Opportunistic key management in delay tolerant networks

Sofia Anna Menesidou*

Information Security and Incident Response Unit,
Department of Electrical and Computer Engineering,
Democritus University of Thrace,
University Campus, Xanthi 67100, Greece
Email: smenesid@ee.duth.gr
*Corresponding author

Vasilios Katos

Department of Computing and Informatics,
Bournemouth University,
Poole House, Fern Barrow, BH12 5BB, UK
Email: vkatos@bournemouth.ac.uk

Abstract: Key management is considered to be a challenging task in delay tolerant networks (DTNs) operating in environments with adverse communication conditions such as space, due to the practical limitations and constraints prohibiting effective closed-loop communications. In this paper, we propose opportunistic key management as a more suitable solution for key management in networks requiring opportunistic behaviour. We show that opportunistic key management is better exploited and utilised when used in conjunction with routing decisions by security aware DTN nodes.

Keywords: opportunistic key management; OKM; security-aware routing decision; delay tolerant networks; DTNs.

Reference to this paper should be made as follows: Menesidou, S.A. and Katos, V. (xxxx) 'Opportunistic key management in delay tolerant networks', *Int. J. Information and Computer Security*, Vol. X, No. Y, pp.xxx–xxx.

Biographical notes: Sofia Anna Menesidou holds a Diploma in Information and Communication Systems Engineering in 2008 and MSc in Information and Communication Systems Security in 2010, both from the University of the Aegean, Greece. Currently, she is a PhD candidate at the Department of Electrical and Computer Engineering at Democritus University of Thrace, Greece.

Vasilios Katos is a Professor and the Head of Computing at the Bournemouth University. Prior to his current post, he was a Professor of Information and Communications Systems Security at the Department of Electrical and Computer Engineering of Democritus University of Thrace in Greece, and Principal Lecturer at the School of Computing at the University of Portsmouth where he participated in the development of the interdisciplinary Masters course MSc in Forensic IT. He has worked in the industry as a Security

Consultant and expert witness in information systems security. His research interests are in information security, privacy, digital forensics and incident response.

1 Introduction

Delay tolerant networks (DTNs) are becoming popular both in terrestrial and deep space environments as they maintain certain advantages over traditional internetworking protocols such as TCP/IP. DTN is an overlay on top of a number of heterogeneous networks including the internet (Ahmad et al., 2010). Differing from traditional networks, DTNs are distinguishably characterised by:

a mobility and dynamic topology

b high transmission delays

c energy consumption

d limited bandwidth

e high bit-error rate

f uncontrolled size of nodes.

The constraints under which such networks function have severe effects making the adoption of a large number of security protocols and traditional solutions impossible. Moreover, in DTNs, the messages are disseminated according to the store-carry-and-forward principle. Although DTNs by their nature support high availability, they are not short of security-related challenges. Cryptographic key management is considered to be a challenging task in such networks and specifically in deep space communications because many mature and secure protocols are not practical in such environments. Naturally, poor or weak cryptographic key management will have an adverse effect on the cryptographic techniques which risk of being rendered insecure or inefficient (Zamani and Zubair, 2014).

In a DTN setting, an end-to-end path between the source and the destination may not exist making routing in DTNs opportunistic. Conventional internet routing protocols are not suitable in a DTN as the end-to-end connectivity requirement is not necessarily satisfied. Despite the broad research on routing protocols in DTNs, very few consider security issues, whereas key management is not treated adequately. By exploiting the opportunity, whenever that arises, to exchange a key for future usage, end-to-end delay can be minimised over time. The purpose of this paper is to propose opportunistic key management (OKM) as a more suitable approach for key management in DTNs. We examine OKM using existing and popular protocols and we show that OKM is better exploited and utilised when used in conjunction with routing decisions by security aware DTN nodes.

The remainder of this paper is structured as follows. Section 2 provides the literature review and the current state of the art of key management and routing considering security aspects in DTNs. In Section 3, OKM is discussed as a more flexible approach for DTNs. This approach is based on the well-known scheduling and critical path methods

(CPMs). In Sections 4 and 5, we present the protocol parser we developed for the evaluation of our proposed method and the simulation results from a number of OKM scenarios. Finally, we conclude the paper with Section 6, where open issues and future word are summarised.

2 Related work

As mentioned earlier, cryptographic key management is considered to be a challenging problem in DTN environments. The difficulties and challenges are due to the peculiarities and constraints of the harsh communication conditions DTNs typically operate in, rather than the actual features of the underlying key management cryptographic protocols and solutions. The constraints of DTN environment make many good quality key management protocols described in literature unsuitable. Up to date, the literature has a relatively long domain of routing in DTNs but very few consider the security parameter. Key management and secure routing are important issues in DTNs, but solutions proposed to date tend to consider them separately. From a practical perspective such paradigm is limiting because security-based routing may be impractical since, due to the constraints of the DTN environment, a node may not possess the necessary cryptographic material needed to perform a cryptographic operation and complete a protocol run required for the underlying secure routing operation. The literature review is two-fold. The first part is about key management in DTNs and the second part is about security-aware routing in such networks.

2.1 Key management in DTNs

The literature has a plethora of fairly mature cryptographic key management protocols. A significant amount of work is published, were key management protocols are available most of which have been extensively analysed, with well-recognised security properties and acknowledged weaknesses. As such, the prohibiting factors relate to the practical communication, performance and efficiency aspects rather than the security capabilities of the key management protocols.

Symington et al. (2014) and Birrane (2013) have explicitly excluded key management as it is a challenging topic. In Farrell et al. (2009), internet draft the author states a series of requirements for key management in DTNs too, without proposing a solution. The recent internet draft (Templin, 2015) proposes requirements and outlines a design for secure key management in DTNs. In addition, Templin (2014) states the key management problem in DTNs and emphasises that traditional security key management mechanisms are not always feasible in environments DTN typically operate in.

Shikfa et al. (2012) propose a specific key management scheme for content-based opportunistic networks that enables the bootstrapping of local, topology-dependent security associations between a node and its neighbour along with the discovery of the neighbourhood topology. They achieve that by using pseudonym certificates and encapsulated signatures. In Menesidou and Katos (2012), one-pass key establishment protocol is used for authenticated key exchange in DTNs. More specifically, an adoption of Horsters-Michels-Petersen Protocol (HMP) is used for key establishment, while the protocol messages are injected in the bundle payload. In another work, Ding et al. (2013)

present an authentication and key agreement protocol with anonymity based on combined public key. Their protocol eliminates the need of public key digital certificate online retrieval and only an offline repository is needed. Djamaludin et al. (2013) build and compare different decentralised trust systems for implementation in autonomous DTN systems. They utilise a key distribution model based on the web of trust principle and compare it with two other decentralised methods.

More recently, Bhutta et al. (2014) present an efficient and scalable key transport scheme (ESKTS) based on public key cryptography and proxy signatures. This scheme ensures that integrity and authentication is achieved at hop-by-hop level as well as end-to-end level. It also ensures end-to-end confidentiality and freshness for end communicating parties. Zhou et al. (2014) propose an autonomic group key management scheme for DTNs. Specifically, they present a logical key tree based on one-encryption-key multi decryption-key key protocol for group key management. In terms of efficiency, their scheme costs less than other proposed, making it more suitable for deep space DTNs. Moreover, Salem et al. (2014) propose a dynamic key distribution protocol for PKI-based VANETs. They also propose a key revocation mechanism that reduces the number of messages needed for revocation through certificate revocation list (CRL) distribution.

2.2 Security-aware routing in DTNs

As already highlighted, the literature contains an extensive study of routing in DTNs but very few consider security aspects. Xiaofeng et al. (2010) present an anti-localisation routing protocol (ALAR), to achieve anonymous delivery in DTNs. The proposed protocol can protect the sender's location privacy through message fragmentation and forwarding each segment to different receivers. Their technique can lower the sender's probability of being localised. However, the authors highlight that they did not consider the energy consumption, the proportion of sensitive messages and the distribution of the sensitive messages in their study. Pushpalakshmi and Kumar (2010), and Pushpalakshmi et al. (2011) present a secure minimised dominating set routing protocol for mobile ad hoc network (MANET) based on DTNs. In their technique, the messages are routed only to trustable nodes in minimised dominating set created by the trust level of the node, and the probability of future contacts. To deal with the effect of a potentially malicious node, trustworthiness of node can be taken into consideration in routing decision making. The work in Bulut and Szymanski (2011) focuses on the problem of routing in compromised DTNsin presence of malicious nodes. They discuss and analyse several message distribution schemes in terms of secure delivery of messages. Chen et al. (2012) present a dynamic trust management protocol for secure routing optimisation in DTN environments in the presence of well-behaved, selfish and malicious nodes. To analyse their protocol, they use stochastic Petri net (SPN) techniques and to validate it they use simulation. Haigang and Lingfei (2013) present the state-of-the-art of routing protocols in delay tolerant mobile networks (DTMNs). They also admit that there is a little study on security in DTMSs and they discuss and accept security-considered routing as an open issue. Moreover, in Ramesh et al. (2013), a routing protocol that provides security in the intermittently connected mobile networks is proposed, while the routing performance metrics are not degraded with the implementation of the security mechanism. Their routing mechanism aids in detecting and preventing intrusion of malicious nodes. Vrinda and Arun (2014) have implemented a trust and reputation management mechanism. Their

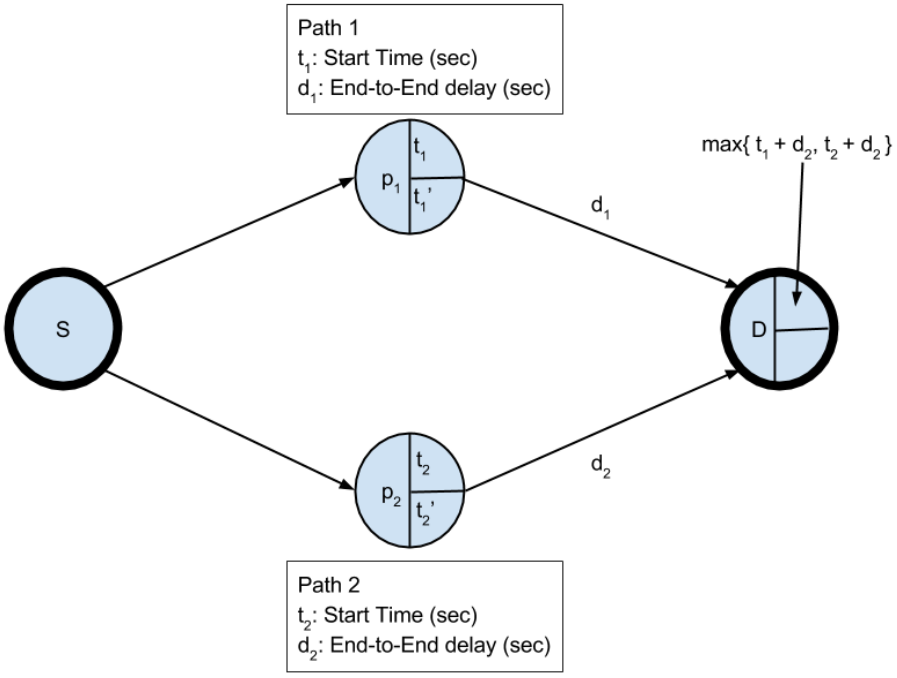
mechanisms serve to ensure secure communication and isolates malicious nodes which may attack the network and reduces network performance in terms of reliability and security. Last but not least, Jadhav et al. (2015) utilise a time-evolving topology model and two-channel cryptography to design an efficient and non-interactive key exchange protocol. The time-evolving model is used to model predetermined behaviour of space DTNs. The two-channel cryptography enables DTN nodes to exchange their public keys and status information, whilst offering authentication and in a non-interactive fashion.

3 Security aware routing and OKM

3.1 Security aware routing – path selection

In this paper, we argue that key management is better exploited and utilised when used in conjunction with routing decisions by security aware DTN nodes. We adopt the CPM as a decision making method for different paths between source and destination. CPM is used for activities scheduling in project management. The duration for each activity is the end-to-end delay. The critical path represents the most costly, in terms of delay, path. Security aware nodes can exploit the slack time, between the critical path and the selected path, for security and key management tasks.

Figure 1 Security aware routing (see online version for colours)



The following simple scenario illustrates the approach. We assume two different paths (p_1 and p_2) with stable bandwidth (b_1 and b_2) from source to destination. Connection with

path 1 starts at t_1 and with path 2 at t_2. The transmission delay is the volume of data to be sent divided by the bandwidth. Assuming further that queuing delay is negligible, total end-to-end delay (d_1 and d_2, respectively) can be expressed as the propagation delay plus the transmission and the processing delay. Cryptographic delay (that is, delay due to the necessary cryptographic operations required by the underlying cryptographic protocol) is included in the processing delay.

$$End\text{-}to\text{-}end\ delay = Propagation\ delay + Transmission\ delay + Processing\ delay$$

The maximum value between the sum of start time and end-to-end delay of the two paths represents the critical path. Figure 1 represents the aforementioned scenario.

The following equation represents the more efficient path selection based on the critical path.

$$P_x = \begin{cases} x = 1 & \text{if } t_2 + d_2 > t_1 + d_1 \\ x = 2 & \text{if } t_1 + d_1 > t_2 + d_2 \\ x = 1 \text{ or } 2 & \text{if } t_1 + d_1 = t_2 + d_2 \end{cases}$$

The slack time is calculated by the difference of the two end-to-end delays, in case we select the most efficient in terms of delay path. Slack time can be utilised to run security tasks.

$$SlackTime = |t_1 + d_1 - t_2 + d_2|$$

3.2 OKM – security task selection

Traditional security key management mechanisms are not always feasible due to operational limitations of huge delays. Bhutta et al. (2009) mention that a single key management scheme might not be sufficient for DTN networks due to heterogeneity of networks, like satellite networks, sensor networks and so forth. Consequently, a more complete framework would be required to provide key management services. In many cases, this can be addressed by a public key infrastructure (PKI) that has a number of geographically distributed certificate authorities (CA) in order to ensure that private key holders maintain valid keys and are properly authorised. In DTNs, a direct path between the source and destination does not always exist. Such a constraint makes routing in DTNs opportunistic. Based on the same idea, we propose OKM.

Key management-related processes can be executed whenever an opportunity arises which could be due to an idle node or slack time. In many environments requiring DTN solutions, the DTN nodes have plenty of opportunities to stay idle for a considerable amount of time. Nodes can exploit the unused bandwidth by updating their information from CA, run key exchange protocols for future use or perform necessary operations for key management. A security-aware node will check the existence of security tasks, prioritise them and finally schedule the most critical task. A security-aware node can estimate the number of needed session keys based on the connectivity, RTT, established keys, available protocols and usage rate. It is self-evident that communication with already established session key has less end-to-end delay compared to nodes that have first to establish a new one.

Furthermore, a good practice is to run more than one key exchange protocol simultaneously with opportunistic batch key confirmation; the more passes a protocol

has, the more end-to-end delay it adds. There is a plethora of key exchange protocols (e.g., MQV, KEA, etc.) where the last pass is not necessary for the key exchange phase but it is used for key confirmation. For instance, an idle node has time to exchange ten session keys using the first pass of MQV protocol and the connection is lost before the second passes of key confirmation. The rest of the messages can be accumulated and sent later in time when there is new connection and opportunity, while the session keys are already in use. By using opportunistically batch key confirmation, the extra added cost of delay is minimised.

In addition, every security-aware node must distinguish and prioritise its forthcoming needs and to select the most urgent and critical one when the opportunity arises. The keys may have lifetimes associated with them, requiring refresh or replacement upon expiry. These keys and policies must be managed and there must be a mechanism to distribute, refresh, and revoke the keys and policies (Ivancic, 2010). For instance, CAs issue a signed data structure called a CRL. Every node must be able to acquire the new issued CRL. Based on the priorities of the node's security policy, the current needs and the remaining time before the operation expires, the node must be able to select the most suitable operation. The current needs of a node depend on the current status of a node. A typical security aware node task list is as follows:

- *Certificate renewal* – The node can start the certificate renewal phase by creating and replacing the old certificate with a new one or by updating the same certificate. Typically, a certificate expires after one or more years. In addition, a good practice is the renewed certificate be issued and published at least 1 week prior to the expiration of the certificate it replaces.
- *CRL update* – A new CRL is issued on a regular periodic basis. For instance, due to the huge delays in DTNs, a weekly update could be a good approach. This task can also be combined, if there is available bandwidth, with another necessary information from CA (e.g., missing certificates retrieval).
- *Missing certificates retrieval* – When there is a missing certificate (e.g., an expired, a revoked or a not acquired certificate) the node can retrieve it from the CA. More specifically, in deep space DTNs, the connectivity to a neighbour node would be known if a contact graph routing (CGR) is present. In addition, each node can keep a list with the neighbour's missing certificates and based on the available bandwidth and the connectivity the node can retrieve them from the CA. This task, when it is feasible, can also be combined with the CRL Update.
- Establish session keys – Every node can maintain a list of a number of established session keys and the usage rate per neighbour node. Based on this information, the node will be able to establish more session keys before all the keys are used. Moreover, the accumulated key confirmation passes of previous runs of key exchange protocols can also be scheduled.

In the proposed framework, a security-aware node will be able to prioritise and analyse these security tasks to find the most critical one and this will be done by employing CPM practices. As such, the node will be able determine which tasks are critical and which can be delayed without managing thus end-to-end delays.

3.2.1 CPM in OKM exemplar

Consider a setup of four nodes, namely A,B,C,D and a separate node running a CA. Node A is currently idle, has three neighbours (B, C, D) and a CA neighbour. In addition, based on the CGR the next connection with CA starts in 30 minutes and with nodes B, C and D starts in 0 minutes, 15 minutes and 1 hour respectively. The RTT delay ranges between 15 minutes to 1 hour. The session key usage rate between node A and the rest of the nodes B, C and D are 0.5, 0.05 and 0.05 keys/sec, respectively. We also assume that for session key establishment one pass key exchange protocol is used. Table 1 summarises all the aforementioned contact information for node A.

In addition, the current status of node A is:

- node's A certificate expires in 7 days and minutes
- CRL expires in 2 hours
- 2 certificates are stored locally
  1 node's B certificate expires in 2 hours 15 minutes
  2 node's C certificate expires in 3 hours
  3 node's D certificate is missing
- 900 session keys are stored locally
  1 450 session keys are established with node B
  2 450 session keys are established with node C
  3 0 session keys are established with node D.

Table 1 Node's A contact information

Nodes	*Connection start*	*RTT*	*sKey usage rate*
CA	30 min	15 min	-
B	0 min	30 min	0.5 keys/sec
C	15 min	60 min	0.05 keys/sec
D	60 min	30 min	0.05 keys/sec

The question is which operation is more suitable to take place first. The security tasks (activities) node A must perform are:

1 Renew certificate from CA – The activity can start when the connection with CA starts (30 minutes). The duration is the same with the RTT (15 minutes). The activity must finish 7 days before certificate expiration (45 minutes).

2 Update CRL from CA – The activity can start when the connection with CA starts (30 minutes). The duration is the same with the RRT (15 minutes). The activity must finish before CRL expiration (120 minutes).

3 AKE with node B (old certificate) – The activity can start when the connection with node B starts (0 minutes). The duration is half the RTT because we have one-pass key exchange protocol (15 minutes). The activity must finish before established session keys are finished (15 minutes) and before expiration of the certificate (135 minutes).

4 Retrieve node's B certificate from CA – The activity can start when the connection with CA starts (30 minutes). The duration is the same with the RTT (15 minutes). The activity must finish before expiration of the new certificate (1 year). Activity 4 must follow activity 3.

5 AKE with node B (new certificate) – The activity can start when the connection with node B starts (0 minutes) and activity 4 is finished. The duration is half the RTT because we have one-pass key exchange protocol (15 minutes). The activity must finish before expiration of the new certificate (1 year). Activity 5 must follow activities 1 and 4.

6 AKE with node C (old certificate) – The activity can start when the connection with node C starts (15 minutes). The duration is half the RTT because we have one-pass key exchange protocol (30 minutes). The activity must finish before established session keys are finished (150 minutes) and before expiration of the certificate (180 minutes).

7 Retrieve node's C certificate from CA – The activity can start when the connection with CA starts (30 minutes) and activity 6 is finished. The duration is the same with the RTT (15 minutes). The activity must finish before expiration of the new certificate (1 year). Activity 7 must follow activity 6.

8 AKE with node C (new certificate) – The activity can start when the connection with node C starts (15 minutes) and activity 7 is finished. The duration is half the RTT because we have one-pass key exchange protocol (30 minutes). The activity must finish before expiration of the new certificate (1 year). Activity 8 must follow activities 1 and 7.

9 Retrieve node's D certificate from CA – The activity can start when the connection with CA starts (30 minutes). The duration is the same with the RTT (15 minutes). The activity must finish before expiration of the new certificate (1 year).

10 AKE with node D – The activity can start when the connection with node D starts (60 minutes) and activity 9 is finished. The duration is half the RTT because we have one-pass key exchange protocol (15 minutes). Activity 10 must follow activity 9.

Table 2 Node's A security activities

Activities of node A		*Start*	*End*	*Duration*
1	Renew certificate from CA	0:30	0:45	15 min
2	Update CRL from CA	0:30	2:00	15 min
3	AKE with node B (old certificate)	0:00	0:15	15 min
4	Retrieve node's B certificate from CA	0:30	1 year	15 min
5	AKE with node B (new certificate)	After activity 4	1 year	15 min
6	AKE with node C (old certificate)	0:15	2:30	30 min
7	Retrieve node's C certificate from CA	After activity 6	1 year	15 min
8	AKE with node C (new certificate)	After activity 7	1 year	30 min
9	Retrieve node's D certificate from CA	0:30	1 year	15 min
10	AKE with node D	After Activity 9	1 year	15 min

Figure 2 Activity network and critical path (see online version for colours)

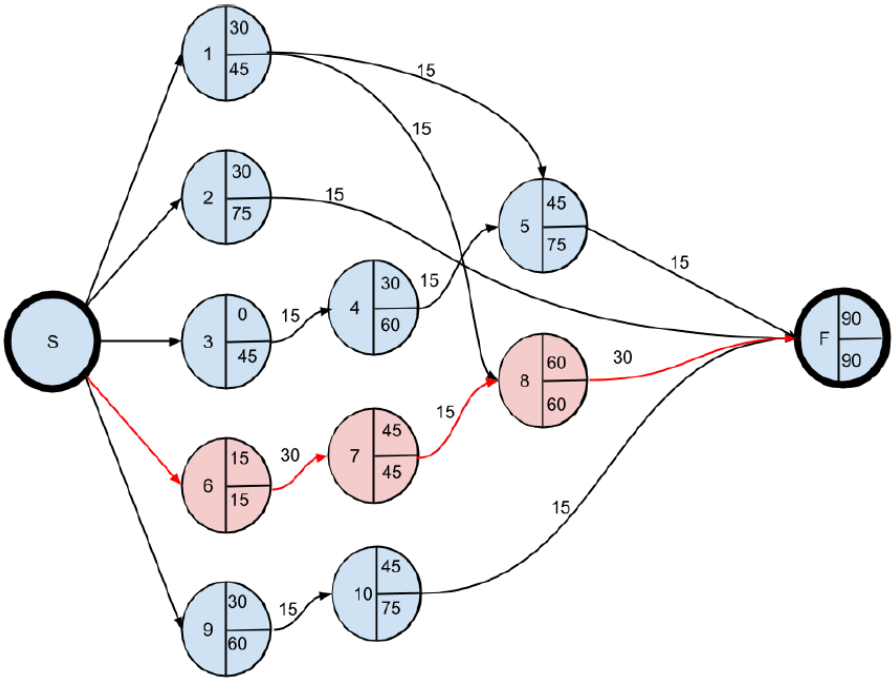


Figure 3 Gantt chart (see online version for colours)

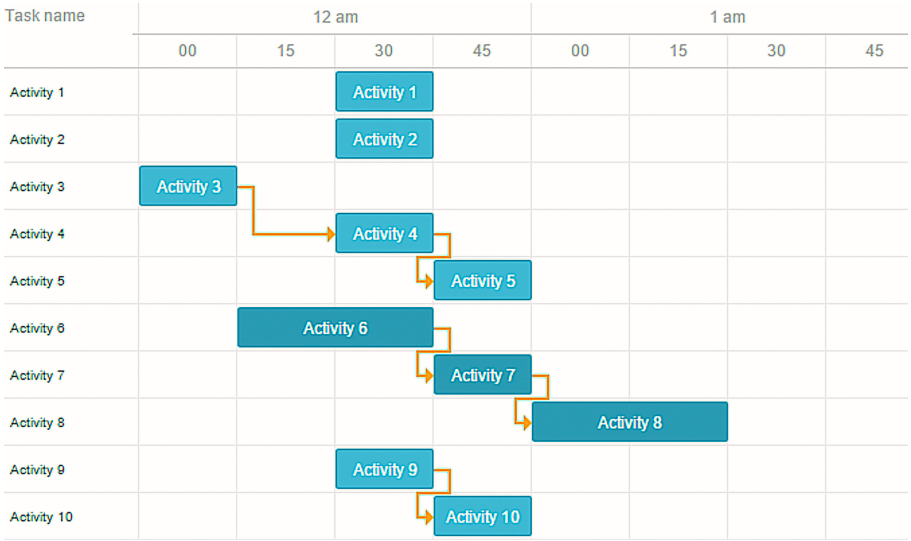


Table 2 summarises all the security activities of node A together with the start time, end time and duration. The start time and the duration are based on the RTT, the session key usage rate and the connectivity. The first step is to create the activity network diagram (see Figure 2) to see the activity dependencies and to identify the critical path. The critical path is depicted with red line.

The next step is to create, the Gantt chart (see Figure 3) which could be proved helpful to see an overall view of all the security activities. Now, node A has a complete idea which task is more urgent to do first. The critical path is presented too, with dotted lines.

4 Protocol parser

A protocol parser was developed to support the evaluation of more complex scenarios of security-aware routing decisions (see Figure 4) and to evaluate the proposed concept of OKM. The main goal of the parser is to automate the comparison of various protocols and scenarios, although the underlying functionality can be included in a DTN node to add advanced decision making capabilities on routing. The parser will recommend injection of additional protocols in the case of missing security parameters. More specifically, message length, number of passes, network topology, access to certificates and keys are analysed in order to produce a ns-2 script for simulation. In addition, the protocol language has been enriched with a cryptographic delay per byte of information parameter, which will be passed on to the DTN agent (Vardalis, 2014). The parser supports import/export in xml format to facilitate integration with other tools and to allow an effective design of scenarios. The supported xml format is depicted in Figure 5 where the introduced xml tags – namely routing, nodes, name and messages – define the corresponding areas in the parser. In this example, we can see the scenario of the two-pass MQV key agreement protocol between nodes A and B. From nodes, we can see the bandwidth (downlink/uplink), the node's credentials (e.g., access to private and public keys) and the message between the nodes.

Table 3 AKA protocol comparison

	HMP	*MQV*	*STS*
Fundamental security goals			
Implicit key authentication	Y	Y	Y
Explicit key authentication	N	N	Y
Desirable security attributes			
Known-key security	N	N	Y
Forward secrecy	N	N	Y
Key-compromise impersonation	-	N (Chalkias et al., 2011)	Y
Unknown key-share	-	N+ (Chalkias et al., 2011)	N
Desirable performance attributes			
Minimal number of passes	1	2	3

Source: Menesidou and Katos (2012)

Figure 4 Protocol parser (see online version for colours)

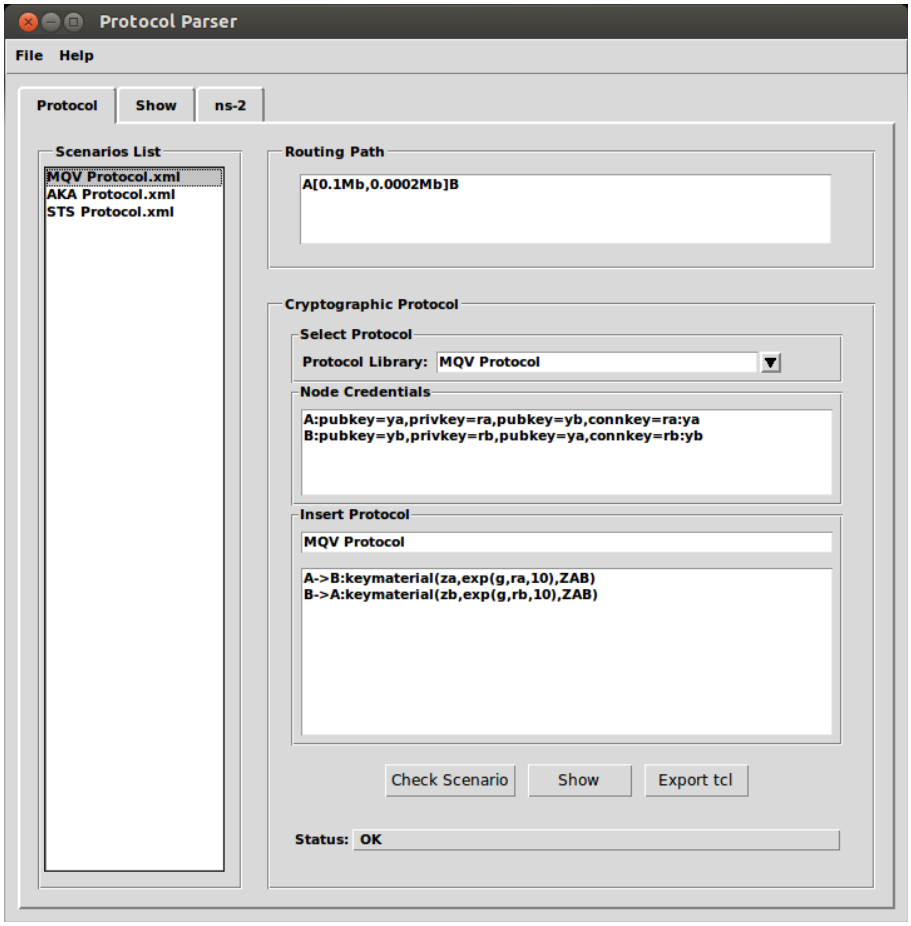


Figure 5 MQV Scenario in XML format (see online version for colours)

```
<protocol>
    <routing>A[0.1Mb,0.0002Mb]B
    </routing>
    <nodes>A:pubkey=ya,privkey=ra,pubkey=yb,connkey=ra:ya
    B:pubkey=yb,privkey=rb,pubkey=ya,connkey=rb:yb
    </nodes>
    <name>MQV Protocol
    </name>
    <messages>A-&gt;B:keymaterial(za,exp(g,ra,10),ZAB)
    B-&gt;A:keymaterial(zb,exp(g,rb,10),ZAB)
    </messages>
</protocol>
```

5 Evaluation

A number of scenarios, described in the following subsections, were created to assess the added value of security-aware routing, the efficiency of OKM and specifically of pre-established necessary keys. Initially the scenarios were created and fed through the protocol parser. The latter produced outputs in the form of tcl scripts that were in turn fed to ns-2 and specifically to the DTN agent for ns-2 (Vardalis, 2014) that was used for the simulation. The performance metric is the end-to-end delay which is defined as the time taken for a bundle to be transmitted across a network from source to destination. It involves propagation delay, transmission delay, and cryptographic delay (encryption/decryption). For the simulations TCP was used as the convergence layer.

5.1 Number of established session keys

In this scenario, a custodian node A (source) wants to forward data to node B (destination). While node B is waiting for the data from node A, OKM is used. More specific, multiple key exchange protocols are established between idle nodes B and C for future use. The bandwidth used for all the available paths are 0.1 Mbps and the propagation delay is 10 minutes, which is a typical value for space DTNs. We calculated how many session keys can be established for different volume of data and for three different key exchange protocols. The authenticated key agreement (AKA) protocols we selected for comparison are the one-pass HMP, the MQV and the STS protocol. Table 3, adopted from Menesidou and Katos (2012) present the three selected protocols against the definitions and requirements for authenticated key establishment from Blake-Wilson and Menezes (1999).

Figure 6 Number of established session keys (see online version for colours)

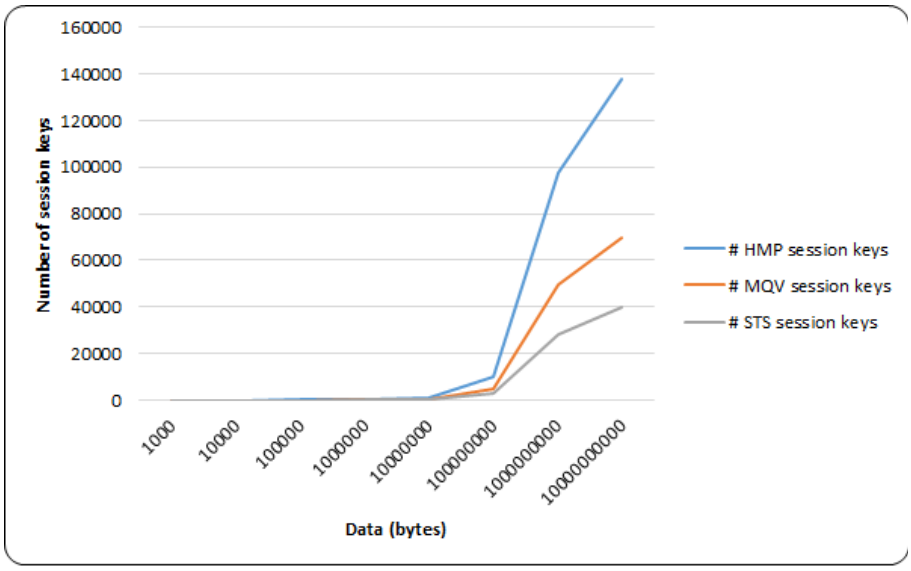
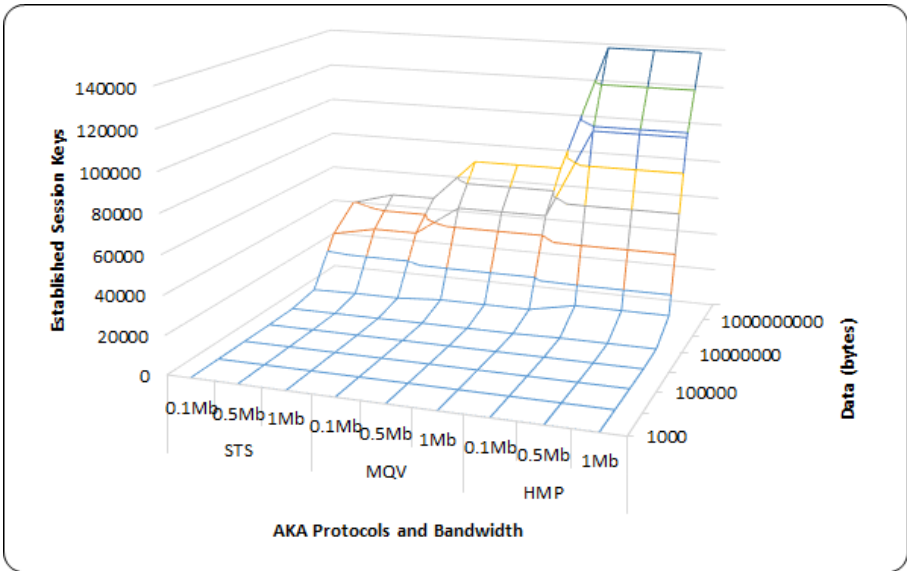


Figure 6 presents the number of possible exchanged session keys, while node B is waiting for the data from node A. However, the number of stored session keys depends on the available storage, the remaining energy of each node and the usage rate. From Table 3 and Figure 6, the trade-off between efficiency and security is evident as we can establish that STS is more secure but HMP is more efficient.

Figure 7 Number of established session keys (3D) (see online version for colours)

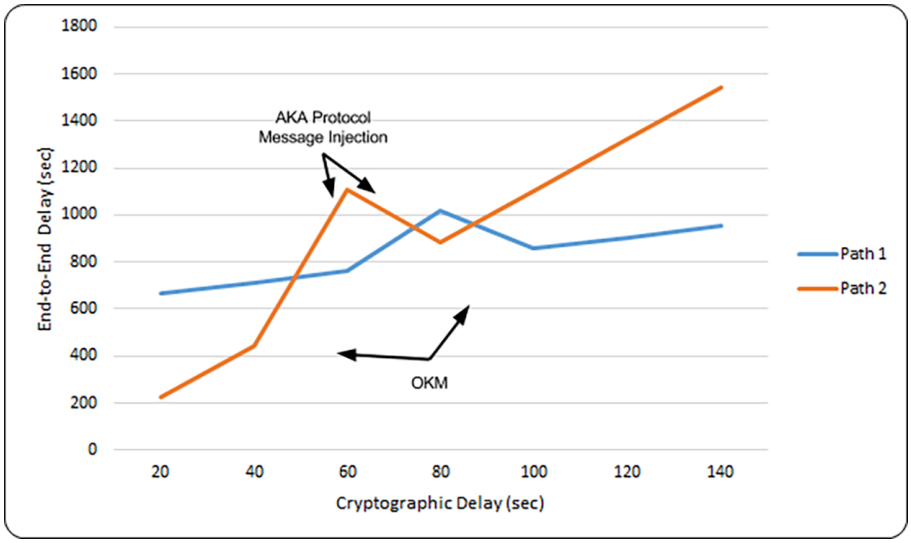


By counting the huge delays in deep space communications, DTN nodes will be idle most of the time and consequently the established session keys will be enough to minimise the imposed delay of the key exchange phase. The idle node will be able to predict which key exchange protocol is better for each situation based on the usage rate, the security policy and the protocol suit. The same scenario has been extended for two more different bandwidth values 0.5 Mb and 1 Mb. The surface graph in Figure 6 highlights the efficiency of HMP protocol as the waiting phase of the idle node and the bandwidth are increasing.

Table 4 Security aware routing simulation scenario parameters

	Path 1 (A-B-D)	*Path 2 (A-C-D)*
Start time (sec)	600 sec	0 sec
Data to send	10–70 MB	10–70 MB
Bandwidth	10 Mbps	1 Mbps
Propagation delay	0.1 sec	0.1 sec
Number of established keys	A–B 90 keys	A–C 80 keys
	B–D many keys	C–D many keys
Usage rate	10 keys/sec	10 keys/sec
Cryptographic protocol	STS	HMP

Figure 8 Security aware routing decision (message injection vs. OKM) (see online version for colours)



5.2 Security-aware routing decision

In this scenario, we compare two different paths between node A (source) and node D (destination). Specifically, we assume path 1: A-B-D and path 2: A-C-D with 10 Mbps and 1 Mbps bandwidth, respectively. In both paths confidentiality is a security requirement. Path 1 is available after 10 minutes and path 2 is available now. Moreover, the propagation delay is 100 ms. All node pairs have already established session keys by a previous run of OKM and the 128-bit AES algorithm will be used for encryption. Between nodes B, D and C, D many session keys established with OKM. Between nodes A, B and A, C, 90 and 80 session keys are established with 10 keys/sec usage rate. In addition, node A supports STS and HMP protocols for key exchange, node B supports only STS protocol and node C supports only HMP. Table 4 summarises the simulation parameters and in Figure 8 we can see the trade-off between end-to-end delay and cryptographic delay before and after AKA protocol message injection for new session keys establishment. The injection phase starts when the previously established session keys are finished. As expected, end-to-end delay is minimised when is used OKM.

6 Conclusions and future work

In this paper, we have attempted to address the challenging task of cryptographic key management in opportunistic networks by incorporating security awareness in routing. The idea of OKM in DTNs is introduced. It was shown that a systematic prioritisation of security tasks can help to effectively control end-to-end delays. For our simulations, the DTN-agent for ns-2 is used to run various scenarios. We also identified that OKM can result in more efficient security aware routing.

Ongoing future efforts include the further development of the parser to enable the implementation of more complex scenarios of various opportunistic networks, other than space DTNs, as well as the inclusion of more parameters that influence the prioritisation and execution of security activities. These parameters include the available storage, the remaining energy of the node and the usage rate, and will affect, among others, the number of protocol runs and session keys stored by the node. Another plan for future work is to identify which key establishment protocol will provide the better efficiency vs. security trade-off. Finally, an area of ongoing research is the integration of the proposed OKM with an existing testbed in order to empirically evaluate it.

References

Ahmad, N., Cruickshank, H. and Sun, Z. (2010) 'ID based cryptography and anonymity in delay/disruption tolerant networks', *Personal Satellite Services, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. 43, pp.265–275.

Bhutta, N., Ansa, G., Johnson, E., Ahmad, N., Alsiyabi, M. and Cruickshank, H. (2009) 'Security analysis for delay/disruption tolerant satellite and sensor networks', *International Workshop on Satellite and Space Communications, 2009, IWSSC 2009*, pp.385–389.

Bhutta, N., Cruickshank, H.S. and Sun, Z. (2014) 'An efficient, scalable key transport scheme (ESKTS) for delay/disruption tolerant networks', *Wireless Networks*, Vol. 20, No. 6, pp.1597–1609.

Birrane, E. (2013) *Streamlined Bundle Security Protocol Specification*, internet-draft [online] http://tools.ietf.org/html/draft-irtf-dtnrg-sbsp-01 (accessed 6 July 2016).

Blake-Wilson, S. and Menezes, A. (1999) 'Authenticated Diffie-Hellman key agreement protocols', *Proceedings of the Selected Areas in Cryptography*, pp.339–361.

Bulut, E. and Szymanski, B.K. (2011) 'On secure multi-copy based routing in compromised delay tolerant networks', *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp.1–7.

Chalkias, K., Halkidis, S.T., Hristu-Varsakelis, D., Stephanides, G. and Alexiadis, A. (2011) 'A provably secure one-pass two-party key establishment protocol', *3rd International SKLOIS Conference on Information Security and Cryptology – Inscrypt*, pp.108–122.

Chen, I., Bao, F., Chang, M. and Cho, J. (2012) 'Dynamic trust management for delay tolerant networks and its application to secure routing', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 5, pp.1200–1210.

Ding, Y., Zhou, X., Cheng, Z. and Zeng, W. (2013) 'Efficient authentication and key agreement protocol with anonymity for delay tolerant networks', *Wireless Personal Communications*, Vol. 70, No. 4, pp.1473–1485.

Djamaludin, C.I., Foo, E. and Corke, P. (2013) 'Establishing initial trust in autonomous delay tolerant networks without centralised PKI', *Computers & Security*, Vol. 39, Part B, pp.299–314.

Farrell, A., Symington, S.F., Weiss, H. and Lovell, P. (2009) *Delay-Tolerant Networking Security Overview*, internet-draft [online] http://tools.ietf.org/html/draft-irtf-dtnrg-sec-overview-06 (accessed 6 July 2016).

Haigang, G. and Lingfei, Y. (2013) 'Study on routing protocols for delay tolerant mobile networks', *International Journal of Distributed Sensor Networks*, Vol. 2013, No. 4, pp.1–16.

Ivancic, W.D (2010) 'Security analysis of DTN architecture and bundle protocol specification for space-based networks', *Aerospace Conference, 2010 IEEE*, pp.6–13.

Jadhav, C.S., Dhainje, P.B. and Pradeep, D. (2015) 'Secure key establishment for bundle security protocol of space DTNs in noninteractive manner', *International Journal of Computer Science and Information Technologies (IJCSIT)*, Vol. 6, No. 2, pp.944–946.

Menesidou, S. and Katos, V. (2012) 'Authenticated key exchange (AKE) in delay tolerant networks', *Information Security and Privacy Research IFIP Advances in Information and Communication Technology*, Vol. 376, pp.49–60.

Pushpalakshmi, R. and Kumar, A.V.A. (2010) 'Security aware minimized dominating set based routing in MANET', *2010 International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp.1–5.

Pushpalakshmi, R., Vincent, A. and Kumar, A. (2011) 'A secure dominating set based routing and key management scheme', *Mobile Ad Hoc Network*, Vol. 10, No. 10, pp.297–307.

Ramesh, S., Indira, R., Praveen, R. and Kumar, G. (2013) 'S-spray routing protocol for intermittently connected mobile networks', *ICTACT Journal on Communication Technology*, Vol. 4, No. 3, pp.761–765.

Salem, A.H., Abdel-Hamid, A. and Abou El-Nasr, M. (2014) 'The case for dynamic key distribution for PKI-based VANETs', *International Journal of Computer Networks & Communications (IJCNC)*, Vol. 6, No. 1, pp.61–78.

Shikfa, A., Önen, M. and Molva, R. (2012) 'Local key management in opportunistic networks', *Int. J. Communication Networks and Distributed Systems*, Vol. 9, Nos. 1/2, pp.97–116.

Symington, S., Farrell, S., Weiss, H. and Lovell, P. (2014) *Bundle Security Protocol Specification*, Request for Comments, RFC 6257 [online] https://tools.ietf.org/html/rfc6257 (accessed 6 July 2016).

Templin, F. (2014) *Delay Tolerant Networking Security Key Management – Problem Statement*, internet-draft [online] http://tools.ietf.org/html/draft-templin-dtnskmps-00 (accessed 6 July 2016).

Templin, F. (2015) *DTN Security Key Management – Requirements and Design*, internet draft, https://tools.ietf.org/html/draft-templin-dtnskmreq-00 (accessed 6 July 2016).

Vardalis, D. (2014) *DTN Agent for ns-2* [online] http://www.spice-center.org/dtn-agent/ (accessed 4 August 2015).

Vrinda, A. and Arun, A.M, (2014) 'Trust and reputation management for secure communication in DTNs', *IJCAT Journal*, Vol. 1, No. 6, pp.245–248.

Xiaofeng, L., Pan, H., Don, T., Juahua, P. and Zhang, X. (2010) 'Anti-localization anonymous routing for delay tolerant network', *Computer Networks*, Vol. 54, No. 11, pp.1899–1910.

Zamani, A.T. and Zubair, S. (2014) 'Secure and efficient key management scheme in MANETs', *IOSR Journal of Computer Engineering (IOSR-JCE)*, Vol. 16, No. 2, pp.146–158.

Zhou, J., Song, M., Song, J., Zhou, X. and Sun, L. (2014) 'Autonomic group key management in deep space DTN', *Wireless Personal Communications: An International Journal*, Vol. 77, No. 1, pp.269–287.