2015

Loss-tolerant bundle fragment authentication for space-based DTNs

Xixiang Lv
Xidian University, xlv@uow.edu.au

Yi Mu
University of Wollongong, ymu@uow.edu.au

Hui Li
Xidian University, lihui@mail.xidian.edu.cn

Publication Details

Lv, X., Mu, Y. & Li, H. (2015). Loss-tolerant bundle fragment authentication for space-based DTNs. IEEE Transactions on Dependable and Secure Computing, 12 (6), 615-625.

Loss-tolerant bundle fragment authentication for space-based DTNs

Abstract
Bundle authentication, which ensures the authenticity and integrity of bundles, is critical in space Delay/disruption-Tolerant Networks (DTNs). When bundle fragment services are needed, the previous solutions directly using digital signatures suffer from heavy computational costs and bandwidth overheads. This paper addresses the issue of fragment authentication for Bundle Protocol by exploiting erasure codes and the batch transmission characteristic of DTNs. Erasure codes are adopted to allow all the fragments of a bundle to equally share only one signature, to tolerate high delays as well as unexpected loss of connectivity. Following this generic idea, we present two approaches, both of which are effective in filtering inauthentic fragments as early as possible. The first one takes a surprisingly low bandwidth overhead, while it makes all received fragments of a bundle to be removed when there is an inauthentic one, because of its failure in locating the inauthentic fragments. Considering this defect, we present an improved scheme which is able to detect inauthentic fragments thanks to a special hash chain and then only remove these inauthentic ones. The performance simulation demonstrates that both our schemes significantly reduce bandwidth overheads and computational costs as compared to the prior works.

Disciplines
Engineering | Science and Technology Studies

Publication Details
Lv, X., Mu, Y. & Li, H. (2015). Loss-tolerant bundle fragment authentication for space-based DTNs. IEEE Transactions on Dependable and Secure Computing, 12 (6), 615-625.

Loss-tolerant Bundle Fragment Authentication for Space-based Networks

Xixiang Lv, Yi Mu, *Senior Member, IEEE,* and Hui Li

Abstract—Delay Tolerant Networks (DTNs), as the principal technology for space internet, implement and support internetworking via the Bundle Protocol (BP). Bundle authentication is a critical security service in DTNs which ensures the authenticity and integrity of bundles during parallel multi-hop transmissions, especially when the bundle fragment service is needed. The authentication based on digital signatures, suggested in the CCSDS bundle security protocol specification, inevitably adds computational cost, transmission overhead and energy consumption. In this paper, we address the issue of fragment authentication for BP by exploiting the well known erasure codes and the batch transmission characteristic of DTNs. In our schemes, erasure codes are adopted to allow the receipts to recover the signature and the hashes of all the fragments of a bundle no matter how many fragments they receive. In other words, only a part of authentication information are sufficient to authenticate the received fragments from a bundle. Such an approach tolerates the high delay and unexpected loss of connectivity of DTNs, and also the BP nature of routing fragments of the same bundle possibly via different paths. With this main idea based on erasure codes, we present three approaches, all of which work well in resisting injection attacks and eradicating traffic storms. The first one takes a surprisingly low communication cost due to fragment authentication, while it makes all received fragments of a bundle to be removed when there is an injected one, because of its failure in locating the false fragments. Considering this defect, we present an improved scheme which is able to detect injected fragments thanks to the special geometrical structure of vector space and then only remove these injected ones. Aiming to further reduce communication overheads, we constructed a specially linked hash chain based on which an injected fragment is quite easy to be filtered with smaller authentication tags appended to fragments. The performance analysis verifies that our schemes, especially the third one, significantly reduce bandwidth overheads and computational costs as compared to the prior works, and thus minimize the energy consumption due to fragment authentication.

Index Terms—Delay Tolerant Networks, Authentication, Security

1 Introduction

Many of the popular applications on Internet today are built on the assumption of immediate end-to-end reachability. However, this assumption never holds true for space internet, which are recognized as a type of disruption/disconnect-tolerant networks (DTNs) [1]. Due to this special nature of space internet, the traditional security mechanisms are not always applicable, such as authentication for bundles. Traditional authentication mechanisms, i.e., calculating a hash, signing the hash and appending the signature to the message, are not optimal when fragmentation of bundles is needed. In this paper, we discuss the efficient authentication mechanisms for fragments of bundles in DTNs. In order to resistant injection attacks and eradicate traffic storms, we consider using the well known erasure codes to construct efficient authentication mechanisms for BP (Bundle Protocol) fragments.

The DTN architecture which is described in RFC-4838 [2] is a generalized store-and-forward network overlay. It originates from NASA JPL's experiences in developing store-and-forward communication networks for deep space with high delay and frequent disconnectivity. In DTNs, the in-transit PDUs (protocol data unit), also named bundles can be sent over an existing link and buffered at the next hop until the next link in the path appears. Networking using this bundle-based store-and-forward techniques allows communication over multiple hops in a parallel manner. Then, malicious routers can arbitrarily inject false information into the bundles. If innocent routers further propagate these injected messages, attackers then are able to generate large amount of malicious traffic in the network. We call this type of attacks as injection attacks, which will lead to traffic storms [3]. Since space DTNs depend on resource-constrained mobile devices, traffic storms may result in a serious threat on the network operation [4]. Therefore, to filter injected messages as early as possible, a secure and efficient bundle authentication mechanism should be in place. Actually, bundles in space DTNs need to be authenticated for several other reasons, such as routing on intermediate nodes. Due to the limited

- *X. Lv is with the National key lab of ISN, Xidian University, Xi'an 710071, China*
  Email: xxlv@mail.xidian.edu.cn
- *Y. Mu is with Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, NSW2522, Australia.*
  E-mail: ymu@uow.edu.au
- *Hui Li is with National key lab of ISN, Xidian University, Xi'an 710071, China*
  Email: lihui@mail.xidian.edu.cn

resources, intermediate nodes may want to use authentication as the basis for policy-based routing and forwarding. A typical scenario is that an intermediate node only wants to store and forward messages from a pre-defined set of known senders. In addition, the recipient might also need to authenticate the originator for correctly interpreting the contents.

With respect to bundle authentication, the DTN Research Group (DTNRG) has produced an Internet draft describing a bundle security protocol specification [5]. The specification describes three IPsec [6] style security headers that can be appended to bundles to provide three aspects of security services. The Payload Integrity Header (PIH) is used to provide end-to-end authentication in a similar fashion. The Payload Confidentiality Header (PCH) is used to encapsulate encrypted payload. The Bundle Authentication Header (BAH) provides authentication for single hop by adding a message authentication code (MAC) or a signature to each bundle.

However, in space DTNs, fragmentation of bundles is often needed. What this means is that an intermediate node can split a large bundle into smaller fragments and route different fragments through different forwarding paths to make the best use of limited resources. With the CFDP/BP/LTP stack, there are several of places where fragmentation and/or aggregation can take place [1]. First, CFDP (CCSDS File Delivery Protocol) and its UT-layer (Unitdata Transfer layer) interface to BP dictates the sizes of the bundles exchanged between CFDP entities. While BP may fragment bundles for transmission through the network, it must reassemble them completely at the destination before delivering them to the application. In addition, in order to recover from unexpected loss of link connectivity, some BP stack implementations may consider reactive fragmentation; that is, if the transmitting BP router knows how much data from a particular bundle transmission has been successfully received when the connectivity is broken off, it could choose to reactively fragment the bundle. This assumes that the receiver forms a bundle fragment from the data already received, and the sender forms another fragment from the last known byte received. The fragment at the source is then re-routed, possibly via another path if one is available.

Thus, when bundle fragmentation is supported, CCSDS Bundle Security Protocol Specification [5] does not work well, in that the additional signature to each fragment of a bundle will involve a series of performance obstacles: high transmission overheads, computational costs and energy consumption. Obviously, the large size of digital signature will introduce increasing extra transmission overhead, and verifying those individual signatures one by one at each intermediate DTN router and the receiver will significantly increase the computational costs of bundle authentication. Another unapparent performance obstacle is energy consumption. In fact, the high transmission and computational overhead also translates to high energy consumption. This energy consumption issue becomes more challenging when multi-copy or even flooding based propagation method is employed in order to enhance the reliability of space DTN transmission [7], [8], since the signature transmission and verification are performed along each fragment delivery path.

This paper discusses methods to authenticate fragments of bundles of space DTNs efficiently in computation and inexpensively in communication. Specifically, our approach tolerates loss of fragments, i.e., loss of some fragments does not affect the authenticity and integrity verification for the remainder fragments; and intermediate routers are able to verify the received fragments even if other fragments of the same bundle are routed to other paths.

Our contributions. This paper investigates a close variation of the above mentioned problem of BP fragment authentication and focuses on "how to authenticate fragments for BP of space internet efficiently in computation and communication". We provide a brief overview of some new ideas to solve this problem. Then, we aim to minimize the costs of computation/communication and to tolerate the nature of multi-hop parallel forwarding of BP as well. We use a combination of hash and signature techniques with FEC, or more precisely, erasure codes. Erasure Codes are utilized to adapt to high delay, unexpected loss of connectivity, and to allow the transmission overhead from the only one signature to be equally shared by all fragments, each of which carries a lower overhead. In this way, the signature and the hashes of all the fragments of a bundle are encoded to generate n codewords appended to n fragments, where only np of them are sufficient to authenticate all the fragments in the bundle ($p < 1$). Since the intermediate router or the receiver can recover the authentication information from at least np codewords carried by the received fragments, such an approach allows up to $(1-p)n$ not received fragments due to high delay and unexpected loss of connectivity, and also tolerates the BP nature of routing fragments of the same bundle possibly via different paths. Actually, here erasure codes are used to operate on packet-sized data objects.

On the above-mentioned basis, we propose three candidate approaches, all of which are effective in resisting injection attacks and eradicating traffic storms. The first one adds a surprisingly low extra overhead to BP, while it makes all received fragments of a bundle to be removed when there is an injected fragment, because of its failure in locating the false fragments. In the second approach, considering this defect, we suggest utilizing the special geometrical construction of vector space to detect maliciously injected frag-

ments and then only remove these injected ones. Aiming to further reduce communication overheads and computation costs, we constructed a specially linked hash chain based on which injected fragments are quite easy to be filtered with smaller additional authentication tags to fragments. Our performance evaluations justified the merits of our schemes.

Paper organization. The remaining sections are arranged as follows. In Section 2, we revisit DTNs security issues and fragment authentication for BP. In Section 3, we describe the models and our design goals. In Section 4, we will present our first erasure codes based scheme, which is improved in Section 5 by using geometrical construction of vector space to locate injected fragments. In Section 6, we further reduce the communication overhead by utilizing a novel hash chain to filter injected fragments. Section 7 concludes the paper.

2 Related Works

DTN Security and Bundle Authentication. A DTN, accepted as the space internetworking approach, can bridge between dissimilar lower-layer networks and can operate simultaneously over heterogeneous networks (i.e., CCSDS, IPv4, IPv6 and Bluetooth) or even directly over a datalink protocol. It operates over convergence layers which are underlying protocols that accomplish communication between DTN entities using the Bundle Protocol. Security issue is one of the major challenges for DTN deployment. Farrell and Cahill review the current state of DTN security works in [4]. They point out that the main threats for DTNs are modification of messages (or "bundles"), unauthorized use of scarce DTN resources and denial of service. Due to resource-scarcity characteristic of DTNs, a general motivation for DTN security is to prevent the attackers from unauthorized accessing and utilizing of DTN resources. To this end, the DTN Bundle Security Specification [5] defines BAH, PIH, and PCH to provide security services on the network layer. An additional draft [9] explaining the rationale for the design choices made in this specification. Even though public key signature based bundle authentication solutions adopted in current Bundle Security Protocol Specification has provided a general framework to secure space DTNs and it also holds the great advantage of providing interoperability, there are still two open issues: fragment authentication issue and performance issue.

Due to fragmentation, traditional authentication scheme [5], e.g., the sender generates the signature over an entire message, may not work well since the intermediate receiver cannot authenticate any of the received fragments if it has not yet received the entire message. To address this problem, a trivial approach was proposed in [10] and has come up at the DTNRG mailing list discussions. The main idea is to make each fragment self-authenticating by attaching a signature to each fragment separately, i.e., each fragment carries its own signature to make it self-authenticating. However, this approach may lead to a more serious performance issue since the intermediate nodes have to take more computational and transmission efforts for transmitting and verifying a growing number of signatures. Because of the resource-scarcity characteristic of space networks, how to minimize the security cost and improve the bundle authentication efficiency becomes a critical problem. Efficiency issue is extremely important in space DTNs because the multi-copy routing/forwarding is very common and the fragmentation issue also makes this problem more challenging.

Aiming to achieve advantages in efficiency over the trivial scheme in [10], Partridge presents a few solution proposals for this fragment authentication problem in [11]. The first proposal is to use cumulative authentication in which each fragment is authenticated by calculating a hash over all the previous fragments including the current fragment. The computation costs for the receiver is less than the trivial approach in [10], since only one signature per a set of received fragments has to be verified. However, this approach does not reduce computational costs of sender or the amount of traffic, since a signature has to be generated and added to each fragment. In addition, this approach assumes that fragments are received in order which might not be the case always. The second proposal in [11] is to authenticate fragments using function definitions. Rather than protecting the data with a hash value from an authentication function, this proposal suggests to protect the data with a dynamically generated function that gives a known result. N. Asokan [3] presents a binary hash tree (or Merkle Tree) based fragment authentication approach in which each fragment has to carry $\log_2 n$ hash values for verifying itself. Accordingly, it is very expensive when n is large. The latest contribution by Zhu et al. [12] presents an opportunistic batch authentication scheme to achieve efficient bundle authentication, in which the well known Merkle Tree is used to minimize communication overhead and a batch verifiable signature is used to minimize the computation costs of the receiver.

Since the above described two issues are closely related, we aim to address the two issues together. The objective of this paper is to minimize the computational and transmission overhead due to fragment authentication by exploiting erasure coding and the bundle buffering characteristics.

Erasure Coding. Now we give a brief introduction to the principle of erasure codes, especially clarifying their special operation manner here. For a more in-depth discussion on erasure codes the interested reader is referred to the literatures [13], [14], [15], [16], [17].

An erasure encoding algorithm $C_{k,r}$ takes a set $X = x_1, x_2, \cdots, x_k$ of k source packets in a block as input and produces $n = (k + r)$ encoded packets: $\{y_1, \cdots, y_{k+r}\} \leftarrow C_{k,r}(X)$. The main property of the set $Y = \{y_1, \cdots, y_{k+r}\}$ is that any subset of k elements of Y is sufficient to recover the source data X via the corresponding decoding algorithm D_k.

Rizzo [17] presents an erasure code which is flexible and efficient to implement, and discusses various issues related to its performance. Reed-Solomon erasure codes given in [17] operate in $GF(2^n)$ and may be inefficient for large data packets with several hundred kilobytes for each. However, they are appropriate for our scenario because we operate on hash values (typically 16 or 20 bytes) that are much smaller than packets. Rizzo also suggested to use erasure codes in multicast and broadcast applications, which is instantiated in [18]. For faster codes, the interested readers are suggested to refer to [19], [20] describing Tornado Codes with near linear coding and decoding times. In this paper, we suggest to consider the so-called linear block codes since they are simple and appropriate for the application scenario of our interest.

It is important to note that erasure codes are used here in a completely new manner. On one hand, here they are not used in the traditional context as telephony, in that an individual tag can intuitively be viewed more like a single code symbol rather than a set of symbols. On the other hand, the coding operation is also different from that of reliable transmission mechanisms based on erasure codes in deep space networks [21], [22], [23], [24], [25], [26], since we use coding on small authentication tags rather than packets themselves. We use erasure codes not to recover damaged packets but rather the lost authentication tags of fragments of a bundle.

3 Models and Design Goals

In this section, we describe the system and adversary models, followed by our design goals.

System Model. We consider space DTNs which can bridge between dissimilar lower-layer subnetworks and operate simultaneously over heterogeneous subnetworks using the Bundle Protocol suits. In such a space internet, a source node can deliver bundles to a destination node via one or multiple paths depending on any particular forwarding algorithm [8]. Specifically, for an intermediate node, it may contemporarily receive bundles from multiple senders via one or multiple hops. These received bundles will be manipulated based on two special technologies: store-and-forward and custody transfer [1]. Store-and-forward means storing data before it is forwarded. Unlike the Internet model of store-and-forward where data is stored long enough to look up a next hop and queue the data on an outbound link, the consideration here is to store PDUs for arbitrary lengths of time, if no outbound path is currently available. Custody transfer is a network service that provides reliability on a hop-by-hop basis rather than an end-to-end basis. With custody transfer, the DTN node is responsible for retransmitting data if that data is lost in progress towards the data destination.

Further, a bundle may be split inside the network and reassembled later before being delivered to its destination [1]. What this means is that the source or an intermediate node can split a large bundle into smaller fragments and route different fragments possibly through different forwarding paths, to adapt to high delay/disruption space links and make the best use of limited resources. We think that BP cannot aggregate multiple bundles together into larger data units for transmission. This paper follows a general assumption such as [27] that there exits an Offline Security Manager (OSM) to take charge of issuing certificate for traditional public key cryptosystem or generating private key for identity based cryptosystem.

Adversary Model. Consider that the adversary's goal is to inject false messages into the network, attempting to deceive other DTN nodes, gain unauthorized access to DTN resources, or exhaust constrained network resources, especially energy resources. Here, we call this type of attacks as injection attack which is a variant of bogus message flooding attack in which a malicious DTN node might arbitrarily inject forged bundles or fragments. However, we consider neither adversaries being able to compromise DTN nodes, nor adversaries aiming at signature algorithm itself. This adversary model is defined in [12].

Design Goals. Our security design goal is straightforward, i.e., resisting injection attacks and eradicating traffic storms with low costs in communication and computation. To be specific, when fragmentation service is needed and only one single signature for a bundle is shared by all of its fragments, all received or relayed fragments should be authenticated even if some other fragments of the same bundle are lost or routed via other paths, so that the false ones inserted by any illegitimate DTN users or external attackers can be efficiently rejected or filtered as early as possible. Meanwhile we concentrate on minimizing the overheads due to authentication for bundle fragments. Especially, computational cost, communication overhead as well as energy consumption are given priority to cope with the resource constrained nature of space DTNs. In addition, we want to detect the false fragments (bing modified or injected) by utilizing the geometrical construction of vector space or a specially linked hash chain, and therefore only remove these false fragments.

4 The Basic Scheme

In this section, we propose a bundle fragment authentication scheme which aims to minimize the computa-

tional overhead by exploiting only one signature and one verification for all fragments of the same bundle. Then, we take advantage of erasure coding technique to tolerate unexpected loss of fragments and the DTN nature of routing fragments of the same bundle via different paths.

In this work, we will consider a bundle to be split into n consecutive fragments. For simplicity, we assume that a bundle exactly contains a number of fragments which is an exact multiple of n. Our fragment authentication scheme is parameterized by n the bundle size in fragments and p the average probability that all fragments of the same bundle are routed to the same intermediate router (or node). For simplicity, we further assume that the average probability p covers fragment loss due to unexpected loss of connectivity or others.

We denote H as a cryptographic hash function such as SHA or MD5. The couple (S, V) denotes the digital signature and verification algorithms with the public key K_p and private key K_s respectively. The size of the signatures will be expressed as s bytes. For RSA, a typical value for s is 128 bytes (or 1024 bits).

4.1 Algorithms

The objective of the following Algorithm 1 is to generate a set of security tags from the signature of a bundle and respectively append each of them to a fragment of this bundle, so that the receiver and intermediate routers can verify and distinguish the authentic fragments conditioned on correctly receiving at least np authentic fragments. In principle, the security tag is extra information that represent communication overhead being used to resist injection attacks.

Generation of Authentication Tags. We denote the fragments of a bundle as $f_1, f_2, \cdots, f_n$, each of which will piggyback an authentication tag before transmission. The authentication tags are extracted as follows:

Algorithm 1.

1) Let $X = \{h_1, h_2, \cdots, h_n | h_i \leftarrow H(f_i), i = 1, 2, \cdots, n\}$;
2) $\{X; X'\} \leftarrow C_{n,r}(X)$;
3) $\sigma \leftarrow S(K_s, H(h_1 \| h_2 \| \cdots \| h_n))$;
4) Split $(\sigma \| X')$ into $\lceil pn \rceil$ blocks $B = \{B_1, B_2, \cdots, B_{\lceil pn \rceil}\}$ with the same size;
5) $\{B; B'\} \leftarrow C_{\lceil pn \rceil, \lceil (1-p)n \rceil}(B)$;
6) Append each of the codewords of $\{B; B'\}$, represented by $T = \{\tau_1, \tau_2, \cdots, \tau_n\}$, to one fragment respectively.

Here, we denote by S a digital signature algorithm and by σ the resulting signature signed with private key K_s. In this algorithm, we let $C_{n,r}$ be an erasure encoding algorithm with r extra redundancy codewords and total $(n + r)$ codewords. The number of extra redundancy codewords r satisfies two requirements: (1) r is larger than or equal to $\lceil (1-p) \rceil n$ for successfully decoding in Step 3 of the following Algorithm 2; and (2) enough redundancy codewords make space networks be able to withstand the average probability of p that all fragments of the same bundle are routed to the same intermediate router. Then, r is determined by this average probability p and a concrete coding algorithm. For example, if we use the erasure code based on Vandermonde matrices [17], the minimum value of r is $\lceil \frac{(1-p)n}{p} \rceil$. We also use the erasure code based on Vandermonde matrices for $C_{\lceil pn \rceil, \lceil (1-p)n \rceil}(B)$, aiming to get exactly n codewords of $\{B; B'\}$. Thus we must split $(\sigma \| X')$ into $\lceil pn \rceil$ blocks with same sizes and each block B_i is an original word to be coded.

Decoding and Verification. Consider the set $R = \{R_i\}$ of the received fragments, where $R_i = \{f_i' \| \tau_i\}$ and $|R| \geq (pn)$. Here, i is f_i''s position, or index in the original encoding operation. we assume that $T = \{\tau_i\}$ includes $|R|$ tags from R. The verifier will use these authentication tags to recover the signature for all the fragments and other authentication information to verify the signature. The decoding and verification is as follows:

Algorithm 2.

1) $\{\sigma \| X'\} \leftarrow D(T)$;
2) For each fragment in R, compute $h_i' = H(f_i')$ and let $E = \{h_i'\}$;
3) $\{h_1, h_2, \cdots, h_n\} \leftarrow D(E \| X')$;
4) $V(\sigma, \{h_1, h_2, \cdots, h_n\})$ to authenticate the received fragments.

Here, $D(\cdot)$ denotes the decoding algorithm. This decoding and verification process tells us that any subset of $\{f_1, f_2, \cdots, f_n\}$ can be authenticated if its size is larger than or equal to $\lceil pn \rceil$. What this means is that if the average probability p is accurately estimated, the only one signature equally shared by all fragments can authenticate all the fragments of a bundle. Thus, we can effectively reduce the transmission cost as well as computational cost and thus minimize the energy consumption due to bundle fragment authentication. It is important to note that the decoding algorithm needs to know the position, or index, of the received fragments and this information can be derived from the sequence numbers of the fragments in the PDUs.

4.2 Parameters and Performance Evaluation

Using Learning to Approximate p. In order to utilize erasure cods, we need the global information such as the probability p. This can be achieved by a history learning process [28]. For example, each node records the number of total received fragments of the same bundle during a specific past time duration. It also periodically updates and broadcasts its probability information. The node computes the overall approximation of p based on its local record and the received

neighboring information. Thus, all nodes will have the global and accurate view about the network history. This history can be limited to some time duration if the network size is large.

The objective of the above scheme is to minimize the cost of bundle authentication when fragmentation service is needed in DTNs. Thus, the greater n, the less often we will need to generate a signature and to transmit the authentication tag. Meanwhile, the lower probability p will result in more redundancy codewords. Theoretically, the erasure coding can withstand arbitrarily small p. In other words, enough extra redundancy will always make all fragments of a bundle to be authenticated. However we need to relate the parameters n and p to concrete average network patterns in space. That is, the sender needs to adjust the parameter n such that most bundles would be verifiable.

Size of an Authentication Tag. Let h define the length of the cryptographic hashes and s the size of the signatures. The size of an individual authentication tag is expressed as $\lambda(n,p)$ of both the number of fragments of a bundle and the average probability that all fragments of the same bundle are routed to the same intermediate router (or node). From Algorithm 1, we have

$$\lambda(n,p) = \frac{rh+s}{\lceil pn \rceil}, \text{ where } r = \lceil \frac{(1-p)n}{p} \rceil.$$

This $\lambda(n,p)$ presents the communication overhead of our bundle fragment authentication scheme, and remains surprisingly small if either n is large or p is reasonably high.

The simulations presented in [12] show that, in a high traffic load case, there exist up to 98.25% DTN contacts during which DTN transmission is performed in a batch (two or more bundles are transferred simultaneously). This result also applies to the fragments of the same bundle, i.e., most fragments of the same bundle are transferred simultaneously in DTN. What this means is that the average probability p with which all fragments of the same bundle are routed to the same intermediate router (or node) is sufficiently high. Thus, our scheme will place a very low communication overhead to BP for fragment authentication. With $s = 128$ bytes and $h = 16$ bytes, we present a sampling of $\lambda(n,p)$ for different p and n in Table 1. Note that we let the value of λ to be the lowest multiple of $\lceil pn \rceil$ greater than or equal to $(rh+s)$.

Computation Costs. The main computation cost for a sender comes from cryptographic hash computations, a digital signature and two coding operations. For each bundle, the source needs to generate n hash values, a digital signature, and encode twice. Here, note that the erasure codes may be computationally expensive for large data packets but this operation on our small authentication tags will be relatively efficient.

TABLE 1: Overhead bytes per fragment.

p \ n	16	32	64	128	256	512	1024
0.9825	9	5	3	2	1	1	1
0.75	19	13	10	9	8	8	8
0.6	31	24	21	20	19	19	18

The computation cost for the recipient depends on the average probability p of the space DTNs. It only needs to compute n hashes and verifies a signature in an ideal situation, i.e., it does not need decoding since it gets all fragments. If the recipient does not receive all the fragments of a bundle, it will need to calculate some additional decoding. The decoding is utilized to recover the hashes of the missed fragments, rather then the fragments themselves. Thus, the decoding is not expensive since the tags are much smaller than general packets.

Discussion about Pollution Attacks. The above scheme is able to eradicate traffic storms in that any injected false information into bundles will be filtered as early as possible by signature verification at the intermediate routers. Actually, it makes all received fragments of a bundle to be removed even if there is a single modified fragment or a malicious fragment injected by an adversary, because it can not locate the false fragments. This may be an issue in a sense and can be addressed by each fragment carrying more extra authentication information. However, it shows a trade-off between the performance and communication/computation costs, especially for resource-scarcity space networks. To locate the modified or injected fragments and to exactly recover the authentication information, a sender must augment each fragment with enough redundancy information. What follows is an improved scheme with this idea.

5 Scheme with Detecting False Fragments

In order to detect false fragments, the source generates another tag and appends it to the corresponding fragment. This tag enables the verifier to identify the authentic fragments and just remove the injected or modified fragments. Here, $f_i, i = 1,2,\cdots,n$ still represent n fragments of a bundle to be sent. This type of authentication tags are extracted as follows:

Algorithm 3.

1) Let $X = \{h_1, h_2, \cdots, h_n | h_i \leftarrow H(f_i), i = 1,2, \cdots, n\}$;
2) $\{X; X'\} \leftarrow C_{n,r}(X)$;
3) $\sigma \leftarrow S(K_s, H(h_1 \| h_2 \|, \cdots, \| h_n))$;
4) Split $(\sigma \| X')$ into $\lceil pn \rceil$ blocks $B = \{B_1, B_2, \cdots, B_{\lceil pn \rceil}\}$ with the same size;
5) $\{B; B'\} \leftarrow C_{\lceil pn \rceil, \lceil (1-p)n \rceil}(B)$;
6) Each of the codewords of $\{B; B'\}$ is used as an tag and let $T = \{\tau_1, \tau_2, \cdots, \tau_n\}$;

7) Map the hashes $h_1, h_2, \cdots, h_n$ of the fragments into vectors $\overrightarrow{h_1}, \overrightarrow{h_2}, \cdots, \overrightarrow{h_n}$ in an n dimensional vector space $\mathbf{V} = F_q^n$;
8) Derive $\overrightarrow{u_i}$, $i = 1, 2, \cdots, n$ from $\overrightarrow{h_1}, \overrightarrow{h_2}, \cdots, \overrightarrow{h_n}$;
9) Append the authentication tag $(\tau_i \| \overrightarrow{u_i})$ to the fragment f_i, $i = 1, 2, \cdots, n$.

Here, note that every set of $(n-1)$ linearly independent vectors in vector space $\mathbf{V}$ spans a unique hyperplane (or an $(n-1)$ dimensional sub-space) $P \subset \mathbf{V}$. Then, for each $\overrightarrow{h_i}$ we can derive a vector $\overrightarrow{u_i}$ which is orthogonal with the hyperplane $P_i \subset \mathbf{V}$ spanned by the other $(n-1)$ linearly independent vectors $\overrightarrow{h_j}$, $j = 1, 2, \cdots, n$ and $j \neq i$. Obviously, $\overrightarrow{u_i}$ uniquely represents the hyperplane P_i.

The coding algorithms and parameter r are similar to that of Algorithm 1.

We assume that the set $R = \{R_i\}$ includes the received fragments, in which $R_i = \{f_i' \| \tau_i \| \overrightarrow{u_i}\}$, $|R| \geq (pn)$, and i is f_i''s position in the original encoding operation. Let the set $T' = \{\tau_i\}$ include $|R|$ tags from R. The verifier will use T' to recover the signature for all the fragments. The vector $\overrightarrow{u_i}$ is used to find and remove a modified fragment or a maliciously injected fragment. The decoding and verification is as follows:

Algorithm 4.

1) For each fragment in R, compute $h_i' = H(f_i')$ and $E = \{h_i'\}$;
2) Map each hash value in E into the n dimensional vector space $\mathbf{V} = F_q^n$ and get another set $\{\overrightarrow{h_i'}\}$;
3) For each vector in $\{\overrightarrow{h_i'}\}$, if $\overrightarrow{h_i'}$ belongs to at least $(\lceil pn \rceil - 1)$ hyperplanes represented by the tags among $\{\overrightarrow{u_j}, j \neq i\}$, f_i' is authentic; otherwise, remove h_i' from E and f_i' from the received fragment set R;
4) If the size of E is not smaller than $\lceil pn \rceil$, decode as $\{\sigma \| X'\} \leftarrow D(T')$ and $\{h_1, h_2, \cdots, h_n\} \leftarrow D(E \| X')$;
5) $V(\sigma, \{h_1, h_2, \cdots, h_n\})$ to authenticate the received fragments.

Making the decision on whether f_i' is an authentic fragment or not is based on the fact that if $\overrightarrow{h_i'}$ is an authentic tag, then it belongs to at least $(\lceil pn \rceil - 1)$ other authentic hyperplanes P_j represented by $(\lceil pn \rceil - 1)$ authentic $\{\overrightarrow{u_j}\}$. This filter is also based on the assumption that there are at least $\lceil pn \rceil$ authentic fragments and $\lceil pn \rceil$ authentic vectors $\overrightarrow{u}$ in the received fragments. If not so, it is impossible to recover the authentication tags and thus to verify the signature, even though all the fragments and tags are authentic. What this means is that $\lceil pn \rceil$ is the threshold making our scheme work well. Note that the screening condition of Step 3 in Algorithm 4 is strict enough in that if f_i is authentic there is at least one hyperplanes P_j represented by the tag $u_j, j \neq i$ satisfying the requirement of $\overrightarrow{h_i'} \subset P_j$, unless all the tags $\overrightarrow{u_i}$ are modified.

Obviously, the probability of the hash vector h_i' of a modified or random injected fragment belongs to the hyperplane represented by the tag $u_j, j \neq i$ of an authentic fragment is $1/q$. Thus, it can be observed that an injected or modified fragment passes the authenticity filtering with the probability of at most $(\frac{1}{q})^{\lceil pn \rceil}$, where q is the field size.

The communication overhead of this improved scheme is

$$\lambda'(n, p) = \lambda(n, p) + |\overrightarrow{u}| = \frac{rh + s}{\lceil pn \rceil} + n \log_2(q),$$

where

$$r = \lceil \frac{(1-p)n}{p} \rceil.$$

In Table 2, we present a sampling of the communication overhead with $q = 4$. From Table 2, we get

TABLE 2: Overhead bytes per fragment.

p \ n	16	32	64	128	256	512	1024
0.9825	13	13	19	34	65	129	257
0.75	23	21	26	41	72	136	264
0.6	35	32	37	52	83	147	274

a conclusion that the extra tag $\overrightarrow{u}$ greatly increases the communication overhead of authentication for bundles, especially at a larger number of fragments. Thus, this scheme is desirable when the number of fragments is relatively small. With $q = 4$, this scheme is not applicable for space DTN when the number of fragments is larger than 256. In this case, it makes no sense and is even worse than the scheme proposed in [10]. Considering this efficiency issue, we give another improved scheme in the following section, in which we utilize a special hash chain to filter injected fragments and adopt erasure coding to tolerate fragment loss and the nature of multi-hop parallel forwarding of BP as well. With this scheme, the false fragments inserted by any illegitimate DTN users or external attackers can be efficiently rejected in an early stage, thus injection attacks can be eradicated effectively.

6 Scheme Based on Special Hash Chain

In this section, we propose a bundle fragment authentication scheme which aims to effectively filter injected or false fragments and minimize the communication overhead. We first present a special hash chain which is used to verify fragments originated from the same bundle and resist injection attacks.

6.1 Construction of Special Hash Chain

The objective is to generate a set of verification tags and append each of them to a fragment such that the recipient can verify and distinguish the fragments from the same bundle. We denote the payload of a bundle as M, and let the fragments of M to be $f_1, f_2, \cdots, f_n$ each of which will piggyback an verification tag derived from a special hash chain before transmission. The hash chain called L is constructed as Figure 1.

As illustrated in Figure 1, each fragment f_i piggybacks a tag derived from the hash valve of its preceding fragment f_{i-1}, i.e., $f_i \parallel h_i$ and $h_i = H(f_i \| h_{i-1})$ initialized at $h_{1,1} = H(f_1)$ and $h_2 = H(f_2 \| h_{1,1})$. Here $H(\cdot)$ is a collision-resistant hash function. The final fragment carries a hash of the entire message. Note that $h_{1,2} = H(f_1 \| h_n)$ makes the hash chain to be an inseparable whole. This chain will lead to an unbreakable cyclic dependency among the fragments of a same bundle. Hash-linking each fragment to its successive fragment makes it difficult for attackers to excessively increase the complexity of filtering false fragments and reassembling the bundle at receivers by inserting or modifying fragments. This specially linked hash chain effectively prevents an adversary to insert malicious fragments without violating the hash verification. We assume that f_i^* is an injected fragment which takes the position, or index, of f_i. Then, f_i^* must satisfy the requirement of $h_i = H(f_i^* \| h_{i-1})$, $h(i-1) = H(f_{i-1} \| h_{i-2})$ and extending to the whole chain as such. Obviously, injection attacks will be found easily unless the adversary replaces all fragments of the target bundle.

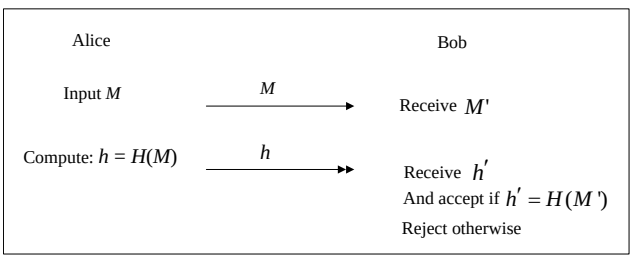

Fig. 1: Hash Chain Against Injection Attacks.

6.2 Fragment Authentication based on Hash Chain

Although the above-mentioned hash chain is effective in verifying fragments originated from the same source and resisting injection attacks, it is built on the assumption that all fragments are received. However, this assumption never holds true for space DTNs, because fragments are often lost and fragments of the same bundle are possibly routed via different paths. Both make the verification at intermediate routers and the receiver to fail. For example, f_i cannot be verified if its successive fragment is lost. Aiming to address this issue, we utilize erasure codes to recover the missed hashes on the chain L so that all received or relayed fragments should be verified even if some other fragments of the same bundle are lost or routed via other paths. Thereby the false ones inserted by any attackers can be efficiently filtered as early as possible. In addition, the signature is utilized to authenticate their common origin or source.

In the following algorithm, the couple (S, V) still denotes the digital signature and verification algorithms with the public key K_p and secret key K_s respectively.

Algorithm 5.

1) $\sigma \leftarrow S(K_s, h_n)$, i.e., sign on the hash h_n including the information of the entire bundle;
2) Split $\{\sigma \| H'\}$ into $\lceil pn \rceil$ blocks $B = \{B_1, B_2, \cdots, B_{\lceil pn \rceil}\}$ with same sizes, where $H' = \{h_{1,1}, h_2, h_3, \cdots, h_n, h_{1,2}\}$;
3) $\{B; \overline{B}\} \longleftarrow C_{\lceil pn \rceil, \lceil (1-p)n \rceil}(B)$;
4) Append each of the codewords of $\{B; \overline{B}\}$, represented by $T = \{\tau_1, \tau_2, \cdots, \tau_n\}$, to one fragment respectively.

In this algorithm, the encoding algorithm $C_{\lceil pn \rceil, \lceil (1-p)n \rceil}$ is same as Algorithms 1. The decoding and verification are presented in Algorithm 6.

It is important to note that the hash chain L itself is effective in verifying fragments of the same bundle and resisting injection attacks. Accordingly, if a scenario does not need origin authentication, the generation of signature will be omitted. Thus, only $(n+1)$ hash values are encoded in Algorithm 5. In this case, injected fragments will be found easily unless the adversary replaces all fragments of the target bundle. However, hash-linking each fragment to its successive fragment increases the complexity of attackers replacing all fragments and reassembling them into a chain, since the fragments of a bundle are forwarded in a multi-hop parallel manner and via different paths.

Algorithm 6.

1) $B \leftarrow D(T')$;
2) Derive σ and H' from B, here $H' = \{h_{1,1}, h_2, h_3, \cdots, h_n, h_{1,2}\}$;
3) For each fragment f_i' in the set R, if $H(f_i' \| h_{i-1}) \neq h_i$, remove this fragment f_i'. Here $h_i \in H'$ including the hash values on the recovered chain;
4) Verify $V(\sigma, h_n)$ to authenticate the source, here h_n is the last hash on the chain.

Here, $D(\cdot)$ denotes the decoding algorithm and T' is the set including all tags carried by the received fragments. The set $R = \{R_1, R_2, \cdots, R_n\}$ contains the received fragments, in which $R_i = \{f_i' \| \tau_i\}$ (The index i is f_i''s position in the original fragment chain.), or $R_i = \perp$ taking the position of the lost fragments on the original chain.

6.3 Performance Evaluation

Size of an authentication tag. Let h represent the length of a hash value and s the size of the signatures. The size of an individual authentication tag is expressed as $\lambda''(n,p)$ of both the number of bundle fragments and the average probability that all fragments of the same bundle are routed to the same intermediate router. According to Algorithm 5, we have

$$\lambda''(n,p) = \frac{s+(n+1)h}{\lceil pn \rceil}$$

which is much smaller than that of the scheme in Section 6, especially when n is large. This $\lambda''(n,p)$ presents the communication overhead and remains very small if either n is large or p is reasonably high.

To evaluate the communication overhead, we still refer to the simulation result of [12], which tells us that most fragments of the same bundle are transferred simultaneously in DTN, i.e., the average probability p is enough high in general. With $s = 128$ bytes and $h = 16$ bytes, we present a sampling of $\lambda''(n,p)$ for different p and n in Table 3. Note that we let the value of λ'' in the following table to be the lowest integer greater or equal to $\lambda''(n,p)$. It is quite obvious form Table 3 that the injection-resistant fragment authentication scheme based on hash chain L places a very low communication overhead to Bundle Protocol of DTNs. This approach dramatically reduces the bandwidth overhead as compared to the prior scheme based on vector space, especially when n the number of fragments is large. In addition, its computation costs are very low as well since it encodes and decodes only once respectively, which we present later.

TABLE 3: Overhead bytes per fragment in Algorithm 5.

p \ n	16	32	64	128	256	512	1024
0.9825	25	21	19	18	17	17	17
0.75	34	28	25	23	22	22	22
0.6	42	33	30	29	28	27	27

Computation costs. For the above scheme based on hash chain L, the main computation cost of a sender comes from generating $(n+1)$ hash values, one digital signature, and one time encoding. Here, the erasure coding operates on small authentication tags and thus is relatively efficient.

The computation cost of the recipient depends on the probability p. In an ideal situation, i.e., $p = 1$, the recipient just computes $(n+1)$ hashes and verifies a signature, since it does not need to decoding. If the recipient does not receive all the fragments of a bundle, it needs some additional decoding operations.

Discussion about pollution attacks. This scheme is able to eradicate traffic storms in that any injected false fragments will be filtered out. It is important to note that our schemes can effectively find the injected fragments and thereafter remove only the injected fragments rather than all the received fragments as [12], or the entire bundle as [5].

7 Conclusion

In this paper, by exploiting the well known erasure coding and the batch transmission characteristic of DTNs, we have given three candidate approaches for bundle authentication when bundle fragmentation service is needed in space internet. All of them are effective in resisting injection attacks and eradicating traffic storms. The first one places a surprisingly low communication overhead to a bundle, while it makes all received fragments of a bundle to be removed even if there is just one false or injected fragment. This is due to its failure in locating the false fragments. Considering this issue, we presented the second one which utilizes special geometrical construction of vector space to detect injected fragments and then only remove these injected ones. Aiming to further reduce communication overheads, we constructed a specially linked hash chain based on which injected fragments are quite easy to be filtered out only with a very small authentication tag appended to a fragment. The performance analysis verifies that our schemes, especially the third one, can effectively reduce the transmission cost as well as computational cost, and thus minimize the energy consumption due to fragment authentication. This advantage more highlights when the traffic load is reasonably high. The proposed schemes can be directly applied to space internetworking scenarios for solutions of bundle authentication when fragmentation service is needed.

References

[1] CCSDS, "Rationale, scenarios, and requirements for dtn in space," *Draft Informational Report, CCSDS. Draft Green Book*, 2010.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," *RFC4838, April*, 2007.

[3] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo, "Towards securing disruption-tolerant networking," *Nokia Research Center, Tech. Rep. NRC-TR-2007-007*, 2007.

[4] S. Farrell and V. Cahill, "Security considerations in space and delay tolerant networks," in *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*, pp. 8–pp, IEEE, 2006.

[5] S. Symington, S. Farrell, H. Weiss, and P. Lovell, "Bundle security protocol specification," *Work Progress, October*, 2007.

[6] R. Atkinson and S. Kent, "Security architecture for the internet protocol," *IETF, RFC2401*, 1998.

[7] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pp. 299–308, ACM, 2009.

[8] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 1, pp. 77–90, 2008.

[9] S. Farrell, S. Symington, H. Weiss, and P. Lovell, "Delay-tolerant networking security overview," *IRTF, DTN research group*, 2008.

[10] DTNRG, "Delay tolerant networking research group: dtninterest mailing list archive," *http://mailman.dtnrg.org/pipermail/dtninterest/2005-April/*, 2005.

[11] C. Partridge, "Authentication for fragments," in *HotNets-IV, The Fourth Workshop on Hot Topics in Networks*, 2005.
[12] H. Zhu, X. Lin, R. Lu, X. Shen, D. Xing, and Z. Cao, "An opportunistic batch bundle authentication scheme for energy constrained dtns," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.
[13] R. Blahut, *Theory and practice of error control codes*, vol. 126. Addison-Wesley Reading (Ma) etc., 1983.
[14] S. Lin and D. Costello Jr, "Error control coding: Fundamentals and applications, 1983," *Prentice-Hall, Inc. Englewood Cliffs*, vol. 3, pp. 2–3.
[15] V. Pless, *Introduction to the theory of error-correcting codes*, vol. 48. Wiley-Interscience, 1998.
[16] J. Van Lint, *Introduction to coding theory*, vol. 86. Springer Verlag, 1999.
[17] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.
[18] A. Pannetrat and R. Molva, "Efficient multicast packet authentication," in *Proc. of Network and Distributed System Security Symposium (NDSS03)*, 2003.
[19] M. Luby, M. Mitzenmacher, M. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 150–159, ACM, 1997.
[20] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *ACM SIGCOMM Computer Communication Review*, vol. 28, pp. 56–67, ACM, 1998.
[21] T. de Cola, H. Ernst, and M. Marchese, "Performance analysis of ccsds file delivery protocol and erasure coding techniques in deep space environments," *Computer Networks*, vol. 51, no. 14, pp. 4032–4049, 2007.
[22] I. Bisio, M. Marchese, and T. de Cola, "Congestion aware routing strategies for dtn-based interplanetary networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1–5, IEEE, 2008.
[23] T. de Cola and M. Marchese, "Joint use of custody transfer and erasure codes in dtn space networks: Benefits and shortcomings," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pp. 1–5, IEEE, 2010.
[24] T. de Cola, "A protocol design for incorporating erasure codes within ccsds: The case of dtn protocol architecture," in *Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc), 2010 5th*, pp. 68–73, IEEE, 2010.
[25] P. Tournoux, E. Lochin, J. Leguay, and J. Lacan, "Robust streaming in delay tolerant networks," in *Communications (ICC), 2010 IEEE International Conference on*, pp. 1–5, IEEE, 2010.
[26] CCSDS, "Erasure correcting codes for use in near earth and deep-space communications," in *CCSDS 131.5-O-0.2*, CCSDS, 2011.
[27] A. Kate, G. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pp. 504–513, Ieee, 2007.
[28] A. Krifa, C. Baraka, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*, pp. 260–268, IEEE, 2008.