Secure Cryptographic Mechanisms for Safeguarding Citizen Communications in the Presence of Tyranny

Soulaf Saab, Wassim Itani*, Ayman Kayssi, Ali Chehab, Cesar Ghali
Department of Electrical and Computer Engineering
American University of Beirut
Beirut 1107 2020, Lebanon
*Department of Electrical and Computer Engineering
Beirut Arab University
Beirut 1107 2809, Lebanon

Abstract

Despite the fast pace at which technology is spreading and communication networks are growing, some environments remain a challenge for communication service providers. In areas characterized by intermittent connectivity, long propagation delays, and high interference, connection-oriented communication protocols do not provide the optimal solution. These limitations increased the interest in developing delay-tolerant networks that can provide the basic means of communication without a strict demand for connectivity, network capacity, or regular mobility patterns of communicating nodes. A possible application of delay-tolerant networks is communication in the presence of oppressive governments. In this paper, we attempt to develop a secure delay-tolerant network system that enables citizens to communicate freely in an environment where public communication methods, such as mobile networks and the Internet, are intercepted and used by the authorities to monitor civilian activities. The proposed system is composed of several disconnected zones in which data marshals between Private Key Generators and normal nodes in different zones through mobile gateway nodes that carry messages between those zones. We simulate the system using GrooveNet, and describe the effects of different parameters on overall performance and security.

1. Introduction

In order to establish a communication network, we need to make that its nodes are able to communicate. Mobile network operators, for example, create coverage plans to distribute their sites in a way that enables a subscriber to place phone calls to any location within the mobile network. In challenged environments, however, it is not always possible to guarantee continuous connectivity for networking entities. In mountainous terrains, rural plantations, and war zones, communication using wireless networks is prone to disruption and long delays. This increased the interest in developing delay-tolerant networks (DTNs) that do not have strict requirements for connectivity, network capacity and mobility patterns of communicating nodes.

Delay-Tolerant Networking was first considered for deep-space communications, due to the huge propagation delays experienced during data transfer between different planets of the solar system [1]. Interplanetary environments are also subject to high bit error rates and long disconnections, which added the term "disruption" to this research field. DTN deployment was later extended to include some terrestrial networks characterized by intermittent connectivity, such as wireless networks in rural areas, military networks, and vehicular networks [14]. However, no "killer" DTN application has been identified yet. DTN models and architectures are being developed for several possible applications including low-cost, minimum infrastructure social networking.

Possible DTN applications include e-governance, telemedicine and citizen journalism in rural areas [3] which impose strict security and privacy requirements. Therefore, DTNs are expected to consider providing the necessary security services to applications and subscribers.

Securing a DTN has to take into consideration the various challenges encountered in such networks. The main challenge is the lack of end-to-end sustained connectivity, which imposes large communication delays and pushes network entities towards opportunistic communication over the intermittent links [4]. DTN nodes are usually characterized by high mobility with undetermined mobility patterns [1]. These nodes could range from laptop computers to tablets, cell phones, or even tiny sensors. Thus, storage capacity and battery life vary considerably among different types of nodes, and could become scarce in some of them. These challenges, particular to the disruption-prone environment of DTNs, render traditional security mechanisms inapplicable [5].

Similar to other networks, DTNs are subject to several security threats. Denial of Service (DoS) attacks could happen at any layer of the DTN. Malicious nodes could launch masquerading attacks by pretending to be valid nodes and intercepting communications.

Unauthorized access to the DTN resources might result in data modification and tampering, and on a less critical scale, eavesdropping. Even authorized access could be harmful if not controlled by strict permission rules and access priorities. DTN resources could be exhausted by valid nodes if they enter a loop of data generation that can eventually flood the network.

With all the challenges and threats at hand, securing a DTN enforces a set of requirements, which could vary depending on the application and the target security level. Security measures are to be provided on a hop-by-hop and end-to-end basis, and between intermediary nodes whose role is to temporarily store and then forward messages. A message sender should be authenticated through its digital signature or a message authentication code (MAC). Data encryption should be used to provide confidentiality of exchanged messages, such that only intended recipients can decrypt these messages. Since the possibility of data tampering always exists, integrity checks should be carried out to detect modifications of message content [1]. If messages of large sizes need to be transmitted over a DTN, fragmentation is usually implemented, which makes authentication and integrity validation more complicated [5].

Access control mechanisms should be applied so that only authorized entities could exchange messages, and in a manner that does not exhaust network resources. For authorization, integrity check, and confidentiality assurance, a proper key management scheme should be used which takes into account the disconnected nature of a DTN. Due to their limited storage and processing capacities, some DTN nodes might need to make simple decisions regarding what messages to store/forward. Thus, policy-based routing/forwarding becomes more important with the scarcity of network resources.

Privacy of DTN communications implies keeping both the message content and the identities of the original sender and the intended recipient confidential. The disconnected nature of DTNs enforces the "take what you can get" routing strategy on infrastructure nodes, and this greatly limits routing options and makes confirmation difficult to obtain. The opportunistic connections and variable delays make traditional anonymity techniques, such as source routing, unsuitable for DTNs [3]. Moreover, existing privacy-enabling solutions, such as successive layered encryption (or Onion Routing [7]), require knowledge of the network topology and inflict additional storage delays and overhead. Consequently, they cannot be used in DTNs.

The aim of this paper is to make use of the DTN's flexible architecture in order to provide secure means of communication within an environment that is still widespread in several parts of the world nowadays: communities under the control of oppressive rulers/authorities. The basic scenario addressed in our model considers a group of citizens who need to communicate short messages, of extremely confidential nature, having only intermittent connectivity to servers and peer nodes. Basically, a DTN node would either be a mobile phone or a laptop. At the same time, for personal safety reasons, members of this group need to hide their identities from possible interceptors or compromised nodes. Therefore, anonymity is a critical security requirement in this case.

The rest of this paper is organized as follows: In Section 2, we briefly discuss the major DTN security protocols presented in the literature. Section 3 presents the proposed system design and architecture. This includes a description of the DTN model assumed in this work, the communication scenarios adopted, and the encryption schemes employed. In Section 4, a simulation of the proposed design is presented and analyzed. Conclusions and future extensions are provided in Section 5.

2. Related Work

In [6], Farrell and Cahill recommend the following policy controls specifically for DTNs: time-to-live controls, "strange routes" (such as loops) controls, controls that handle information about resource constraints in the DTN such as neighbor's storage capacity, and fragmentation controls.

The Delay-Tolerant Networking Research Group (DTNRG) defines in [8] a set of goals in order to overcome the security threats present in a DTN. These goals include:

- Promptly preventing unauthorized applications from using the DTN to carry or store data or control the DTN infrastructure.
- Preventing otherwise authorized applications from sending bundles at a rate or class of service for which they lack permission.
- Promptly discarding bundles that are damaged or improperly modified in transit.
- Promptly detecting and de-authorizing compromised entities.

In the definition of security services for DTNs, senders and recipients of bundles are distinguished from senders and recipients of security service applications. The Bundle Security Protocol Specification [10] defines three security headers that can be used to provide different security services to message bundles:

- BAH (Bundle Authentication Header) adds a MAC or a signature to the bundle to authenticate a single hop communication.
- PSH (Payload Security Header) adds a MAC or signature to a bundle to provide end-to-end authentication.
- CH (Confidentiality Header) to encapsulate encrypted payload [5].

The authors in [15] introduce a Bundle layer security protocol, SEGURA, for authentication and integrity check of message bundles in DTNs. Their protocol employs probabilistic set membership constructs to amortize the cryptographic functionality of traditional authentication mechanisms such as MACs, hash trees, authentication graphs, and digital

signatures in relatively small-sized data structures. The SEGURA integrity enforcement mechanism gives intermediate DTN routers and gateways the ability to verify the integrity of bundles without employing computationally expensive public-key operations.

As an alternative to the traditional Public Key Infrastructure (PKI), Identity-Based Cryptography (IBC) enables message encryption and signature verification using a public identifier, such as a name or an email address of the target recipient, as a key [5]. An IBC system consists mainly of:

- Principals (P) as message senders and recipients.
- Private Key Generator (PKG) as trusted third party.

Asokan et al. [5] discuss the applicability of IBC for DTN communications, and compare it to the traditional cryptographic methods.

Because of their disconnected nature, nodes in a DTN cannot constantly access servers to fetch public keys and check certificate revocation lists (CRLs). Therefore, a traditional PKI is not suitable to provide security for DTNs, and IBC presents a more suitable solution. Furthermore, Hierarchical IBC (HIBC) extends IBC by establishing a cooperative hierarchy of PKGs consisting of top-level or root PKG and domain PKGs. Seth and Keshav present in [4] a security architecture based on HIBC for DTNs, and use it to create secure channels, provide mutual authentication, and achieve key revocation for disconnected nodes. The basic elements of the proposed architecture are:

- Region: collection of mutually-reachable DTN routers.
- Gateway: DTN router with interfaces to more than one region.
- Custodian: DTN router acting as always-available proxy for intermittently-connected hosts.
- Local DTN router: DTN router communicating directly with an endpoint (could be a custodian).

Seth and Keshav also propose mechanisms for mutual authentication between a disconnected node and a local DTN router.

Kate et al [3] use IBC to create a security infrastructure for DTNs. They introduce a new anonymous authentication protocol, which they use as the main building block for the first anonymous communication solution for intermittently-connected networks. Their DTN security architecture is based on the Sakai-Ohgishi-Kasahara (SOK) key agreement protocol [11] in a Boneh-Franklin identity-based encryption (BF-IBE) setup [12].

Kate et al. also argue in [3] that their solution has two main advantages compared to Seth and Keshav's solution for end-to-end security:

- The mutual authentication scheme is more efficient and does not need to be interactive.
- The secure data transfer mechanism between different users of the same disconnected network is more efficient.

Moreover, in [13] the authors present the Single Territory Reliable Anonymous Protocol (STRAP-k), a scheme to ensure reliable delivery of messages to all multicast receivers in DTNs, by making use of non-multicast nodes to reduce delivery latency. Parameter k designates the maximum number of non-multicast nodes that can be used in delivering a message. This reliability scheme is then extended to provide anonymity of multicast receivers in the group. The authors assume the following:

- All non-multicast nodes are not to be trusted.
- One or more non-multicast nodes in the network might be malicious.
- All group members follow the protocol and do not misbehave. If compromised, however, they could potentially reveal the identity of other multicast nodes.
- Nodes use pseudonyms instead of their real identifiers in the multicast message distribution process. These pseudonyms must change dynamically, and should be statistically unique and cryptographically verifiable.

The anonymity part of this scheme ensures that, for a certain node *i*, no other node in the network can predict whether it is a multicast receiver from previous message interactions. Multicast receiver identities are kept secret from the custodial nodes that provide guarantees to such receivers. In addition, the anonymous authentication scheme enables a node *i* to authenticate a multicast group member *j* while keeping their identities hidden.

3. System Design

This research work aims to provide robust and secure means of communication for oppressed citizens through the development of a suitable DTN model. The scenario under study consists of a group of citizens under the rule of oppressive authorities. Mobile networks and other public communication systems are intercepted and used by the oppressors to closely observe citizens' movement and subdue any conspiracy or resistance. Civilians whose lives and freedom are threatened by such regimes try to communicate via other channels, unknown to the authorities. However, they cannot dismiss the possibility of being discovered. Therefore, in addition to interacting over a secret network, they need to make sure the exchanged information remains confidential and the communicating members remain anonymous.

3.1 DTN Model

The DTN model proposed for this scenario is composed of disconnected cells, which are typically

public areas that are usually crowded with families, friends, and work colleagues. Members of resistant groups can go to such places without raising any suspicion, and they can communicate with fellow members without even knowing their true identities. A cell can be a mall, a café, a park, or a library where it is normal to see people using their cell phones or laptops.

Since the true identities of the DTN members must remain secret, pseudonyms are used during message exchange. Due to the fact that cells are disconnected and there is no network infrastructure to ensure continuous connectivity, Identity-Based Encryption is used for secure communication within the cells.

During the system initialization phase, subscription credentials are stored in sealed USB sticks and manually distributed by authorized agents to members who want to be part of this network [5]. These credentials will be used by each member to connect to a zone PKG for the first time and establish a trust-based relationship.

Within each cell, a PKG handles the distribution and update of Private Keys (Pr) of all the DTN members in that cell. The PKG also maintains a database containing the Public IDs (Pu) and updated status of the nodes within its cell. PKGs of all cells securely communicate with each other to synchronize their databases.

Members of a DTN use the wireless connection capabilities of their cell phones or their laptops, such as Wi-Fi [19], to communicate inside a cell. Based on the IBE cryptosystem, DTN nodes use their Public IDs during communication. These IDs are associated with a timestamp to ensure the freshness of the used keys. The nodes collaborate to send messages to their destinations by rebroadcasting received packets within a cell. Moreover, a DTN node may travel from one cell to another, and might be assigned by the PKG to carry messages of other nodes to the new cell. This node is called a "Mobile Gateway".

Gateways are ordinary nodes that have enough storage and processing capabilities to store and forward messages of other nodes, and that are believed to be reliable and trustworthy by the PKG. Identities of the chosen gateways are not known to nodes and other gateways within their cell or anywhere in the DTN.

To send a message, a node encrypts it and broadcasts it within its cell. If the intended receiver is present in that cell, it can open and read the message. Otherwise, the message is carried by a traveling gateway node to another location closer to its destination.

A sender knows the Public ID of the intended recipient of its message but is not aware of the recipient's location. Collaboration between the different PKGs enables them to locate the recipient and arrange for the message to reach the destination through one or several gateways. Figure 1 presents a schematic diagram of the proposed DTN system design presented above.

For the proposed model, we assume the following:

- The PKG is trusted by the DTN nodes and acts as a centralized router.
- Members of the DTN are capable of broadcasting their messages wirelessly.
- Members of the DTN know the locations of the different cells, but do not know who the PKG is.
- Members of the DTN know the Public IDs of one another and use them for messages exchange, but they do not personally know the owners of these IDs.
- PKGs have established communication channels among each other, and they communicate with cell nodes wirelessly.

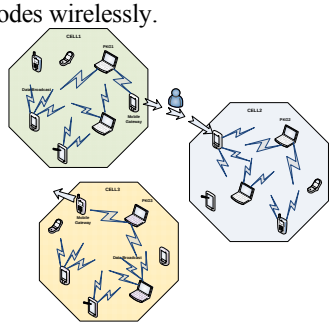

Figure 1. DTN Network Model

3.2 Communication Scenario

We consider a sender S who needs to send a message M to a recipient D. For the message to be sent, S must be within a PKG zone, and it should have a fresh Private Key (Pr_S) so that it can decrypt any messages it might receive during the validity period (which could be a day, for example). This requires that S connects to the PKG at least once every validity period and requests a new Pr. In order to do so, S needs to know the Public ID of the PKG.

Public IDs of PKGs can be watermarked over the radio channel of a local broadcasting station. Interested nodes of the DTN would be listening to this channel and waiting to get the ID of the PKG where they will be located. PKG IDs are refreshed every ΔT, and the private key generation function is not restricted to particular nodes. In order to decrease the possibility of compromise, new nodes can be selected to replace existing PKGs, based on their reputation. PKG replacement remains transparent to the remaining DTN nodes.

Private Key Request Message: Being the authorized user of the Public ID Pu_S, S connects to the PKG and requests a corresponding new Private Key. The request of S is encrypted using the Public ID of the PKG (Pu_{PKG}). The encryption function used (Enc) is

an identity-based encryption function that will be described later.

$$Req\_Pr_S = Enc\ Pu_{PKG}\ [\ Identifier_S,\ Pu_S||timestamp,\ Pu_D,\ K\ ]$$

The Identifier is a secret parameter associated with the Public ID of S; it verifies that S is authorized to use Pu_S. Pu_S is needed by the PKG to generate the private keys of S. By associating Pu_S with a timestamp, we ensure that Pr_S is fresh. As mentioned earlier, this serves to replace the certificate revocation lists (CRLs) of traditional encryption schemes.

Pu_D is required by the PKG to assign the message ID (ID_M), which is associated with the location of D. The PKG knows this location through the cooperation with the other PKGs and by maintaining synchronized databases containing updated node locations.

Since S does not have any private key yet, it generates a session key (K) for the initial communication with the PKG. The PKG uses K to encrypt Pr_S and to send it over a secure channel to S.

Private Key Reply Message: Upon receiving the Req_Pr_S, the PKG decrypts it using its private key. If S has the necessary credentials and the timestamp is valid (corresponds to the current key validity period), the PKG generates Pr_S and updates the entry corresponding to S in its database (see Figure 2). Using Pu_D, PKG searches for the location of D in the main database and generates ID_M accordingly.

Node S
Pu_S
Fresh Pr_S
Current Location
Message ID(s)
Status

Figure 2. Node Database Information

The main database contains information about all active nodes during a key validity period. This information is shared among all PKGs and synchronized periodically every ΔT_{Syn} over a secure communication channel. Moreover, all PKG databases are assumed to be protected using a deniable encryption technique (e.g. TrueCrypt) that creates a hidden disk partition containing the stored data. If compromised, this hidden partition can be decrypted to reveal an empty volume. The use of deniable encryption in the proposed communication scenario is explained later.

The PKG then encrypts Pr_S using the shared session key and sends it directly to S. We assume that the transmission range of the PKG is large enough to cover the whole DTN cell area, and therefore, we are certain that replies from the PKG arrive at their destination. Sym_Enc is a classical symmetric key encryption function:

$$Rep\_Pr_S = Sym\_Enc_K\ [Pr_S]$$

For the message of S to reach its destination, PKG creates gateway tickets that help gateway nodes identify which messages they should carry to their next location. The DTN nodes are mobile and have varying resources. Because a PKG needs to know what nodes are present within its cell and what capabilities they have, it requires the nodes to send status update messages frequently and periodically (every ΔT_1).

Heartbeat Message: Every ΔT_1, each node within a PKG zone sends a status update to the PKG, containing the following: presence flag (I'm here/I'm leaving), available battery resources, available storage capacity, processing capacity, communication range, and next location (in case presence flag indicates "I'm leaving").

The presence flag informs the PKG of the nodes present within its cell, and the nodes leaving the cell. This is how a possible recipient node location is identified (through PKG cooperation). Moreover, a leaving node has to declare its next location, whether it is another cell within the DTN, or totally outside the DTN. In the former case, the PKG can assign this node to carry a number of messages from other nodes in its cell to the new cell. These messages could be intended for recipients in the new cell or in another cell close by, and in this case, they need to be forwarded again by another gateway.

PKG uses the information it receives from the Heartbeat messages to decide which nodes can act as mobile gateways, and which mobile gateway should carry each outgoing message (depending on the previously assigned message IDs).

Accordingly, PKG sends updated gateway tickets to all the active nodes within the cell. When a node broadcasts a message intended for a remote recipient, the corresponding mobile gateway can store and forward it. Therefore, PKG sends the following tickets to S, so that it can send messages to D:

Gateway Tickets:

For message M_1: $R_1||Enc_{PuGi}\ (ID_{M1}||R_1)$
For message M_2: $R_2||Enc_{PuGi}\ (ID_{M2}||R_2)$
For message M_n: $R_n||Enc_{PuGi}\ (ID_{Mn}||R_n)$

Where R is a random number generated by the PKG to camouflage the ticket structure. When the mobile gateway decrypts its tickets, it compares the random number obtained from decryption with that

appended to the ticket to confirm that this ticket belongs to it. If S wants to send messages to another destination D', then it has to communicate with the PKG to update the destination and get new tickets.

Destination Update Message: S sends to PKG:

$$Dest_S = Enc_{PuPKG}[Pu_{D'}]$$

Consequently, in the next heartbeat reply, PKG sends S its updated tickets.

For message M'_1: $R'_1||Enc_{PuGj}(ID_{M'1}||R'_1)$

For message M'_2: $R'_2||Enc_{PuGj}(ID_{M'2}||R'_2)$

For message M'_n: $R'_n||Enc_{PuGj}(ID_{M'n}||R'_n)$

Message Communications: S wants to send M_1 to D. It uses the first ticket it received from the PKG and appends to it the encryption of M_1 along with its own ID, then broadcasts the whole ciphertext (C_1) in its cell.

$$C_1 = R_1||Enc_{PuGi}(ID_{M1}||R_1)||Den\_Enc_{PuD}(S_{ID}, M_1)$$

Den_Enc is a deniable encryption function that we use here to further protect the sender and the receiver in case they were compromised. We describe this function in more details in the next section.

When gateway G_i receives C_1, it performs the following steps:

- Parses the received ciphertext to get the random number R and two encrypted blocks (1 and 2).
- Decrypts the first block (1) using its private key (Pr_{Gi}) to get ID_{M1} and another random number r.
- Compares R to r to confirm that the ticket belongs to it.
- Stores ID_{M1} with the second encrypted block (2).

Gi, carrying the messages from S and some other senders, travels to its second indicated location (PKG'). Gi sends a heartbeat message to PKG' to announce its arrival. Then it sends a forwarding destination request message to PKG', containing all message IDs that were carried from PKG zone, in order to get the proper ticket to forward each of these messages:

$$Req\_FwDest_{Gi} = Enc_{PuPKG'}[ID_1||ID_2||...||ID_n]$$

Upon receiving the heartbeat message and the forwarding destination request message from Gi, PKG' sends a reply message containing the corresponding tickets according to the different message IDs:

$$Rep\_FwDest_{Gi} = Enc_{PuGi}$$
$$[ID_1||ticket_1||ID_2||ticket_2||...||ID_n||ticket_n]$$

This forwarding process is repeated until each of the carried messages reaches its destination D. The location of D is known through the heartbeat message it sends to the PKG in its zone, and through the data synchronization between the different PKGs. If D is available at one of the network zones, it should finally receive its message from the original sender S:

$$C = R||Enc_{PuD}(ID_M||R)||Den\_Enc_{PuD}(S_{ID}, M)$$

D parses the received ciphertext into a random number and two encrypted blocks. D decrypts the first block successfully and realizes that the ticket actually belongs to it. If D is not traveling to another zone, and is therefore not a mobile gateway, it knows that the message was sent to it and decrypts it. However, if D is actually travelling, then at one point it sends the message ID to the PKG to get the corresponding ticket. Knowing that D is the intended recipient of this message, PKG sends back a "receive permission" signal instead of a ticket. Thus, D decrypts the second block using its private key and gets S_{ID} and the original message M.

D sends a "receive acknowledge" message to its PKG. This enables the different PKGs to establish the gateway reputation database since a successfully received message means that the assigned gateways have been reliable and performed their tasks as required. The gateway reputation database may be used, for example, in a PKG nomination and selection process.

3.3 Alternative Communication Option

We have assumed so far that PKGs have established communication channels between them in order to synchronize their databases and determine the path each outbound message should take. However, in such a disconnected networking environment, it might not be possible to guarantee such communication.

Therefore, another option would be to limit the role of the PKG to just private key generation. Nodes traveling out of their zones will carry with them *all* the messages they have received through broadcast from peer nodes. They would again broadcast these messages in the new zones they travel to.

Hence, messages would circulate between zones with a good probability of eventually reaching their intended destinations. However, this would come at the expense of network resources and performance.

Broadcasted messages would flood the network and DTN nodes would have to do more processing. Since the number of communicating nodes in such

an environment is supposed to be relatively small, the impact on network performance should be tolerable. However, we would still have to deal with a larger latency.

3.4 Encryption Schemes

To ensure security and privacy of communication, the following encryption methods are used:

IBE Cryptosystem: Consider the Public ID of sender Alice, Pu_{Alice}: alice@dtn.com and the Public ID of receiver Bob, Pu_{Bob}: bob@dtn.com.

Setup: The PKG creates its private and public key pair, which we denote by S_{PKG} *(Master Key)* and *PP* (public/global parameters) respectively. *PP* is a public parameter known to all the principals (senders/receivers) within the PKG zone [20].

$$PP = p \times q, \text{ where } p \text{ and } q \text{ are two large prime numbers, } p \neq q$$

PKG generates the private keys of the principals as follows [21]:

$$Pr = ID^{-1} \bmod \phi(PP)$$

$$\phi(PP) = (p - 1) \times (q - 1)$$

$$ID = H(principal@dtn.com||timestamp)$$

where H is a SHA hash function [22].

Encryption (Enc): Alice encrypts a message Msg to Bob as follows:

$$Ciph = Enc(Msg) = Msg^{ID} \bmod PP$$

Decryption (Dec): Upon receiving the encrypted data from Alice, Bob requests his private key from the PKG using his Public ID appended to a timestamp (bob@dtn.com||TimeStamp). He then decrypts the received data as follows:

$$Msg = Dec(Ciph) = Ciph^{PrBob} \bmod PP$$

Deniable Encryption: The IBE cryptosystem described above is used for sending requests and receiving replies. However, when the actual message Msg is sent, an extra level of security is added to make sure Msg is not revealed in case the sender or the receiver are compromised. We implement a "deniable encryption" mechanism, represented by function "Den_Enc".

We used the deniable encryption algorithm proposed in [23]. This algorithm is based on the encryption of several interleaved message blocks; some of these blocks belong to the real message, while the others belong to random messages that are used as a decoy. Blocks of bogus data are randomly inserted, but the encrypted output includes additional information to determine whether a particular interleaved block is part of the original message or not. This algorithm was implemented by the author of [24] and its deniability was successfully tested.

In order to further protect the DTN data stored in the PKG databases, we can use an existing deniable encryption system, such as TrueCrypt [25], which provides deniability to stored data through the creation of hidden volumes. Within the free space of a storage volume, a user creates an encrypted outer volume to store semi-sensitive data. This data can be revealed to a coercer that forces the user to hand out the password. Within the remaining space of the outer volume, the user can create a hidden volume with a totally different password to store the truly sensitive data.

4. Simulation and Results

We have used GrooveNet [26] to demonstrate the basic communication process in the proposed DTN scenario. In our simulation, we distinguished between a PKG node (server) and a normal DTN node, where the former is fixed while the others are mobile.

We demonstrate the status update process in which each normal node broadcasts its status update, hoping to reach the PKG by getting rebroadcasted by other local nodes. For every received status update message, the PKG replies directly to the node that sent it, and we are certain that the reply is received since the transmission range of the PKG covers (actually defines) the whole zone. The simulation includes separate zones. We auto-generate several normal nodes and one PKG in each zone. The normal nodes are bounded within the PKG zone by a MAX_DISTANCE parameter that we set in the simulator. This parameter defines the maximum distance a node can travel from its starting point until it returns.

We also separately auto-generate a Mobile Gateway node that starts in one zone and moves to another, communicating with the PKGs of both zones. The Mobile Gateway node is a normal node, but with a greater value for MAX_DISTANCE that enables it to move longer distances in order to reach another zone. Since it is generated separately, we can define a different mobility model for this node in order to better control its movement between zones.

4.1 Single DTN Zone

For the first set of simulations, we used some fixed settings throughout a group of eight test cases (see Table 1), in particular: Communication

model is SimpleCommModel, Link layer model is SimpleLinkModel, Mobility model is UniformSpeedModel for normal nodes and FixedMobilityModel for PKG, Trip is SightseeingModel, Simulation duration is 3 minutes, and Reception range of normal nodes is 200m.

We define new performance parameters to describe these results and compare them with other test cases:

PKG Reply Rate (PRR). The DTN nodes have a limited transmission power and not all broadcasted messages reach the PKG. Thus, we define the PKG Reply Rate during one Status Update Period as:

PRR = number of PKG replies / number of normal nodes per zone

PKG Reply Delay (PRD). For the PKG, the average delay between the time it receives a Status Update Message and the time it replies to this message is:

PRD = Σ reply delay / number of nodes that get replies

PRD is calculated over one Status Update Period.

Time to First Reception of Heartbeat (T_{Rx}). To update its status, each normal node retransmits its Status Update Message (Heartbeat Message) over a one-second interval, to increase its chances of being received by another node that is in range, or by the PKG itself. Let T_{Rx} be the time it takes a Status Update Message to be received for the first time by a DTN node, calculated from the original transmission time of this message. Thus,

T_{Rx} = *Σ time till first reception of a Status Update Message / number of nodes*

Delay of Message Rebroadcast (δT_{Rb}). When a node receives a broadcas Status Update Message from a peer node, it rebroadcasts this message after a certain delay.

δT_{Rb} = *Σ rebroadcast delay / number of rebroadcasts per Status Update Period*

We summarize the results of all 8 test cases in Table2.

Reception Range of PKG: We assume originally that the PKG can reach any node within its zone. Therefore, whenever the PKG receives a request, it replies directly to the original sender. However, the issue here is whether or not the PKG receives the broadcast requests. This depends on the Reception Range of the PKG. When we increase this range, the PKG is able to receive requests from a greater number of nodes, and therefore, the PRR increases. This is demonstrated in test case 4, where the reception range is increased from 70m to 100m.

Table 1. Test Cases for Simulating a Single DTN Zone.

Param	Case1	Case2	Case3	Case4	Case5	Case6	Case7	Case8
# of normal nodes	10	5	5	5	5	5	5	5
Max trip distance (m)	50	50	100	100	100	100	100	100
Reception range of PKG (m)	70	70	70	100	100	100	100	100
Uniform speed (mph)	25 - 35	25 - 35	25 - 35	25 - 35	25 - 35	2 - 5	2 - 5	2 - 5
Physical Layer Model	Simple Phys	Simple Phys	Simple Phys	Simple Phys	Collision Phys	Simple Phys	Simple Phys	Simple Phys
SNR/ RSSI	0/ -1	0/ -1	0/ -1	0/ -1	0/ -1	0/-1	40/-55	0/-1
Status update period (sec)	42.7	42.7	42.7	42.7	42.7	42.7	42.7	85.4

Table 2. Results of Test Cases for Simulating a Single DTN Zone.

Performance Parameter	Case1	Case2	Case3	Case4	Case5	Case6	Case7	Case8
PRR	33.3%	66.6%	0%	66.6%	73%	100%	33.3%	100%
PRD (in s)	0.077	0.077	-	0.382	0.096	0.08	0.106	≈ 0
T_{Rx} (in s)	0.649	0.725	0.74	0.738	0.76	0.822	0.806	0.834
δT_{Rb} (in s)	0.462	0.733	0.404	0.808	0.428	0.517	0.035	0.35

Physical Layer Model. This model represents the physical communication medium; wireless in our case. The SimplePhys model that we use in all but one of our test cases performs only a simple check to enforce the communication range. However, the CollisionPhysModel has the additional functionality of collision modeling. It drops packets if concurrent transmissions are heard within the transmission radius of the receiver. In test case 5, we use CollisionPhysModel instead of SimplePhysModel, and we notice an improvement in all the performance parameters.

Node Speed. As mentioned previously, the simulated DTN nodes are supposed to be within or moving around a small geographical area. Therefore, their speed should be negligible compared to that of a car. When we reduce the speed from 25mph to 2mph, we increase the chances of a message to be received by a peer node or by the PKG, and this is clear in the increased PRR in test cases 6 and 8. What makes a difference in this case is the original distribution of the nodes, whereby a node could remain within the reception range of the PKG or a peer node throughout the simulation time.

Communication Delays: In the considered DTN model, propagation delays within one DTN zone are negligible (in the range of few microseconds). This is because the transmission bit rate of a Wi-Fi device is in the order of several to tens of Mbps, whereas the propagation distances are around 100m per hop. Thus, the communication delays are caused mainly by the limited reception ranges of the PKG and the other nodes. Thus, a node rebroadcasts a message over a certain interval (1 second in our simulation), hoping that in the meantime, another peer node will become in range of communication, and consequently, is able to receive the message. The PKG, however, does not rebroadcast its replies, because we assume its transmission range covers the whole DTN zone. Thus, the reply delays come basically from processing the message queue.

4.2 Dual DTN Zone

The second simulation is intended as a proof-of-concept of the proposed communication model. It consists of two disconnected DTN zones, with a PKG and three nodes each, in addition to a Mobile Gateway node that moves from Zone1 to Zone2. The simulation parameters used for this test case are the following: Communication model is SimpleCommModel, Link layer model is SimpleLinkModel, Mobility model is UniformSpeedModel for normal nodes, FixedMobilityModel for PKG, Trip is SightseeingModel, Simulation duration is 3 minutes, Reception range of normal nodes is 200m, Reception range of the PKG is 100m, Maximum travel distance is 100m in Zone1, 50m in Zone2 and 1000m for the gateway node, SNR/RSSI is 65/-30, Uniform speed is 5-10mph, and Status Update Duration is 1 minute.

The simulation results are shown in Table 3.

Table 3. Simulation Results for Two DTN Zones.

Performance Parameter	Zone 1	Zone2
PRR	55.5%	55.5%
PRD (in s)	0.0117	0.0317
T_{Rx} (in s)	0.776	0.71
δT_{Rb} (in s)	0.35	0.588

4.3 Parameters Affecting DTN Security

Although the simulation described in this section does not take into consideration the security aspect of the DTN model, we can anticipate the effect of some parameters on the overall security of the communications.

- Key validity period ΔT: Every ΔT, the PKG identities are changed, and the nodes need to request fresh private keys to be able to exchange messages within this period. Thus, as ΔT becomes shorter, private keys are refreshed more frequently, and stored data is changing dynamically. Moreover, since the PKG is being replaced more often, it becomes less vulnerable to compromise, and the risk of it being discovered decreases. However, a shorter ΔT means that nodes will have to connect to the PKG more frequently, which reduces the efficiency of the DTN model.
- Status Update (Heartbeat) Period ΔT_1: Nodes send their status updates to the PKG every ΔT_1. This is important if they need to send messages to nodes outside the PKG zone, since the latter has to decide which Mobile Gateway is going to carry these messages to the next zone. For local communications, the status update period does not carry a great significance. However, if the major traffic is going outside the DTN zone, then a shorter ΔT_1 means a faster update and a more dynamic data exchange between zones. Again, this requires that the PKG actually receives these update messages, which means

that nodes need to be in contact with the PKG more often.

5. Conclusion

In this paper, we presented a security solution that uses DTNs to protect communications in dangerous environment and territories under the control of oppressive governments. We started by a brief overview of DTN applications and architecture, and then we investigated the challenges and concerns faced when trying to secure such networks.

A lot of research has already been done to enable secure communications within DTNs, and many algorithms and protocols have been devised to ensure the privacy and anonymity of these communications. Existing security mechanisms are reviewed, and some of them are utilized to develop the security solution of the proposed system.

The DTN model proposed considers a group of people who want to communicate secretly, without raising the authorities' suspicions, for fear of death or captivity. They want to exchange critical information while keeping their identities hidden, and the content of the exchanged messages confidential. This model is presented in details, and the communication scenario is explained with the proposed security-providing mechanisms.

We simulated the DTN model using GrooveNet, a simulator originally intended for vehicular communications. We presented the simulation setup and the different test cases used to study the behavior of this system. We then analyzed the effect of the different parameters on the security and overall performance of the network.

6. References

[1] I. Psaras, L. Wood and R. Tafazolli, "Delay-/disruption tolerant networking state of the art and future challenges," submitted to Elsevier, November 2009.

[2] V. Venkataraman, H. B. Acharya, H. Shah and S. Lam, "Delay tolerant networking – a tutorial," 2009.

[3] A. Kate, G. Zaverucha, and U. Hengartnerl, "Anonymity and security in delay tolerant networks," in *SecureComm 2007*. IEEE, September 2007.

[4] A. Seth, U. Hengartner, and S. Keshav, "Practical security for disconnected nodes," in *First Workshop on Secure Network Protocols (NPSec)*, November 2005.

[5] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott and C. Luo, "Towards securing disruption-tolerant networking," Nokia Research Center, 2007.

[6] S. Farrell and V. Cahill, "Security considerations in space and delay tolerant networks," in *Proceedings of the 2nd IEEE Conference on Space Mission Challenges for Information Technology*, 2006.

[7] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Communications of the ACM*, *42(2)*, February 1999.

[8] IRTF. RFC 4838 - DTN Architecture, 2007. http://www.ietf.org/rfc/rfc4838.txt

[9] IRTF. RFC 5325 - Licklider Transmission Protocol, 2008. http://www.ietf.org/rfc/rfc5325.txt

[10] S. Symington, S. Farrell, and H. Weiss. Bundle Security Protocol Specification. IRTF, DTN research group, October 2006. Draft version -02; expired in April 2007.

[11] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *Proceedings of Symposium on Cryptography and Information Security (SCIS 2000)*, 2000.

[12] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of CRYPTO 2001*, 2001.

[13] K. Srinivasan and P. Ramanathan, "Reliable anonymous multicasting in disruption tolerant networks," in *Proceedings of IEEE GLOBECOM*, 2008.

[14] A. Lindgren and P. Hui, "The quest for a killer app for opportunistic and delay tolerant networks," 2009.

[15] W. Itani, A. Tajeddine, A. Kayssi, and A.Chehab, "Slow but certain wins the race: authenticated bundle communication in delay tolerant networks," *in proc. Of Q2SWinet'10*, Bodrum, Turkey, October 2010.

[16] Mack, Jefferson. Invisible Resistance to Tyranny: How to Lead a Secret Life of Insurgency in an Increasingly Unfree World. Paladin Press, 2002.

[17] C. Sonn and A. Fisher, "Sense of community: Community resilient responses to oppression and change," 1998.

[18] Umoja, Akinyele Omowale(1999) 'Repression breeds resistance: The black liberation army and the radical legacy of the black panther party', New Political Science, 21: 2, 131 — 155.

[19] New Wi-Fi Direct Gets Peer-to-Peer Connections. Business Center. October 14, 2009.

[20] M. Rhevathi, "Report on identity-based cryptography and its applicability in disruption-tolerant networking," 2007.

[21] A. Shamir, "Identity-based cryptosystems and signature schemes," In G. Blakley and D. Chaum,

editors, *Advances in Cryptology – CRYPTO '84*. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1985.

[22] X. Ding and G. Tsudik, "Simple identity-based cryptography with mediated RSA," in *Proceedings of the 2003 RSA conference on The cryptographers' track*, 2003.

[23] P. Mental. (1998, Fall). Cushioned Encryption and Deniability. 2600 Magazine. pp. 20-22.

[24] N. L. Karstens, "Deniable Encryption," 2006. http://www.iac.iastate.edu/iasg/libarchive/0607/20070213_nate_presentation.pdf

[25] TrueCrypt. (2010). TrueCrypt User Guide. TrueCrypt Foundation. [Online]. Available: http://www.truecrypt.org/user-guide/.

[26] GrooveNet. (2010). [Online]. Available: http://www.seas.upenn.edu/~rahulm/Research/GrooveNet.

