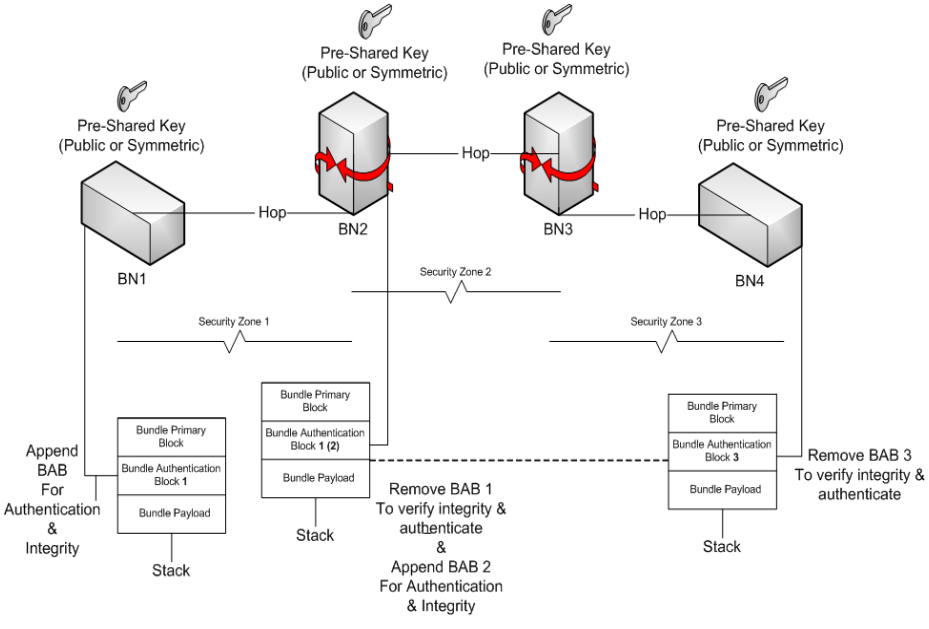

Fig. 1. DTN hop-by-hop authentication and integrity validation [8]

There are a number of key components in a DTN that provide critical services such as monitoring or query access points, data servers, cryptographic key servers, routers, security gateways, network uplinks and network nodes. These network components can come under serious DoS attacks when an attacker sends loads of requests which engage them in computationally intensive authentication protocol [9]. Currently, it is not possible to stop DoS attacks since most attacks are based on the use of protocols and services in an enormous scale. Solutions can only mitigate these attacks. Figure 1 shows the hop-by-hop authentication/integrity check using the Bundle Authentication Block (BAB). The BAB is used to assure the authenticity and integrity of the bundle along a single hop from forwarder to intermediate receiver. The communication path is divided into security zones as shown. Similarly, for end-to-end security services, the Payload Integrity Block (PIB) and Payload Confidentiality Blocks (PCB) are used. Further details on the DTN security architecture can be found in [10].

Our design seeks to provide a weak authentication phase using a unidirectional message exchange prior to signature verification. With this approach, communication cost is reduced significantly. Also our solution is power efficient and low in computational cost since the proposed DoS mitigation technique is lightweight and suitable for low-power devices like sensors.

1.2 DTN Scenario

Based on the description of a DTN, we present Figure 2 which depicts a United Nations (UN) peacekeeping scenario with three isolated regions bridged by a satellite. Region 1 and 2 are wireless sensor networks and region 3 is a satellite network and represents the UN headquarters in New York. Each region can be accessed through a security gateway (a base station in the case of regions 1 and 2, and a Network Control Centre "NCC" for region 3). The satellite network acts as a transit region (backbone in the sky) [11] for the two isolated DTN regions. In this scenario, the security gateways, sensor nodes and routers are stationary while the satellite terminals and Mobile Sink Nodes (MSNs) can be mobile. Some environmental constraints may include node failure, intentional sleep cycles, energy savings and node mobility. The patterns of connectivity can be scheduled, predicted or opportunistic.

The remainder of this paper is structured as follows. In Section 2 we highlight the threats posed by DoS attacks which prevent the DTN from fulfilling its functions. Section 3 reviews related work on DoS mitigation in terrestrial networks and the proposed techniques. Our initial design specification, the assumptions made, the

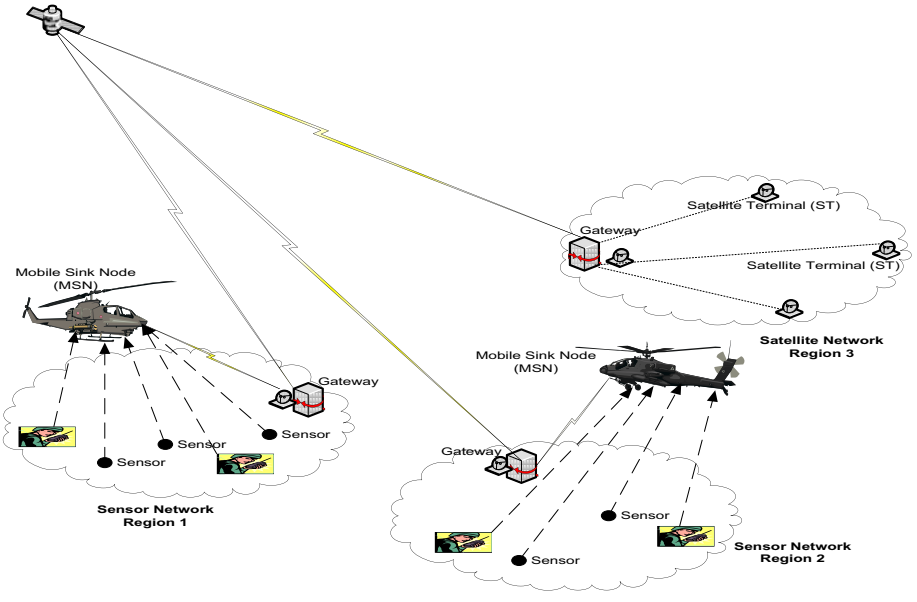

Fig. 2. A DTN-based wireless UN peacekeeping scenario

networking and security requirements are presented in Section 4. In section 5, we discuss time synchronization issues in DTN and analyze the proposed design. Section 6 concludes the paper and suggests the direction for future research.

2 Threat Analysis

Security threats can be divided into passive and active threats. A careful threat analysis of the DTN scenario depicted in Figure 2 shows that the network is susceptible to passive attacks through eavesdropping and traffic analysis. This is due to the wireless communication medium and broadcast nature of the satellite channel which allows intruders to have access to private information and the identity of the communicating entities. Also the depicted scenario is prone to active attacks such as replay attacks, masquerading, modification attacks and Denial of Service (DoS) attacks. The main focus of this paper is on DoS attacks though our design takes into consideration the active attacks mentioned above to provide a more robust solution.

Based on a classification by the Computer Emergency Response Team (CERT), there are three kinds of DoS attacks [12]:

- Destruction or alteration of configuration information
- Physical destruction or alteration of network components
- Consumption of scarce, limited, or non renewable resources

The first and second kinds of DoS attacks listed above are not dealt with in this paper. The focus of this paper is primarily on protocol or design-level vulnerabilities in the authentication process which makes it easy for attackers to launch DoS attacks on DTN security gateways which may lead to resource exhaustion. The target resources include battery power, memory, CPU time, disk space and network bandwidth.

3 Related Work

The very protocols we use to protect communication networks against unauthorized access can be used as a hook for DoS attacks by clever attackers. Any protocol where the server commits to expensive computations especially using public key cryptography or to memory allocation by storing protocol state information before or as part of client authentication is susceptible to network DoS. A number of techniques have been proposed to tackle this problem in terrestrial networks one of which is the client puzzle technique. In order to prevent junk mail, [13] proposed a technique which requires a sender to compute a cryptographic puzzle for each message. The cost of computing the puzzle is negligible for normal users but expensive for mass mailers.

Juels and Brainard [14] extended this idea and introduced client puzzles to tackle the problem of connection depletion attacks. The robustness of authentication protocols against DoS attacks can be improved by asking a client to commit its computational resources to the protocol run before the server allocates its memory and processing time. The solution to the puzzle requires a brute-force search for some bits of inverse of a one-way hash function. The difficulty of the puzzle is adjusted based on server load. The assurance of the server is increased when it establishes that the

intention of the client is good. This is achieved gradually through a series of weak authentication prior to signature verification [15].

Another solution defined by IPSec, is the Internet Security Association and Key Management Protocol (ISAKMP) which is a framework for key exchange and security associations. It is based on an anti-clogging technique which requires a client to return a server generated cookie. This is a technique derived from the PHOTURIS protocol, and can be used to prove a client's identity and is verified by the server before any costly authentication protocol is triggered [12]. A cookie as defined by [16] *"is a unique nonce computed from the names of sending and receiving parties and local secret information available only to the sender*". A specification of the ISAKMP anti-clogging technique which is based on the PHOTURIS protocol is given in [12] as follows:

- the cookie must depend on the addresses of the communicating parties
- the cookie is generated based on a local secret known only by its generator
- nobody must be able to forge the cookie that will be accepted by the server
- the cookie generation and verification must be efficient in CPU and memory consumption.
- the server must not keep per-client state until IP address has been verified.

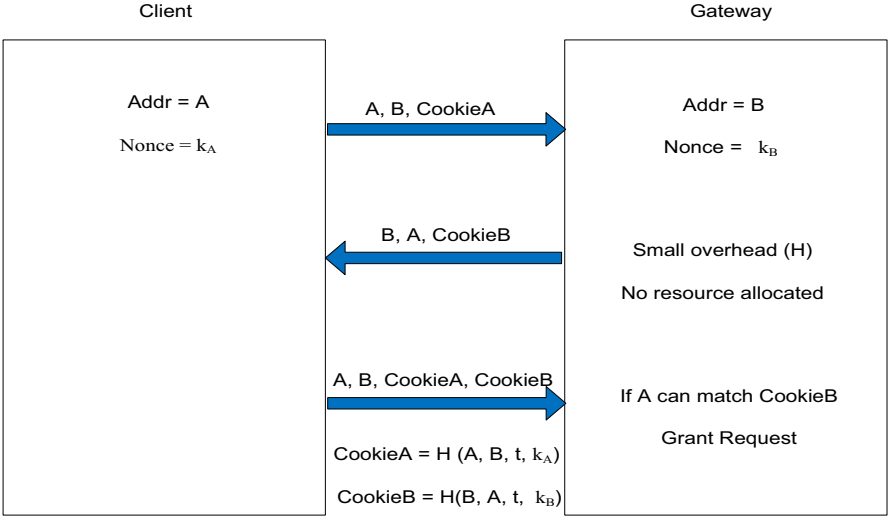

Fig. 3. The cookie anti-clogging technique [12]

PHOTURIS uses a keyed one-way hash function which uses both IP addresses, both UDP ports, and some locally-generated secret value (which is same for all clients but must be changed periodically). In some terrestrial systems, a threshold is placed on the rate of requests that a legitimate client can generate. In order to justify an increased rate from a single source, attackers emulate different source addresses by using false source addresses. The cookie exchange helps in the verification of a client's

claim of presence at a particular IP address. See [17] for more details on the ISAKMP anti-clogging technique.

4 Protocol Design for DoS-Resilience in DTN

A DTN is potentially susceptible to DoS attacks due to limited connectivity, highly variable round-trip communication time, and limited computing resources at some nodes [16]. Therefore, in order to provide security additional overhead in terms of bandwidth utilization and computational costs may be introduced on the nodes. In IP Security (IPSec), bogus traffic injected into the network is carried all the way to the security destination and this consumes valuable resources [7]. As shown in Figure 1, DTN Bundle Security Protocol (BSP) includes a hop-by-hop security feature where each bundle is checked at a DTN router and any traffic that cannot be authenticated as coming from a legitimate node is discarded.

We adopt a security architecture which is domain/region-oriented. A domain/region is made up of a security gateway and nodes that are allowed to communicate with one another and our focus will be on a single domain (Region 1). In this architecture, security gateways acting as access control points use public key cryptography to authenticate clients. During a connection request, control messages are sent to the security gateway signed using the client's private key. Protocols that use strong authentication from the very beginning can be used by an attacker to cause availability problems in the network. This is due to the fact that public key algorithms use computationally expensive methods such as exponentiation and factorization to provide security. We assume that in the interaction between the attacker and a security gateway, the attacker's primary objective is to waste the gateway's resources by interacting with it.

Due to the scarcity of resources, DTN does not allow complex security mechanisms to be deployed. Therefore, the cookie and the client puzzle techniques as proposed for terrestrial networks are not suitable for implementation in DTN environments. Both methods result in longer protocol runs due to the additional messages during the initial phase of the protocol. The client puzzle technique is especially not suitable for this environment since it forces clients to work more when the security gateway load is high. A security gateway might be attending to a high volume of legitimate requests from network nodes. Devices especially sensor nodes might not have the capability to solve the puzzles. The wireless communication medium used in the sensor networks (regions 1 and 2), and the broadcast nature of the satellite channel coupled with the roundtrip delay makes the cookie anti-clogging technique as proposed for terrestrial environments unsuitable for our proposed architecture. These DoS mitigation techniques are designed to operate in low-delay, well connected networks and may not perform well in DTN environments. In order to counter the attacks listed in section 2, our design has to fulfill a number of networking and security requirements.

4.1 Networking Requirements

The protocol should operate when no end-to-end path exists from source to destination. Also, the protocol should be able to withstand changes in scheduling and/or in contact of nodes. It should provide support for varying data rates. The protocol must

be resilient to delays and disruptions which may be in the order of minutes, hours or days. Where power-saving is a system-level requirement, the protocol should be able to run on nodes that are resource-constrained. And finally, the protocol should be able to work when faced with significant node mobility.

4.2 Security Requirements

We ensure availability by making sure that all security processing is performed by more capable nodes (DTN-aware nodes e.g. MSNs). We also ensure data freshness by preventing the replay of old messages through the use of timestamps, sequence numbers and nonces. All message requests must be authenticated by the security gateways in order to verify the authenticity of the entities and validity of the data source. Lastly we ensure the integrity of message requests received in order to guarantee that the contents have not been modified or deleted. The DoS mitigation technique proposed in this paper addresses these requirements.

5 Design Specification

In this section we describe the assumptions made during our design, and provide a detailed specification of the design including the notations used. We discuss issues relating to timestamps and time-synchronization in DTN and also provide an analysis of the design.

5.1 Design Assumptions

- a security gateway has bounded resources that could possibly be exhausted by a clever attacker.
- the attacker may or may not have bounded resources and may or may not have resources greater than the security gateway.
- the attacker is assumed to have the ability to replay, modify, transmit, receive, and execute the protocol.
- the attacker is within wireless communication range to the security gateway and does not need the help of other nodes to launch an attack.
- trust is established during initial registration of a node with the security gateway.
- only DTN-aware MSNs and Satellite Terminals (STs) can interact with the security gateway.
- the public/private keys are generated and pre-shared during initialization
- only nodes that are registered with the security gateway and share a secret (which is used to generate the nonce) with it can generate a valid request.
- the protocol is between two communicating parties i.e. the client (MSN/ST) and security gateway.

The client always initiates the communication between a client and a security gateway. Figure 4 shows the generic bundle (message) format exchanged between a client and a security gateway.

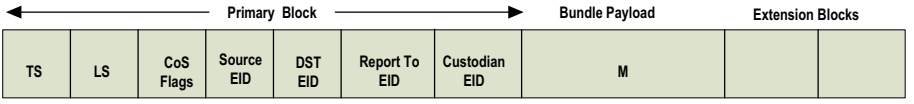

Fig. 4. Generic bundle format

In Figure 5, additional security extension blocks such as the Bundle Authentication Block (BAB) or the Payload Integrity Block (PIB) are added to the generic bundle as shown in Figure 4 to provide authentication or integrity checks respectively. Also a DTN-cookie is added as a last block to the bundle to provide a weak authentication phase during the verification process. This is our contribution to satisfy the requirement that control or data bundles within a domain/region do not trigger any form of DoS attack which might exhaust the resources of DTN nodes.

Fig. 5. A bundle with additional security extension blocks

Notations. We use the following notations in the description of our design.

- 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 EID.
- LS: the bundle lifespan or expiration time. The LS value of a bundle equals X where X can be in days, hours or minutes.
- Source_EID: the identity of each DTN-aware node, we assume that each EID is a singleton.
- DST_EID: the identity of the destination node i.e. the entity for which the bundle is destined (security gateway).
- RSAwithSHA256: represents the ciphersuite and gives an idea to what security block is in use.
- M: the payload of the request message.
- 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 algorithm.
- h (M): the hash value or message digest derived by passing the payload M through the function h.
- $pubK_{Xi}$: the public key of node X_i
- $privK_{Xi}$: the private key of node X_i

DoS attacks are not addressed fully in the current security extensions of the Bundle Protocol. At present, DTN nodes simply discard any traffic that fails the authentication and access control checks. This in itself provides minimal protection and makes it harder to launch a DoS attack. However, the potential danger is in the computation

and storage overheads at DTN nodes which we seek to address. We propose three examples of a lightweight authenticator which we refer to as DTN-cookie each having varying levels of complexity. The choice of a particular type of DTN-cookie will depend on the level of DoS threat. The DTN-cookie is calculated by the bundle originator and attached as the last block of the bundle as shown in Figure 5 and must be verified by the security gateway.

(a) DTN-cookie is a nonce which represents the result when a random number generator is seeded using an Initialization Vector (IV). The IV is known only to all registered nodes in the domain/region during the initialization phase.
Nonce = RNG (IV or seed). The Initialization Vector is kept secret and changed periodically to ensure freshness.
(b) DTN-cookie is the result of an XOR operation on the timestamp and the result of the operation in (a) i.e. timestamp (TS) $\oplus$ nonce in (a).
(c) DTN-cookie is derived by passing the result of the operation in (b) through a keyed Message Authentication Code such as HMAC. The underlying hash function is SHA-1 and the secret key is shared between the client and the gateway. This is represented as follows:
DTN-cookie = HMAC ((timestamp (TS) $\oplus$ nonce) + secret key).

A secure Key Management mechanism is required for the safe generation and distribution of keys. Also, regular key refreshments are very important to thwart attacks which might attempt to use compromised keys.

5.2 Timestamps and Time Synchronization in DTN

Timestamps are typically used for logging events. The Bundle Protocol uses timestamps for three reasons to guarantee network efficiency and resource protection [3]:

- the useful life indicator of bundles prevents the network from storing and forwarding bundles whose useful lifespan have expired.
- the lifetime helps prevent bundles from looping continuously in the network as a result of routing loops.
- the originating timestamp helps in the unique identification of bundles and the re-assembly of bundle fragments.

The original purpose of timestamps is to help DTN intermediary or destination nodes to determine when it is appropriate to drop a bundle or determine the age of a bundle. In our design, timestamps are used to provide an accurate record of the bundle creation time and act as a freshness identifier to protect against replay attacks. Timestamps provide a more efficient way to test for freshness without introducing too much traffic into the network. This replaces the idea of the gateway broadcasting or sending a nonce as a freshness identifier at certain time intervals. Based on RFC 5050 the creation timestamp is a concatenation of the bundle creation time and a monotonically increasing sequence number. The sequence number is incremented continuously (say every second) in order to make the concatenation of source EID and creation timestamp the unique bundle identifier. With this bundles created at the same time interval by different nodes can be differentiated.

The Bundle Protocol has an in-built assumption that DTN nodes have a basic time-synchronization capability with a common, synchronized view of the Coordinated Universal Time (UTC) [3, 6]. This is based on DTN's initial design for deep space environments where the space-based agents are synchronized and highly scheduled with bandwidth and delays known in advance. This assumption becomes invalid when extended to terrestrial DTN environments due to the fact that DTN nodes are isolated or disconnected most of the time, may use discovery methods or might implement intentional sleep cycles to conserve power as in wireless sensor networks. In these different instances the clocks of such nodes can drift. It has been noted that in challenged environments like DTN, time synchronization is a problem.

The use of classical time protocols like the Network Time Protocol (NTP) is limited because it is not resilient to disruption. A very good source for UTC is the Global Positioning System (GPS) which is one non-DTN time solution. DTN nodes cannot rely on GPS due to the fact that most nodes such as sensors may lack GPS receivers and the cost implication of using GPS as a reference clock maybe too high making deployment infeasible. DTN has the in-built capability to read the local system clock values and determine the correct time. This is the reason why it possible for DTN nodes to set the creation time when bundles are created and sent into the network and checked for expiry on reception.

At present, there is on-going debate in the DTN research community on the justification for DTN reliance on time synchronization to perform store-and-forward networking as specified in RFC 5050. The use of absolute timestamps and coordinated clocks in DTN to meet the requirements stated earlier has been criticized or supported by many. There are suggestions to either strengthen or replace the lifetime mechanism with a countdown counter such as a "hops-to-live". Hops-to-live explicitly discards bundles that have traversed more than a set number of hops. This will allow bundles to expire when their lifetime is decremented to zero without any requirement for loose synchronization.

5.3 Analysis of Design

As a requirement, our design should be secure against masquerade by providing a mechanism to identify and discard any attack bundle on initial processing of the DTN-cookie. Also the DTN-cookie should be lightweight, efficient and simple. The DTN-cookie is the result of an XOR operation on the timestamp and a nonce generated by the client. The nonce generated by the client should match that calculated by the security gateway. The design should discourage Transport Layer Security (TLS) type of handshake but use minimized number of roundtrips i.e. only a single unidirectional message for connection establishment or access. DTN-aware nodes act as security-sources for sensor nodes that maybe IDless or have resource restrictions. We add a DTN-cookie to every bundle for quick identification of attack bundles. In our design, computational efficiency is achieved since no gateway resource is dedicated to service any attack bundle except those required to identify and discard the traffic. A client wishing to communicate with the security gateway has to generate a message request in the form of a bundle as shown in Figure 5 and send to the security gateway.

The pseudo code below shows the operation that takes place at the security gateway during the weak authentication of a bundle.

Client ⟶ Server:

request = {TS, LS, Source_EID, DST_EID, M, h (M), priv (h (M)), DTN-cookie}

Set Bundle Lifespan (Ls) = X (days, hours or minutes)

The Server performs an XOR operation:
result = timestamp ⊕ DTN-cookie

```
if   (result ≠ server_cookie) || (bundle_in_network > Ls) ||
       (DST_EID ≠ serverAddress)
            Discard Bundle
  else
          (retrieve public key credentials from cache)
          calculate pubK_Xi (privK_Xi (h (M)))
   if  pubK_Xi (privK_Xi (h (M))) ≠ request.h (M)
                  Discard Bundle
            else
                  calculate h (M)
                  if    h (M) ≠ request.h (M)
                        Discard Bundle
                   else
                         Accept Bundle
```

One component of the security gateway is a multiple socket server with the capability to run multiple threads simultaneously and can service more than one connection request at a time. The security gateway on receiving a bundle, evaluates it to determine if it is from a legitimate client (DTN-aware node). Every legitimate entity runs a random number algorithm. This algorithm is seeded with an Initialization Vector which the security gateway shares with all DTN-aware nodes during initialization. The security gateway also shares a secret key pair with every DTN-aware node within its domain. The value of the Initialization Vector or the secret key is only known to legitimate DTN-aware nodes. These values are prerequisites for the computation of a valid nonce or DTN-cookie. In our design specification, it is impossible to generate a valid bundle that will be accepted by the security gateway without having the capability to compute a valid nonce or DTN-cookie. Every bundle sent by an attacker is discarded during the weak authentication phase when the DTN-cookie is evaluated. Placing the DTN-cookie and signature at the end of the bundle allows memory-constrained nodes like sensors to be able to process the bundle and verify its security result.

With a single unidirectional message request, the security mechanism is able to identify attack bundles and discard them. Replay attacks aimed at modifying the timestamp field will invalidate the bundle during the weak authentication of the DTN-cookie. Also, attacks that are aimed towards substituting the payload of a legitimate message will be thwarted since any modification or tampering with the payload will be

detected during the signature verification phase. If a bundle passes the weak authentication phase, a computing intensive signature verification phase will be triggered. The cost of an attack will be more on the side of the attacker since at every attempt he has to compute a nonce, generate a fake DTN-cookie, sign the bundle and send. The gateway is not affected since any client request that cannot pass the weak authentication phase is discarded and the connection closed. This is to safeguard the gateway's resources. If the signature verification phase succeeds, then the client is authenticated, request granted and resource allocated based on security gateway policy.

From our evaluation of the protocol's performance, it is clear that the protocol follows the description of Gong and Syverson's model of fail-stop protocols [18] but different because it does not use strong authentication from the very beginning. It also follows the framework proposed by Meadows [9] where the server's assurance of the client's intention is increased at every step of the protocol execution. This is achieved by introducing a weak authentication phase prior to the more expensive signature verification phase.

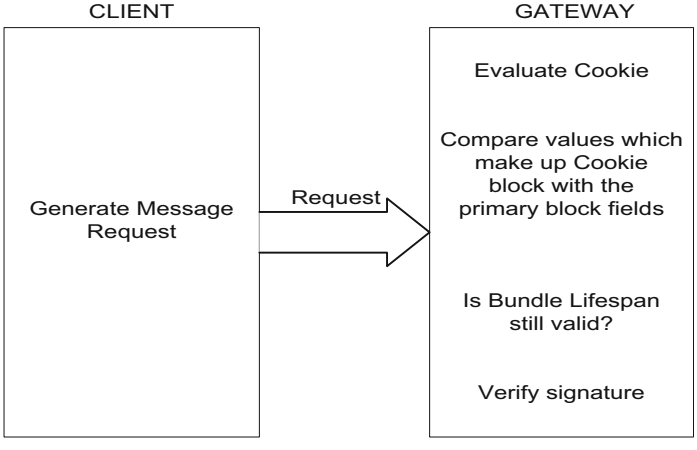

Fig. 6. The DTN-cookie technique

The gain in our design is that we have been able to reduce the number of message exchanges in the weak authentication phase from three (as in the PHOTURIS protocol) to one unidirectional message request thereby saving a lot of bandwidth. Also our design has achieved its objective of making the bundle authenticator (i.e. DTN-cookie) light-weight and efficient by using simple techniques like XOR and HMAC operations. Through these methods, we have been able to provide weak authentication for bundles with less overheads in the computation cost and power. In order to provide maximum protection, the HMAC variant of DTN-cookie proposed in this paper can be used. It is faster to hash the timestamp, nonce and key than hashing the entire bundle making it computationally more efficient. Also the inputs to the MAC algorithm [19] (i.e. the timestamp, nonce and key) have a very high degree of randomness as shown in their method of generation. It is hard to forge but inexpensive to verify which makes it appropriate for nodes with very low power budget. However, this comes with a slightly higher price of overheads due to the additional security processing.

6 Conclusion

DoS attacks are becoming a great threat to the proper functioning and survivability of networks. From section 3, it is obvious that traditional DoS mitigation methods such as the cookie anti-clogging technique and client puzzle used in terrestrial networks are not suitable for DTN environments. These methods require the exchange of several messages to achieve weak authentication or solving server generated puzzles respectively.

In our work we propose the use of a security extension block (DTN-cookie) which is added to every bundle to provide weak authentication. The weak authentication phase precedes a computationally intensive signature verification phase. The lightweight authenticator comes in three variants which is applied based on the perceived level of DoS threat. We have implemented the XOR variant of the DTN-cookie which provides moderate protection for DTN security gateways. This is to protect against resource exhaustion and ensure availability of DTN services. As future work, these mechanisms will be tested using the DTN2 Reference Implementation. We will analyze the protocol's performance against metrics such as communication cost, computation cost and power efficiency. We will broaden our scope to cover a more hierarchical hop-by-hop DoS resilient mechanism and also consider inter-domain DoS threats in DTN.

Acknowledgements

We will like to thank the EU Information Society Technologies SATNEx II Network of Excellence VIII for supporting this research work.

References

1. Warthman, F.: Delay Tolerant Networks (DTNs): A tutorial.v1.1 (2003)
2. Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets. In: ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 27–34 (2003)
3. Wood, L., Eddy, W., Holliday, P.: A Bundle of Problems. In: IEEE Aerospace Conference, Big Sky Montana (2009)
4. Farrell, S., Cahill, V., Geraghty, D., Humphreys, I., MacDonald, P.: When TCP Breaks: Delay-and Disruption-Tolerant Networking. IEEE Internet Computing 10(4), 72–78 (2006)
5. Fall, K.: A Message-Switched Architecture for Challenged Internets. Intel Research Berkeley. IRB-TR-02-010 (2002)
6. Cerf, V., et al.: Delay-Tolerant Networking Architecture. RFC 4838, Network Working Group (2007)
7. Cerf, V.G.: An Interplanetary Internet. Space Operations Communicator 5(4) (2008)
8. Bhutta, N., Ansa, G., Johnson, E., Ahmad, N., Alsiyabi, M., Cruickshank, H.: Security Analysis for Delay/Disruption Satellite and Sensor Networks. In: IWSSC 2009, Siena Italy (2009)
9. Meadows, C.: A Formal Framework and Evaluation Method for Network Denial of Service. In: Proc. IEEE Computer Security Foundations Workshop (1999)

10. Farrell, S., et al.: Delay Tolerant Networking Security Overview. DTN Research Group, Internet Draft (draft-irtf-dtnrg-sec-overview-06) (2009)
11. Franck, L.: Delay Tolerant Networking with Satellites: Overview and Research Directions. In: COST272 - 7th MCM, Telecom Paris (2004)
12. Onen, M., Molva, R.: Denial of Service Prevention in Satellite Networks. In: IEEE International Conference on Communications, vol. 7, pp. 4387–4391 (2004)
13. Dwork, C., Naor, M.: Pricing via Processing or Combating Junk Mails. Springer, Heidelberg (1998)
14. 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)
15. Aura, T., Nikander, P., Leiwo, J.: DoS-resistant Authentication with Client Puzzles. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2000. LNCS, vol. 2133, pp. 178–181. Springer, Heidelberg (2001)
16. Feng, Q., Lutz, R.: Assessing the Effect of Software Failures on Trust Assumptions. In: 19th Int'l Symposium on Software Reliability Engineering, pp. 291–292 (2008)
17. Arkinson, R.: Security Architecture for the Internet Protocol. RFC 1825 (1995)
18. Gong, L., Syverson, P.: Fail-stop Protocols: An Approach to Designing Secure Protocols. In: Proc. of IFIP DCCA-5, Illinois (1995)
19. Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)