Providing Authentication in Delay/Disruption Tolerant Networking (DTN) Environment

Enyenihi Johnson, Haitham Cruickshank, and Zhili Sun

Centre for Communication Systems Research (CCSR),
University of Surrey, Guildford, United Kingdom
{e.johnson,h.cruickshank,z.sun}@surrey.ac.uk

Abstract. DTN environment is characterized by intermittent connectivity, high/variable delay, heterogeneity, high error rate and asymmetric data rate amongst others. These characteristics accounts for the poor behavior of Internet protocols in this environment. To address these problems, DTN was conceived and designed together with specialized protocols to carry out its services. Its emergence called for a new concept in security that was considered at the design stage. The main aim of this paper is to propose a traditional cryptography based authentication scheme that does not depend on network administrator's availability during post network authentication communication and facilitates bundle processing by the recipient in the absence of connectivity. In this paper, we present and discuss the system model, the proposed credential and the propose authentication scheme. A simulation framework is developed for the implementation of the proposed and referenced schemes. From the simulation results, the proposed scheme was observed to be independent of network administrator's availability during post network authentication communication and facilitates bundle processing in the absence of connectivity.

Keywords: Security, Delay/Disruption Tolerant Networking (DTN), Authentication, Communication Satellite, Traditional Cryptography (TC), Public Key Infrastructure (PKI).

1 Introduction

Delay/Disruption Tolerant Networking (DTN) [1-4] is a networking architecture designed based on message switching mechanism to provide reliable communication in networking environments with long/variable delay, intermittent connectivity, high packet loss rates, heterogeneity and asymmetric data rate amongst others using store-and-forward operation. The poor behavior of existing internet protocols in DTN led to the design of specialized protocols like Bundle Protocol (BP) to provide DTN services as an overlay network. To facilitate interoperability between heterogeneous networks with different network characteristics with DTN, a new layer called Bundle Layer was introduced between the application layer and the transport layer of the internet protocol stack. The design of DTN and the protocol did not evolve without consideration for security which led to the development of relevant security documentations [5] [6] to address DTN-related security issues. The security

P. Pillai, R. Shorey, and E. Ferro (Eds.): PSATS 2012, LNICST 52, pp. 189–196, 2013.

documentations highlight security requirements, define design considerations, identify possible threats as well as open issues.

From the DTN security documentations and the security analysis in [7], the identified threats this work is designed to address are masquerading, modification and replay. To protect the DTN network from these threats, security solutions are required to support both hop-by-hop and end-to-end services as well as policy based routing. The policy based routing requires that an entity involve in DTN communication must be able to verify the authenticity of both the original sender and intermediate forwarder of the bundles (messages) as well as the integrity of the received bundles. The security blocks required to secure a transmitted bundle are defined and described in [6]. Farrell and Cahill in [7] while highlighting the security issues associated with designing the bundle to contain all the required keys and algorithm(s) for security processing, emphasized the need for an additional authorization checks with PKI as a possible solution. The use of PKI is associated with constraints like authorization server unavailability and limited capabilities of certain nodes for cryptographic operations.

The focus of this paper is to investigate how PKI concept can be used to provide an authentication solution that does not depend on server availability during post trust establishment network communication while taking the capabilities of the entities into consideration. The existing PKI based schemes in DTN are [1] and [8]. These schemes either use certificates and encourage large storage of security credentials or depend on server availability. We propose an authentication scheme that combines both asymmetric and symmetric cryptography to provide source authentication as well as message authentication and integrity. The contributions of this paper are summarized as follows: 1) Implementing traditional PKI to provide trust initiation/establishment; 2) introducing our proposed Authorization Pass (APass) as a substitute for PKI based certificate; 3) proposing a scheme that uses symmetric based Hash-based Message Authentication Code (HMAC) for hop-by-hop bundle authentication and integrity, and asymmetric based APass for source authentication; and 4) evaluating the performance of the proposed and reference schemes through simulation.

2 The System Model

The DTN environment in fig. 1 assigns a communication satellite to a zone with each zone having more than one heterogeneous regional network administered by a regional administrator (gateway). Inter-zonal communication is facilitated through satellite-to-satellite communication. Hierarchical routing is implemented with each satellite maintaining a routing table of all RAs within its zone of location and other satellites in the network. Each RA maintains a routing table containing all RAs within its zone and the designated zonal satellite for the zone. RAs in the common territories between two zones maintain routing tables of RAs in the two zones and the designated satellites. The routing tables are computed from the global topology table generated and updated by DTNNA each time an entity joins the network. The global topology table is accessible whenever DTNNA is online.

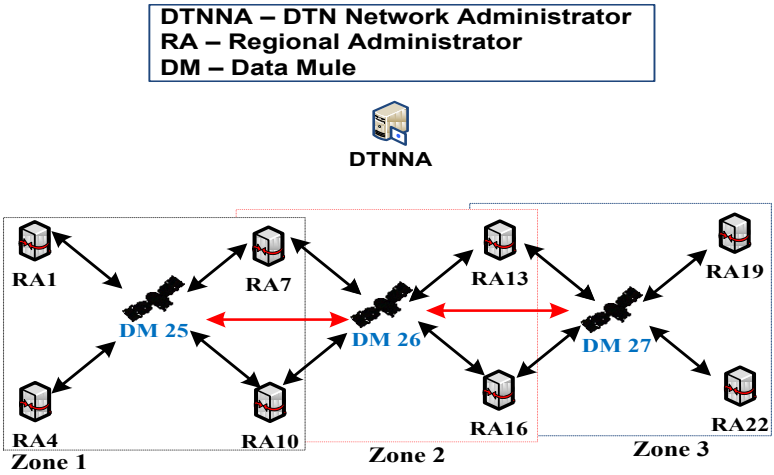

Fig. 1. DTN-Satellite Scenario

The use of satellite is aimed at facilitating bundle transmission between partition or remote networks that cannot communicate directly either due to obstruction or limited range of communication facility. Communication satellites with low computing and storage capabilities are now designed to provide data forwarding services. To minimize cost, Low Earth Orbit (LEO) satellites designed to accommodate limited number of brief contacts at a given time are employed. The limited number of brief contacts at a given time might result in discontinuous coverage. To address this issue, some LEO satellites are designed to provide DTN concept of store and forward services [9]. Security solutions should not be computationally heavy as well as encourage large storage of security credentials.

3 The Proposed APass

The PKI based digital certificate is considered too heavy for implementation in DTN environment. Its usage is associated with revocation and storage of Certificate Revocation List (CRL). It is also identified to provide partial trust management since it does not bind identity to access rights. To address the above issues, we proposed an asymmetric based APass introduced in [10] and shown in fig. 2 which is a modification of the digital certificate. The APass excludes entity's public key considered significantly large and incorporates *role* field to bind entity's identity to authorized action. To offer revocation flexibility and eliminates the storage of CRL, the *validity end* field in the APass is uniform for all APass issued irrespective of when the entities join the network while the *validity start* field differs.

An entity stops sending and receiving bundle when its own APass expires until it gets a new APass from DTNNA since other entities' APass are also assumed to have expired. DTNNA is assumed to know when a new APass must be generated for the entities because allowing the existing APass to expire in the absence of compromise will affect network communication.

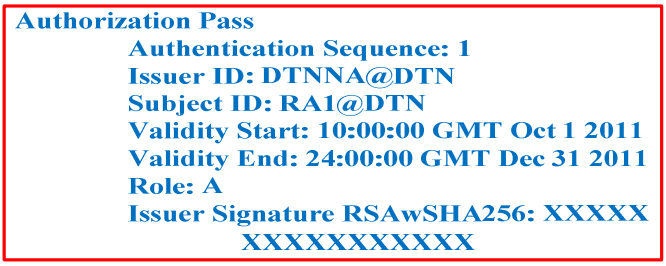

Fig. 2. The Proposed APass

4 Authentication Model

The authentication model is sub-divided into three phases of registration, network authentication and data exchange. The registration phase is facilitated by a public trusted entity called Registration Authority (RegAuth) which provides security information (secInfo) required for network authentication. The network authentication phase is facilitated by DTNNA and provides credential required for the data exchange phase. The data exchange phase is facilitated by network entities (RAs and DMs) using credentials obtained from DTNNA during network authentication for secure communication. We assume the entities cannot be compromise and the DMs are part of the DTNNA service providing network. Every RA is customized with pre-install initial public/private key pair, device identifier (devID) and RegAuth's public key access on activation. For every node (RA) manufactured the vendor provides RegAuth with node's devID and initial public key. Every node generates its public-private key pair. RAs must pass through the registration/network authentication phase to communicate in the DTN overlay network. RegAuth generates security information required for network authentication, while DTNNA generates trust information for post network authentication communication.

REGISTRATION/NETWORK AUTHENTICATION: The registration and network authentication phases are shown in fig. 3 below.

***Registration Phase*:**
RA1→RegAuth: $Pb_{RegAuth}\{rtj\ DTN\ network|devID_{RA1}\}$ (1)
RegAuth→RA1: $Pb_{RA1}\{devID_{RA1}|\ secInfo_{RA1}|ID_{DTNNA}|secInfo_{DTNNA}|Pb_{DTNNA}\}$ (2a)
RegAuth→DTNNA: $Pb_{DTNNA}\{nfa|devID_{RA1}|\ secInfo_{RA1}\}$ (2b)

***Network Authentication Phase*:**
RA1→DTNNA: $Pb_{DTNNA}\{authReq|devID_{RA1}|\ secInfo_{RA1}|Pb_{RA1}\}$ (3)
DTNNA→RA1: $Pb_{RA1}\{authConf|ID_{DTNNA}|secInfo_{DTNNA}|K_{dtn}|APass_{RA1}\}$ (4)

RSA encryption/decryption is used to secure communication with entity's public key used to encrypt and private key to decrypt. DTNNA and RA1 use security information from RegAuth for mutual authentication during network authentication where received secInfo must match the one obtained from RegAuth. RA1 uses Pb_{DTNNA} obtained in 2a to encrypt 3 while DTNNA uses Pb_{RA1} obtained in 3 to encrypt 4.

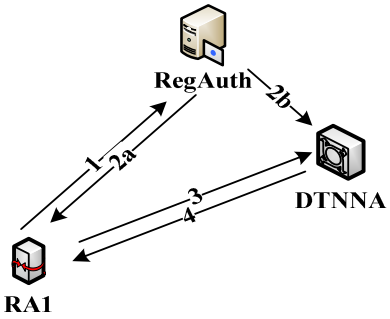

Fig. 3. Registration/Network Authentication Phase

***Notations*:** Pb_{RA1}, Pb_{RA7}, $Pb_{RegAuth}$, Pb_{DTNNA}: Public keys of RA1, RA7, RegAuth and DTNNA; $secInfo_{RA1}$, $secInfo_{DTNNA}$: Security information of RA1 and DTNNA; $APass_{RA1}$, $APass_{DM25}$: APass of RA1 and DM25; $devID_{RA1}$: Device identifier of RA1; K_{dtn}: Network-wide shared symmetric key; N_{RA1}: Nonce generated by RA1(randomly generated string); *Tstmp*: Bundle's timestamp; ID_{DTNNA}, ID_{RA1}, ID_{RA7}: Network identifiers of DTNNA, RA1 and RA7; CustID: Reserved for network identifiers of the data mules; $Pb_{RA7}\{Hello\}$: Encrypted payload processed only by RA7; $\{APass_{RA1}|N_{RA1}\}$: Access control block; $\{Tstmp \mid ID_{RA1} \mid CustID \mid ID_{RA1}\}$: Our primary bundle block; |: Concatenated operation; rtj: Request to join; nfa: Notification for authentication; authReq: Authentication request; authConf: Authentication confirmation.

DATA EXCHANGE PHASE: The entities (RAs/DMs) are assumed to have passed through registration/network authentication and in custody of $secInfo_{DTNNA}$ and K_{dtn} for HMAC computation/verification, their respective APass and Pb_{DTNNA} for DTNNA signature verification. The authenticated resource is the bundle and is designed to provide authentication, integrity and confidentiality. Our version of bundle and acknowledgement considering the first hop between RA1 and DM25 with RA7 as destination in fig. 1 is shown below.

(1). RA1 → DM25: $(Pb_{RA7}\{Hello\}|\{APass_{RA1}|N_{RA1}\}|\{Tstmp|ID_{RA1} \mid CustID \mid ID_{RA7}\})\cdot hmac$

(2). DM25 → RA1: $(isAccepted \mid APass_{DM25} \mid N_{RA4})\cdot hmac$

We assumed the entities are communicating for the first time after network authentication and Pb_{RA7} was obtained by RA1 when DTNNA was online. RA1 generates the bundle in (1), append hmac computed using K_{dtn} and $secInfo_{DTNNA}$ and send to DM25. DM25 upon receiving the bundle verifies the hmac by comparing the appended hmac with the one it computes using K_{dtn} and $secInfo_{DTNNA}$. The bundle content is only access if the appended and computed hmacs matches. DM25 identifies source/destination and ascertains bundle validity and freshness as well as APass signature verification using DTNNA's public key. If all conditions are met, DM25 accepts custody of the bundle and sends acknowledgement (2) appended with hmac to

RA1. DM25 then replaces the content of access control block with its APass ($APass_{DM25}$) and generated nonce (N_{DM25}) after which it inserts ID_{DM25} in the CustID field. It then computes hmac and append at the end of the bundle as discussed earlier before forwarding to RA7. RA7 accesses the bundle content after hmac verification to ascertain bundle validity and freshness as well as APass signature verification using DTNNA's public key. If all the conditions are met, it accepts custody and sends acknowledgement containing its APass ($APass_{RA7}$) and the received nonce N_{DM25} DM25. RA7 being the destination then proceeds with the decryption of $Pb_{RA7}\{Hello\}$ using its private key. Upon receiving acknowledgement, an entity verifies the appended hmac with the one it computes and verifies received APass without signature verification. The acknowledgement is confirmed as reply to a sent bundle if the received nonce matches the sent nonce after which a stored copy of the sent bundle is deleted. The recipient of a bundle only verifies the last forwarder on the assumption that the last forwarder must have verified the source/previous forwarder before accepting custody of the bundle.

COMPARATIVE ANALYSIS: The proposed scheme is compared with the TC based protocol by Asokan et al in [8] shown in fig. 4 below.

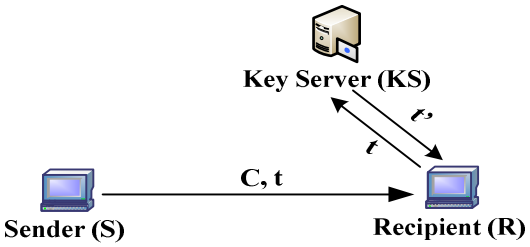

Fig. 4. Compared TC based Authentication Protocol

The protocol assumes that S knows the public key of KS and R identity Id_R prior to communication. KS shares a secret key with each entity and the protocol works as follows: (1) S generates a random secret key (k) and encrypt a message to have C. k and Id_R are then encrypted with KS's public key to have t which is sent together with C to R. (2) R forwards t to KS which decrypts (asymmetric) it with its private key to access the content and verify Id_R. KS then encrypts (symmetric) k retrieved from t with the key it shares with R (K_{SR}) to have t' forwarded to R. R upon receiving t' decrypts (symmetric) it with K_{SR} to retrieve k used to decrypt (symmetric) C to access the content. This protocol like the proposed scheme is PKI based combining symmetric and asymmetric algorithm and was design for analysis in DTN environment even though it was never validated through simulation. We modelled this protocol and implement it for DTN multi-hop communication. During network authentication phase, DTNNA generates different symmetric keys it shares with each RA/DM instead of K_{dtn} in the proposed scheme. S's APass and DTNNA's APass were included in t and t' to provide source authentication. The access control and primary blocks in the

bundle are encrypted with randomly generated k to form C. The random key k is used to encrypt/decrypt the acknowledgement.

5 Simulation and Performance Evaluation

To evaluate the performance of the proposed and compared schemes, the DTN-satellite scenario in fig. 1 was modeled in C++ using Microsoft Visual Studio 2008 with security integrated using Crypto++ Cryptographic Library. The modeled framework implements hierarchical routing with DTNNA the only source of updates. We used Sony VAIO laptop running Windows 7 with the following parameters: Intel CoreTM 2 Duo with T5500 Processor @ 1.66GHz speed, 2.50RAM, 120 GB/Go HDD and a system type of 32 bit Operating System.

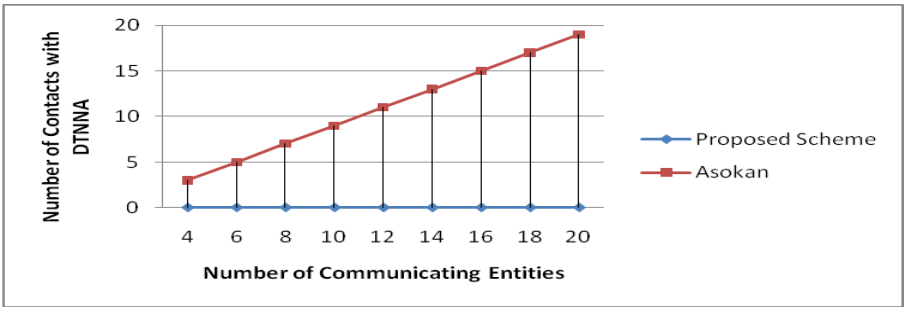

Fig. 5. Simulation result for Number of Contact with DTNNA

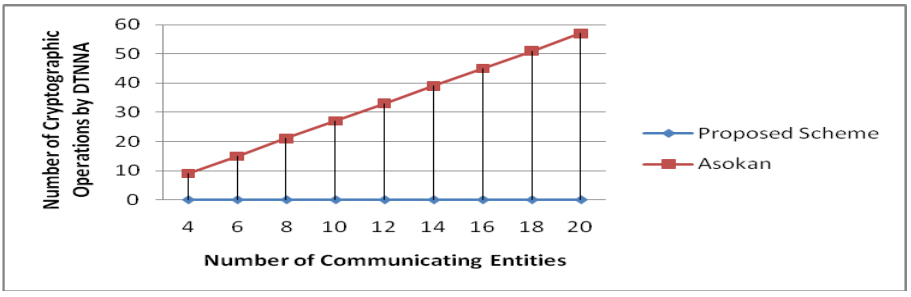

Fig. 6. Simulation result for Number of Cryptographic Operations by DTNNA

Fig. 5 shows the simulation result for the number of contact of the communicating entities with DTNNA while fig. 6 shows the number of cryptographic operations carried out by DTNNA during bundle transmission. The number of communicating entities include one sender, one destination and as many intermediate nodes (DM) as possible. While the proposed scheme has zero contact and zero cryptographic operations by DTNNA, the number of contacts and cryptographic operations by

DTNNA for Asokan increases with increase in the number of communicating entities. For every contact establish DTNNA carries out three cryptographic operations of public key decryption, signature verification and symmetric key encryption.

6 Conclusion

We modeled a PKI-based system model with online server (DTNNA) in C++ using Microsoft Visual Studio 2008 and Crypto++ Cryptographic Library for security integration. We presented and discussed the system model, the proposed APass, proposed authentication model and the TC based protocol in [8] considered for comparative analysis. From the simulation results in the last section, we confirmed that the TC based protocol in [8] depends on server availability and places more load on the server during multi-hop communication. The proposed scheme shows how traditional cryptographic techniques can be used judiciously to provide authentication and integrity solution that facilitates bundle processing by the recipient without depending on server availability. The proposed scheme suits the communication satellite because it facilitates onboard switching and processing in addition to providing lighter cryptographic operations and limited storage. The only issue with our scheme is that the sender will need contact with DTNNA for destination' public key ahead of communication or will have to store the keys.

References

1. Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets. In: SIGCOMM 2003, Karlsruhe, Germany, August 25-29 (2003)
2. McMahon, A., Farrell, S.: Delay- and Disruption-Tolerant Networking. IEEE Internet Computing 13(6), 82–87 (2009)
3. Cerf, V., et al.: Delay-Tolerant Networking Architecture. IETF Network Working Group RFC 4838 (2007)
4. Scott, K., Burleigh, S.: Bundle Protocol Specification. IETF Network Working Group RFC 5050 (2007)
5. Farrell, S., Symington, S., Weiss, H., Lovell, P.: Delay-Tolerant Networking Security Overview. IETF Internet Draft, draft-irtf-dtnrg-sec-overview-06 (2009)
6. Symington, S., Farrell, S., Weiss, H., Lovell, P.: Bundle Security Protocol Specification. IETF Internet Draft, draft-irtf-dtnrg-bundle-security-15 (2010)
7. Farrell, S., Cahill, V.: Security Considerations in Space and Delay Tolerant Networks. In: Second IEEE International Conference on Space Mission Challenges for Information Technology (2006)
8. Asokan, N., Kostiainen, K., Ginzboorg, J., Ott, J., Luo, C.: Towards Securing Disruption-Tolerant Networking. Nokia Research Centre, NRC-TR-2007-007 (2007)
9. Communication Satellite, http://en.wikipedia.org/wiki/Communications_satellite
10. Johnson, H., Cruickshank, H., Sun, Z.: Managing Access Control in Delay-/Disruption Tolerant Networking (DTN) Environment. In: 4th IEEE/IFIP International Conference on New Technology, Mobility and Security (2011)