Delay and Disruption Tolerant Authentication for Space Communications

Susanna Spinsante, *Senior Member, IEEE*

***Abstract*—Some space communication scenarios, such as Deep Space communication networks, represent an example of Delay and Disruption Tolerant Networks, which may experience dynamic, long-delay links, and outages. Networks of this kind require a strong re-engineering of many of the protocols for data transmission usually adopted in traditional, terrestrial data networks. The Bundle Protocol has been proposed by the IETF as an overlay communication infrastructure, to cope with the heterogeneous components of a Disruption Tolerant Network; however, there are still many open issues that need to be analyzed. This paper focuses on the impact of delay and disruption tolerant networks on the efficiency and robustness of authentication mechanisms, and discusses some solutions possibly suitable to the Bundle architecture.**

***Index Terms*—Authentication, Bundle Protocol, Delay and Disruption Tolerant Network**

I. Introduction

THE rapid and outstanding advances in space technology are enabling to push the boundaries of human space exploration further afield. As a consequence, the vision of future space exploration includes missions to deep space, that require the availability of communication links among planets, satellites, spacecrafts and crewed vehicles. InterPlaNetary (IPN) Internet has consequently become the widely accepted paradigm in the design and development of deep space networks, as the Internet of the deep space planetary networks.

Studies and research activities about the IPN have been developed since several years. The fundamental paper by Akyildiz et al. [1] outlines a number of research challenges about the design of the IPN, which are still under investigation. As a matter of fact, the peculiarities of deep space communication scenarios require a re-thinking of many of the basic concepts Internet-based protocols rely upon.

Among them, we may cite extremely long and variable propagation delays, asymmetrical forward and reverse link capacities, high link error rates for Radio Frequency communication channels, intermittent link connectivity, and the effects of planetary distances on the signal strength and the protocol design. Moreover, differentiated data are to be carried over IPN communication links, such as time-insensitive scientific data (collected from planets and moons, for example), time-sensitive scientific data (usually, multimedia data about the local environment, that are to be sent to Earth for control purposes), mission status Telemetry (that requires periodic, or event-driven, transmission services), command and control data (that require a closed-loop among the space elements to be controlled and the control sites on Earth).

Space communication networks also represent an example of Delay and Disruption Tolerant Networks (DTN) [2,3], i.e. networks which may experience dynamic links and outages, besides long-delay links. Networks of this type are suited to applications that are mostly asynchronous and insensitive to large variations in delivery conditions. DTNs require a strong re-engineering of many of the protocols for data transmission usually adopted in traditional IP networks, as they differ from terrestrial networks in their characteristics and connectivity. Link, Network, and Transport protocols need to be carefully considered and chosen, to cope with the peculiarities of DTNs.

Among the protocols specifically proposed for adoption in DTNs, the so called Bundle Protocol [4] has been designed to meet the requirements of many different types of DTNs, including networks aimed at supporting deep space exploration. A *bundle* consists of a number of concatenated blocks, including some common shared metadata (the bundle header, or primary bundle block), followed by a number of other payload blocks. The Bundle Protocol is intended to provide a common format for store-and-forward networking messages, assuming that a storage capability in bundle agents located inside the network may help in overcoming many of the challenges specifically characterizing a DTN.

Many open issues still remain within the definition of the Bundle Protocol; among them, addressing and forwarding strategies, Quality of Service mechanisms, network management and monitoring protocols, security mechanisms, and means for key exchange and establishment of security associations. Optional security extensions to the Bundle Protocol have been actually proposed in [5], however related to the need of providing a common mean for data integrity and error checking, thus making it not possible to distinguish between check failures due to errors, or security attacks.

The IPN, as any other possible DTN, results from the combination and overlay of several, usually heterogeneous, subsystems. As a consequence, the different components that contribute in establishing the IPN have their own architectures, with different sets of protocols that best fit the communication environment. In the peculiar context of space networks, a reference architecture is represented by the space/ground protocol stack defined by the Consultative Committee for Space Data Systems (CCSDS), namely the set of Space Communications Protocol Standards (SCPS) [6], which is a suite of four Recommendations, parallel in function to, and interoperable with, the protocol stack of the Earth-based Internet (typically FTP/TCP/IP). The SCPS protocol stack consists of eight layers; among them, a layer is specifically foreseen to provide protection against attacks on the flow of user data, in order to ensure space End-to-End security. Security issues in the framework of the CCSDS protocols have been examined and discussed in previous papers, such as [7,8]; this paper, instead, focuses on the security evaluation and analysis of the Bundle Protocol, and on the proposal of possible authentication mechanisms suited to the bundle architecture.

The paper is organized as follows: Section II presents a detailed review of the authentication solutions currently proposed within the Bundle Protocol, and outlines their limitations; Section III discusses the possible adoption of the Galois Message Authentication Code (GMAC) scheme within the Bundle Protocol, to cope with its peculiar features and requirements, and to provide the possibility of ensuring authentication and confidentiality through a single security primitive; Section IV provides preliminary evaluations of the proposed scheme, with reference to the Bundle Protocol; finally, Section V concludes the paper.

II. Authentication procedures within the Bundle Protocol

A DTN may be conceived as an overlay network built on top of lower layer networks, which may vary from node to node. This heterogeneous foundation may place severe limitations on the network performance, such as intermittent loss of connectivity, long or variable delays, asymmetric data rates, or high error rates. As a consequence, a DTN protocol should be able to support interoperability across such potentially stressed lower layer networks.

Following this paradigm, the Bundle Protocol proposed for DTNs is layered on top of a “convergence layer”, which is itself on top of other lower layers. The DTN Bundle Protocol describes the format of the messages, called bundles, passed between DTN bundle agents that participate in bundle communications, to form the DTN store-and-forward overlay network. The Bundle Security Protocol [5] extends the scope of the Bundle Protocol to provide support for data integrity and confidentiality services, in order to counter the possible security threats identified in a DTN. Among them, we may cite non-DTN node threats, i.e. security threats generated from network elements which are not directly part of the DTN; resource consumption, due to unauthorized access and use of DTN infrastructure resources; Denial of Service attacks; traffic storms due to manipulation of bundle content, and general threats against confidentiality and integrity.

The stressed environment of the underlying networks over which the bundle protocol has to operate makes it important to protect the DTN from unauthorized use; at the same time, this stressed environment presents unique challenges on the mechanisms needed to secure the bundle protocol. Furthermore, a portion of a DTN may be deployed in environments where it could get compromised, so that the usual security challenges related to confidentiality, integrity and availability, still hold.

Authentication services applied to check and endorse a DTN node genuineness may help in avoiding unauthorized access and use of DTN resources, such as unauthorized applications controlling the DTN infrastructure, or authorized applications sending bundles at a rate, or class of service, for which they lack permission, and unauthorized bundle content modification. Moreover, DTN nodes could be involved in resource consuming behaviors, such as forwarding bundles that were not sent by authorized DTN nodes, generating reports not originally requested, and not detecting unplanned replays or other misbehaviors. If an effective mean to authenticate legitimate DTN nodes is provided, this may help in counteracting all these potential threats to the DTN resources, security, and efficiency.

A. Bundle Fragmentation

As for the case of packets fragmentation in traditional IP networks, fragmentation of bundles is an issue debated for a long time. Fragmentation is basically motivated by the need of adapting relatively large bundles for transport by protocols with limitations on message size. Fragmentation may play a fundamental role in DTNs, where the possibility of routing a bundle (called contact) is related to the storage capability of a node, given by the product between the available bandwidth, and a time window of opportunity to use it. Bundle fragmentation, however, is one of the most challenging issues in DTNs: it may work well for some scenarios, but it may be useless for others.

Fragmentation in DTNs can be classified as proactive or reactive. The former can be defined as the process performed by a node, which has an entire bundle, to break it into smaller pieces; the latter is usually needed to optimize retransmission after a connection failure of some kind. Reactive fragmentation assumes some level of interaction between the sender and the receiver, so that the sender can restart transmission from the point of failure. By this way, even very large bundles can be sent across intermittent or episodic links, piece by piece, and the fragments may be reassembled later.

A bundle having a payload of size M bytes can be replaced by two fragments, i.e. new bundles, with the same source endpoint identifier (ID) and creation timestamp as the original bundle, and payloads comprising the first N, and the last (M - N) bytes of the original bundle's payload, where $0 < N < M$. Fragments may be fragmented on their turn, so that fragmentation may in effect replace the original bundle with more than two fragments. However, only one level of fragmentation is admitted, as in IP networks. The concatenation of the payloads of all fragments produced by fragmentation must always be identical to the payload of the original, fragmented bundle. The payloads of fragments resulting from different fragmentation episodes, in different parts of the network, could be overlapping subsets of the original bundle's payload.

Reassembly of application data units from fragments occurs at destination endpoints as necessary; an application data unit may also be reassembled at some other nodes on the route to the destination.

B. Interactions between security and fragmentation

Proactive fragmentation is reasonably interoperable with security processing, but reactive fragmentation may be troublesome. As an example, fragments transferred over a link that undergoes a failure and cannot be recovered, cannot be integrity checked, since the remaining data necessary to compute the integrity check value are missing; as a consequence, forwarding the fragments as a bundle could generate a security leak. In the example situation, once the link is recovered, the receiving node might request the sender to create and send a signature for the amount of data already received, which would be faster than a complete retransmission of the bundle. By this way, the first fragment with its integrity check could be forwarded; the original sender could then create another fragment-bundle containing the remainder of the initial bundle data. This approach could solve the issue of ensuring integrity validation of the bundle fragments, however it relies on the possibility of a strict coordination between the sender and the receiver. Unfortunately this cannot be ensured in a DTN, where long link outages between nodes may result in connections that are more similar to a one-way link, than a two-ways one.

An alternative solution may be conceived, by associating not a single checksum with the bundle, but a number of checksums, one for every given amount of data included in the bundle. By this way, several checksums are used to provide end-to-end integrity, and a reactively forwarded fragment may be integrity checked if it carries all the checksums corresponding to the amount of data included in the fragment. Unfortunately, this solution comes at the expense of additional computational complexity at each node, and additional bytes of overhead transmitted over any available link. This scheme, where each checksum protects a part of the payload, needs the definition of proper ciphersuites in the security protocol specification, in a way similar to the traditional Transport Layer Security (TLS) protocol, with the relevant difference that, in general, DTNs cannot support the use of the TLS handshake protocol, as used in the traditional, terrestrial Internet.

An additional problem about security in DTNs deserves investigation: various operations performed on the bundle payload may affect its features (for example, block cipher encryption may alter the payload length), thus creating ambiguity for custody-transfer and fragment reassembly.

III. GMAC FOR BUNDLE AUTHENTICATION

Among the security mechanisms that may be applied to data and information to ensure their authentication and confidentiality, Authenticated Encryption with Associated Data (AEAD) techniques [9] can generate Message Authentication Codes (MACs), and provide encryption of the input data, at the same time: this may be a valuable feature, in the perspective of a possible need for both bundle authentication and encryption.

AEAD techniques can avoid static associated data processing, without affecting robustness and efficiency of the process, and may be applied by using a single key. In particular, they can provide a variable length authentication tag: while classical authentication schemes, such as HMAC, can generate only fixed length digests (determined by the hash function used), AEAD modes can tune the length of the authentication code, according to the amount of data to be transmitted. Shorter tags could be applied to shorter data units, thus reducing the authentication overhead and still maintaining the system robustness.

AEAD modes are shared-key encryption schemes, in which the underlying encryption algorithm takes a key, a plaintext, and a nonce, and returns a ciphertext. The decryption algorithm takes a key, a ciphertext, and a nonce, and it returns either a plaintext or a special symbol, namely *Invalid*. The definition of Authenticated-Encryption with Associated Data is related to the fact that often it is unnecessary for all the data to be ciphered, or privacy-protected: data, like a packet header, which are only authenticated but not encrypted, are called associated data. An AEAD scheme is obtained by appropriately combining an encryption scheme and a MAC, but with the goal of using a single key, and requiring a computational cost significantly lower than the cost due to encrypt, plus the cost to MAC.

Among the AEAD modes that could be suitably applied to DTNs, the Galois Counter mode of operation (GCM) [10] is considered in this paper.

GCM is a counter mode providing authenticated encryption based on universal hashing over a binary Galois field. The encryption operation has four binary inputs: a secret key K of length appropriate to the underlying block cipher, an Initialization Vector (IV) that can have any number of bits between 1 and 2^{64}, a message M of length varying between 0 and 2^{39}-256 bits, and additional authenticated data, denoted as A, of length between 0 and 2^{64} bits. Two outputs are generated: a ciphertext C of the exact length of M, and an authentication tag T, whose length τ may vary between 64 and 128 bits. The additional authenticated data A are used to protect the information that needs to be authenticated, but not encrypted. Examples for A in a traditional network environment are addresses, ports, sequence numbers. GCM decryption has five inputs: K, IV, C, A, and T, and a single output, either the message M or a special symbol, *Fail*, indicating that the inputs are not authentic (i.e. the inputs were not created by the encryption operation with the same key used for authentication).

GCM accepts *IV*s of arbitrary length, which makes it easier for applications to meet the requirement that all *IV*s must be distinct, as a nonce of any size can be used as the IV. Actually, an IV of 96 bits is recommended, for a more efficient GCM processing.

The strength of the GCM authentication of M, IV, and A is determined by τ. The value of τ must be fixed for any fixed value of the key K, and must be $\tau \geq 64$. If possible, a value of 128 bits is recommended; when $|IV| \neq 96$, a tag length of 128 bits is mandatory, for a fixed key. There is no need to pad the input message, since any message length is admitted.

An opponent can try to forge a generic τ bit MAC by choosing it at random; his attack will succeed with probability $2^{-\tau}$, or at most $2^{-\tau/2}$, according to the birthday paradox [12]. If GCM is used, the success probability of such an attack equals $(B+1)\cdot 2^{-\tau}$, where B is the number of 128 bit blocks in the message and the additional authenticated data. The effective tag strength for GCM is consequently about (τ - log$B$) bits.

A. GCM Incremental Authentication

When there is no data to encrypt, GCM can act as a stand-alone MAC (known as GMAC), authenticating messages without any modifications in the algorithm. Further, it can work as an incremental MAC: given a message M and a corresponding tag T = GMAC(K, IV, M), it is possible to efficiently compute the tag T' for a new message M', with a computational effort proportional to the Hamming distance between M and M' (i.e. the Hamming weight of $M \oplus M'$). This peculiar property of GMAC, that is unique among all the AEAD modes, is inherited from GHASH.

Function GHASH is defined by GHASH(H, A, C) = X_{m+n+1}, where H = E(K, 0^w) is the hash key derived from the GMAC key K (w is the block length, in bits, required by the encryption algorithm E(.)). Integers m and n are related to the encryption algorithm block length w: n and u denote the unique pair of positive integers such that the total number of bits in the plaintext is $(n\text{-}1)w + u$, with $1 \le u \le w$, whereas m and v denote the unique pair of positive integers such that the total number of bits in A is $(m\text{-}1)w + v$, and $1 \le v \le w$.

The variables X_i, for $i = 0,\ldots, m+n+1$, depend on H, and blocks of A and C (see [11] for details):

$$X_i = \begin{cases} 0 & i = 0 \\ (X_{i-1} \oplus A_i) \cdot H & i = 1,\ldots, m-1 \\ (X_{m-1} \oplus (A_m^* \,||\, 0^{w-v})) \cdot H & i = m \\ (X_{i-1} \oplus (C_{i-m})) \cdot H & i = m+1,\ldots, m+n-1 \\ (X_{m+n-1} \oplus (C_n^* \,||\, 0^{w-u})) \cdot H & i = m+n \\ (X_{m+n} \oplus (len(A) \,||\, len(C))) \cdot H & i = m+n+1 \end{cases}$$

where || denotes string concatenation, $len(.)$ is a function that returns a w/2-bit string containing the nonnegative integer describing the number of bits in its argument, with the least significant bit on the right, and A^*_m, C^*_n denote partial blocks taken from A and C bit strings, respectively.

Function GMAC may be decomposed into two lower-level functions:

$$\text{GMAC}(K,IV,M) = \text{GPRF}(K,IV) \oplus \text{GHASH}(H,M,\{\}) \quad (1)$$

where GPRF is the pseudorandom function used to encrypt the output of the hash function. GHASH has a number of algebraic properties that make it suitable to the bundle environment: it is linear in terms of its arguments, provided they have the same length, and it is also possible to efficiently compute the value of GHASH applied to one string appended to another string, given the GHASH values of each string, if some alignment restrictions are met.

An interesting property holds when the data that is authenticated is formatted as a sequence of fixed-length blocks $A = B_1, B_2, \ldots, B_l$ where each block is wq bits long, for some value of q, and $l = \lceil len(A) / wq \rceil$. In this case, the following result may be demonstrated:

$$\begin{aligned} &GHASH(H, B_1 \,||\, B_2 \,||\, \ldots B_l, \{\}) = \\ &H \cdot \left((len(A) \,||\, 0^{w/2}) \oplus \bigoplus_{i=0,l-1} H^{iq} \cdot h(B_i) \right) \end{aligned} \quad (2)$$

where the function $h(.)$ is a degree q polynomial in H. If one of the input data blocks changes from B_j to B'_j, the new, entire GHASH value can be computed by adding the term $H^{qj} \cdot h(B_j \oplus B'_j)$ to the value previously computed. In its turn, the value of H^{qj} can be efficiently computed through a repeated square-and-multiply algorithm, which requires no more than $\lceil j \rceil$ squarings and multiplies in GF(2^w).

Finally, GMAC supports incremental tag generation for several different types of message edits, such as changes within a fixed length message, appending or prepending data to a message, truncating data from the start or the end of a message. The linearity property of GHASH can be exploited to reduce the computational load required in each situation.

IV. Performance evaluations

In this section we try to discuss some properties of the GMAC solution that can make it more suitable to the Bundle context, with respect to classical approaches such as HMAC [13] with SHA-1 [14] (as suggested by the Bundle Security Protocol itself), or the CBC-MAC with Advanced Encryption Standard (AES), as adopted in wireless 802.11i networks.

In the case of HMAC with SHA-1, it is well known that the MAC computation is performed according to the following relation:

$$\begin{aligned} &HMAC(K, IV, M) = sha1((K \oplus opad) \,|| \\ &(sha1((K \oplus ipad) \,||\, M))) \end{aligned} \quad (3)$$

where $sha1(.)$ represents the SHA-1(.) hash function applied to the provided inputs, and the output authentication tag has a fixed length of 160 bits, i.e. 20 bytes. The strings $opad$ and $ipad$ represent two specific binary patterns used to pad the outer and inner data, respectively.

If CBC-MAC (Cipher Block Chaining mode) with AES is applied, the output authentication tag has a length of 128 bits, i.e. the length of a single AES block, and the following relations hold:

$$\begin{aligned} &CBC(K, M) = C_m \\ &C_i = AES(K, C_{i-1} \oplus M_i), i > 0 \\ &C_0 = 0^{128} \end{aligned} \quad (4)$$

where M is the whole message to authenticate, M_i denotes the i-th message block of length 128 bits, and $m = \lceil len(M)/128 \rceil$. The above equations state that it is not possible to compute C_i until C_{i-1} has not been computed, due to the chaining mechanism introduced by the CBC mode. As a consequence, CBC-MAC verification is not possible at the receiver, if one or even more blocks C_i are missing.

Now, let us assume that a given file F is to be transferred from bundle agent BA1 to bundle agent BA2; given the big amount of data in F, it is fragmented into a number of Fragment Bundles B_i, so that it is possible to write: $F = B_1||B_2||\ldots||B_{10}$, i.e. in our scenario the file can be fragmented into 10 Fragment Bundles. Let us further assume that the link between BA1 and BA2 becomes unavailable when only 8 out of 10 Fragment Bundles have been transferred between the agents. What shall we say about authentication issues in this possible scenario?

First, in order to increase the probability of successfully transferring and verifying the whole file authenticity, its global authentication tag T is included in each Fragment Bundle transmitted over the link. This obviously implies a transmission overhead, which amounts to 128 bits per fragment, in the case of AES GMAC and AES CBC-MAC, and 160 bits per fragment, in the case of HMAC with SHA-1. In the example scenario we are considering, the amount of this overhead is not significant, as it will result into 1280 bits, or 1600 bits, over a total amount of data in the file F that can be reasonably assumed to be around several MBytes.

Both in the case of HMAC and CBC authentication, the received tag T, which has been computed over the entire file F, is useless in the case not all the Fragment Bundles carrying the file are received. Even if we assume that the bundle agent BA2 is in its turn able to transmit all the 8 received Fragment Bundles to a third agent BA3, a new tag T' referred to Fragment Bundles $B_1\ldots B_8$ shall be computed, and computation is performed in such a way that the received information about T cannot be exploited. As a matter of fact, the CBC-MAC tag T has been computed over all the 10 Fragment Bundles composing file F, and cannot be reused to compute T' over $B_1||B_2||\ldots||B_8$; similarly, HMAC tag computation requires the whole file F. To compute a new HMAC tag T', the concatenation $B_1||B_2||\ldots||B_8$ is needed, and the previously computed tag T cannot be reused.

If an AES GMAC authentication tag has been computed over file F, there is the possibility of exploiting this information to efficiently compute a new tag T' for the concatenation of Fragment Bundles $B_1||B_2||\ldots||B_8$, to be transferred to a third agent BA3. As shown in [11], among the properties of function GHASH, the following one is specifically tailored to the scenario herein considered:

Lemma 1: Appending and Prepending

For any $H \in \{0,1\}^w$, any bit string A with $len(A) < 2^{64}$, and any P such that $len(P) = lw$ for some value of l, the value of GHASH applied to $P||A$ can be computed as:

$$\begin{aligned} & GHASH(H, P \,||\, A, \{\}) = GHASH(H, A, \{\}) \oplus \\ & H^a \cdot GHASH(H, P, \{\}) \oplus H \cdot (len(P) \,||\, 0^{64}) \oplus \\ & H \cdot ((len(A) \oplus len(P \,||\, A)) \,||\, 0^{64}) \end{aligned} \tag{5}$$

where $a = \lceil len(A)/w \rceil$.

By exploiting the previous relation, it is possible to show how a new AES GMAC tag T' may be derived for the concatenation of Fragment Bundles $B_1||B_2||\ldots||B_8$, having received tag T computed over the whole F, i.e. over $B_1||B_2||\ldots||B_8||B_9||B_{10}$.

Let us rename $B_1||B_2||\ldots||B_8$ as F', and $B_9||B_{10}$ as S, so that $F = F'||S$. By this way, we have:

$$\begin{aligned} & T = GMAC(K, IV, F) = GMAC(K, IV, F' \,||\, S) = \\ & = GPRF(K, IV) \oplus GHASH(H, F' \,||\, S, \{\}) \end{aligned} \tag{6}$$

and

$$\begin{aligned} & T' = GMAC(K, IV', F') = \\ & = GPRF(K, IV') \oplus GHASH(H, F', \{\}) \end{aligned} \tag{7}$$

supposing that different values for the Initialization Vector are used at each AES GMAC computation, as required for a secure implementation. By developing such relations, we get:

$$\begin{aligned} & T \oplus GPRF(K, IV) = GHASH(H, F' \,||\, S, \{\}) = \\ & GHASH(H, S, \{\}) \oplus H^s \cdot GHASH(H, F', \{\}) \oplus \\ & H \cdot (len(F')) \,||\, 0^{64}) \oplus H \cdot ((len(S) \oplus len(F' \,||\, S)) \,||\, 0^{64}) \end{aligned} \tag{8}$$

where $s = \lceil len(S)/w \rceil$ (w=128, when AES is used as the basic GPRF).

Now, GHASH(H, F', $\{\}$) is what we need to compute T', which means that computation of T' is actually included into the value of T. Addition over GF(2^{128}) is identical to the bitwise exclusive-or of two terms, as in GF(2), and subtraction is identical to addition. Multiplication over GF(2^{128}) is performed according to Algorithm 1:

Algorithm 1: *Multiplication in* GF(2^{128}). *Computes the value of $Z=X \cdot Y$, where X, Y, and $Z \in$* GF(2^{128})

```
$Z \leftarrow 0, V \leftarrow X$
for $i$=0 to 127 do
   if $Y_i = 1$ then
      $Z \leftarrow Z \oplus V$
   end if
   if $V_{127} = 0$ then
      $V \leftarrow rightshift(V)$
   else
      $V \leftarrow rightshift(V) \oplus R$
   end if
end for
$Z$
```

where each element in GF(2^{128}) is seen as a vector of 128 bits (the leftmost bit is X_0 and the rightmost bit is X_{127}), $R = 11100001||0^{120}$ is a special element, whereas function *rightshift(.)* moves the bits of its argument one bit to the right.

Consequently, the value we need, i.e. GHASH(*H, F′*, {}), may be computed as:

$$\begin{aligned}&H^{s} \cdot GHASH(H, F', \{\}) = \\ &T \oplus GPRF(K, IV) \oplus GHASH(H, S, \{\}) \oplus H \cdot (len(F') \,||\, 0^{64}) \oplus \\ &H \cdot ((len(S) \oplus len(F' || S)) \,||\, 0^{64})\end{aligned} \tag{9}$$

where *T* and GPRF(*K, IV*) are already known, and GHASH(*H, S,* {}) is to be computed over the suffix *S*, i.e. a smaller amount of data than *F′*.

The same property of GHASH may be exploited to improve efficiency and performance of the so-called "toilet paper" scheme [15], proposed to include multiple authentication codes across pieces of the bundle, when bundle fragmentation occurs.

Let us assume, as an example, that a whole file *F* to be transferred between two bundle agents BA1 and BA2 in a DTN, may be fragmented into 5 Fragment Bundles, i.e. F = $B_1||B_2||\ldots||B_5$. Following the reasoning developed above, we can assume to transfer each Fragment Bundle from BA1 to BA2 together with a fixed-length GMAC tag that, unlike what is suggested by the toilet paper scheme, does not refer to the single Fragment Bundle only, but also to the previously sent ones, as described in the following lines:

- Send B_1 together with GMAC(K, IV, B_1)
- Send B_2 together with GMAC(K, IV, $B_1|| B_2$)
- Send B_3 together with GMAC(K, IV, $B_1|| B_2|| B_3$)
- Send B_4 together with GMAC(K, IV, $B_1|| B_2|| B_3||B_4$)
- Send B_5 together with GMAC(K, IV, $B_1|| B_2|| B_3||B_4||B_5$)

By this way, we create a dependence among the authentication tags of each Fragment Bundle, which, however, does not add complexity either in the transmitting, nor in the receiving node, thanks to the ``incremental'' nature of the GHASH function. As we have shown:

$$GMAC(K, IV, B_i) = GPRF(K, IV) \oplus GHASH(H, B_i, \{\}) \tag{10}$$

and, according to Eq. (2), we have that, for example:

$$\begin{aligned}&GHASH(H, B_1 \,||\, B_2 \,||\, B_3, \{\}) = GHASH(H, B_1 \,||\, B_2, \{\}) \oplus \\ &\oplus H^{2q} \cdot h(B_2)\end{aligned} \tag{11}$$

According to this processing, in the case that a Fragment Bundle is lost, the concatenation provided by the scheme among the authentication tags may help in recovering some of the missing information.

For example, let us assume that we have correctly received the Fragment Bundle B_1, whereas Fragment Bundle B_2, together with GHASH(H, $B_1||B_2$, {}) it carries, gets lost. In the case that the following Fragment Bundle B_3 with its tag GHASH(H, $B_1||B_2||B_3$, {}) is received, it allows us to recover the term $H^{2q}\cdot h(B_2)$ even if it is dependent on the missing information B_2.

The block structure of a non-protected bundle includes a primary block of fixed 121 bytes dimension, and a variable-length payload field. The primary block does not change its size and content when considering a protected bundle. In fact, the primary block contains information on source and destination addresses, that cannot be encrypted or altered: intermediate routers need such information to properly forward bundles to destination within the DTN. If the protected bundle is obtained by application of HMAC with SHA-1, the global dimension of the bundle increases by 20 bytes, and only by 8 bytes if CBC-MAC with AES or AES GMAC are applied. Fig. 1 shows the percent incidence of the security overhead on protected bundles, with respect to non-protected ones, for different types of security solutions applied.

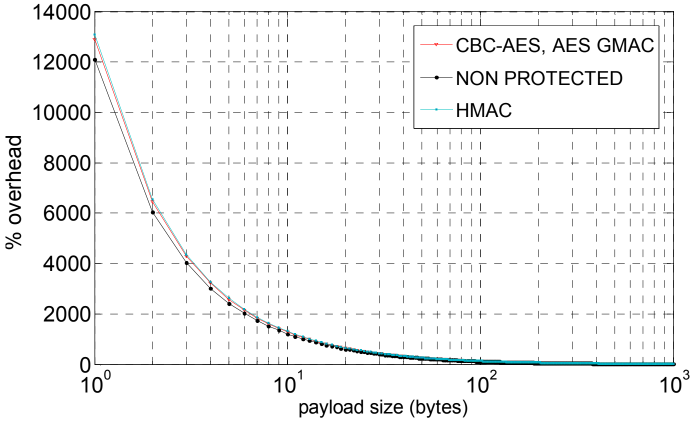

Fig. 1. Percent incidence of the overhead due to different security algorithms applied on bundles, for a payload dimension varying from 1 to 10^3 bytes

As expected, if no security algorithms are applied, i.e. the bundles are left non-protected, the percent overhead incidence on the payload is limited. It increases at maximum values when using HMAC with SHA-1, with lower impact due to CBC-MAC or GMAC with AES. By increasing the dimension of the bundle payload it is possible to reduce the impact of overhead, thanks to the fixed number of bytes used for security purposes.

GMAC does not directly support incremental tag verification. The verification of a single data block out of a large set of blocks may be performed through a *memory checker*; even if GMAC cannot act as a memory checker by itself, it would be possible to define such a kind of function on the basis of GMAC.

V. Conclusion

This paper examined the security issues related to authentication in Disruption Tolerant Networks, with specific reference to Space Networks, where the peculiar features of the communication links make a number of classical solutions inefficient. The Bundle Protocol security options have been examined in details, in order to identify the open issues needing further discussion; among them, the problem of Fragment Bundle authentication, for which the adoption of the Galois Counter Mode scheme has been suggested through a number of positive features that make it suitable to the scenario of interest. Other issues still remain to be addressed, such as key management, *IV*s generation, and the possibility of performing incremental MAC verification.

References

[1] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary Internet: state-of-the-art and research challenges," International Journal of Computer and Telecommunications Networking, Vol. 43, Issue 2, October 2003, pp. 75 - 112.

[2] K. Fall, S. Farrell, "DTN: an Architectural Retrospective," IEEE Journal on Selected Areas in Communications, vol. 26, no. 5, pp. 828 - 836, June 2008.

[3] N. Asokan, K. Kostianinen, P. Ginzboorg, "Towards Securing Disruption-Tolerant Networking," Nokia Research Center Report NRC-TR-2007-007,March 2007.

[4] K. Scott, S. Burleigh, "Bundle Protocol Specification," Internet RFC 5050, November 2007.

[5] S. Symington, S. Farrell, H. Weiss, P. Lovell, "Bundle Security Protocol Specification," draft-irtf-dtnrg-bundle-security-07

[6] http://www.scps.org/index.html, retrieved on February 2009.

[7] L. Zhang, S. Spinsante, C. Tang, E. Gambi, "Application and performance analysis of various AEAD techniques for space telecommand authentication," IEEE Trans. Wireless Communications, Vol. 8, pp. 308 - 319, January 2009.

[8] D. Fischer, M. Merri, T. Engel, "Security Extensions for Space-Link Communication," Proc. of 17th International Conference on Computer Communications and Networks, 3-7 Aug. 2008, Page(s):1 - 6.

[9] P. Rogaway, "Authenticated encryption with associated data," Proc. ACM Conference on Computer and Communications Security (CCS-9), ACM Press, pp. 196 - 205, November 17-21, Washington DC, USA, 2002.

[10] D. A. McGrew, J. Viega, "The Galois/Counter Mode of Operation," Submission to NIST Modes of Operation Process, January, 2004.

[11] D. McGrew, "Efficient authentication of large, dynamic data sets using Galois/Counter Mode (GCM)," Proc. 3rd IEEE International Security in Storage Workshop, pp. 89 - 94, 2005.

[12] E. H. McKinney, "Generalized Birthday Problem," American Mathematical Monthly, Vol. 73, pp. 385 - 387, 1966.

[13] National Institute of Standards and Technology, Federal Information Processing Standards Publication FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code (HMAC)," July 2008.

[14] National Institute of Standards and Technology, Federal Information Processing Standards Publication FIPS PUB 180-1, "Secure Hash Standard," April 1995.

[15] C. Partridge, "Authentication for fragments," ACM SIGCOMM Fourth Workshop on Hot Topics in Networks, November 2005, USA.

S. Spinsante (M'01–SM'11) received her PhD in Electronics and Telecommunications in 2005, from Università Politecnica delle Marche, Ancona (Italy). Since then she has been a research fellow at the Department of Information Engineering of the same University, where she works on security for TM and TC transmission in space applications, and spread spectrum systems for communications and radar applications.