IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Public-key infrastructure validation and revocation mechanism suitable for delay/disruption tolerant networks

ISSN 1751-8709
Received on 14th January 2015
Revised 7th January 2016
Accepted on 15th January 2016
E-First on 31st March 2016
doi: 10.1049/iet-ifs.2015.0438
www.ietdl.org

Muhammad Nasir Mumtaz Bhutta[1] ✉, *Haitham Cruickshank*[2], *Zhili Sun*[2]

[1]*NexGin Research Center, Electrical Engineering Department, Institute of Space Technology, Islamabad, Pakistan*
[2]*Department of Electronic Engineering, University of Surrey, Guildford, UK*
✉ *E-mail: nasir.mumtaz@ist.edu.pk*

Abstract: Public-key infrastructure (PKI) is based on public-key certificates and is the most widely used mechanism for trust and key management. However, standard PKI validation and revocation mechanisms are considered major reasons for its unsuitability for delay/disruption tolerant networking (DTN). DTN requires mechanism to authenticate messages at each node before forwarding it in the network. So, certificate revocation lists (CRLs) being distributed in DTN network will need to be authenticated and validated for issuer certificate authority (CA) at each node. In this study, the authors propose new validation and revocation mechanism which is compliant with DTN semantics and protocols. This study also proposes a new design for CRL in compliance with standard PKI X.509 standard to make the proposed mechanism easy to implement for DTN. The new designed CRL is of reduced size as it contains fewer entries as compared with standard X.509 CRL and also arranges the revocation list in the form of hash table (map) to increase the searching efficiency.

1 Introduction

Delay and disruption tolerant networking (DTN) has evolved from a focus on deep space networks to a broader class of heterogeneous networks, e.g. wireless *ad hoc* networks, wireless local area networks etc. which may have the characteristics of long delays, asynchronous down and uplink latency and bandwidth, disruption, unexpected partitioning in the network etc. [1–5].

DTN handles the delays and disruptions at new layer called 'bundle layer or bundle protocol' (BP). BP provides store-and-forward kind of services in DTN for better performance as compared with standard TCP/IP protocols [1–5]. BP specifications in [3] provide bundle message format, operations of BP and details about custody transfer and convergence layer. BP also describes about extensions to provide security services including confidentiality, integrity and authentication by bundle security protocol (BSP).

BSP has provided a mechanism to prevent the unauthorised access and utilisation of DTN resources while also maintains the end-to-end confidentiality, integrity and authentication and hop-by-hop integrity and authentication by introducing different security blocks appended with actual payload [6]. The provision of all these security features is dependent on cryptographic keys and BSP assumes presence of keys on all the intermediate and end nodes. Setting up cryptographic keys is called 'Key Management' and is still an open issue to be solved [7–10].

Public-key infrastructure (PKI), the most widely used mechanism for key management in internet is considered unsuitable for DTN due to DTN characteristics of, intermittent-connectivity, long delays and limited storage space at intermediate nodes. PKI validation and revocation mechanisms are considered major obstacle for its suitability for DTN [11–14]. PKI uses two standard mechanisms, online certificate status protocol (OCSP) and certificate revocation list (CRL), for validation and revocation of certificates. OCSP is an online status checking protocol in which the status validation entity is always online to respond to the status checking queries. When an entity sends a request to check the status of a selected certificate, the responder checks the database and sends the status response signed by its own private key. Though as compared with CRL, OCSP provides fresh status information but the request response model may not be very suitable for DTN as link may not be available all the times. In DTN an offline mechanism such that CRL may be suitable for revocation and validation of certificates [10–13, 15, 16]. CRL is an offline mechanism in which the revoked certificate status list is populated and distributed in the network. However, to verify that CRL is actually signed by original CA at each intermediate DTN node does not look a viable solution and also as CRL grows, it becomes very difficult to manage. However, an offline mechanism such that CRL can be useful if the CRL size is manageable at intermediate nodes and validity of CRL can be checked by each intermediate node offline [10–13, 15, 16].

Our proposed scheme solves the above-described issues and provides an efficient and better way for intermediate nodes to easily verify the certificate of CA offline from our new proposed CRL as compared with original PKI CRL in compliance with BSP. The proposed mechanism generates CRL of reduced size in the form of hash table (map) to increase the searching efficiency.

The remainder of this paper is organised as follows: Section 2 presents the background on DTN, PKI and other related work. Section 3 explains the new proposed PKI validation and revocation mechanism in detail along with its design goals. Analysis and simulations of the proposed methods are detailed in Section 4 and at the end Section 5 concludes this paper.

2 Related work and background

To best of our knowledge, this paper is first an effort to propose a suitable PKI validation and revocation mechanisms for DTN. However, in the past some similar ideas are presented for other types of networks including vehicular networks. In this section, we discuss those ideas and analyse their suitability for DTN-based networks. This section also outlines the topics which are building blocks of our proposed mechanism.

CRLs can be feasible solution for DTN if their validity can be checked at each intermediate node in DTN and also their size is manageable. There are variations of CRLs already developed in the past. The simplest of all is the standard CRL as described in above section. CRL is the most mature approach and has been part of X.509. CRL is digitally signed list of revoked certificates and these signatures are needed to be validated at each intermediate node which is in contrast of intermittent communication link availability of DTN. However, there are also different variations of CRLs

available which are designed to increase the distribution performance in the network.

Overissued CRL provides option of issuing multiple CRLs overlapping the validity period of CRL. It provides a way to reduce the peak request rate of CRLs. Delta CRL (D-CRL) provides the list of the revoked certificates which are issued after the issuance of base CRL. Also, certificate revocation tree (CRT) and authenticated dictionary (AD) based on merkle hash tree (MHT) are also proposed to improve the efficiency of certificate verification process. CRT and AD contains the revoked certificates as MHT leaves and the root node of the tree contains the digest of all leaves nodes. The root is digitally signed by the revocation data issuer. So if root digest is verified, then it approves the authenticity of all leaves nodes. However, the main drawback of CRT is the management of dynamic binary tree. In case of freshness requirements, the addition, searching and removing leaf can be costly operation with $O(n)$ complexity [17]. To solve the bandwidth and processing capacity bottle necks in the revocation system and AD-MHT system is proposed by Jose in [17].

AD-MHT tries to utilise the benefits of both MHT and AD systems and propose way for revoking certificates, deleting expired certificates, the status checking protocol for communicating the AD-MHT repository with users, verifying response etc. However, for AD-MHT to be effective, certificate using applications must connect to any of the available AD-MHT repositories. In the event such connection cannot be obtained, AD-MHT proposes to use standard OCSP mechanism [17]. The absence of always available link is the basic assumption of DTN. That is why AD-MHT is not a useful option for DTN. Our proposed validation and revocation mechanism does not require the always presence of link to the CRL issuing CA.

On the basis of MHT, bandwidth efficient certificate status information distribution mechanism for VANETs (BECSI) is proposed in [18] to provide more reliability, less bandwidth usage and to improve freshness in PKI validation and revocation mechanism. BECSI arranges revoked certificates in the form of BECSI base-tree (MHT) and consists of four phases. In bootstrapping, the CRLs are distributed to road side units (RSUs) in VANETs. In the second phase, RSUs and any other moving vehicle containing complete CRL become repositories which can respond to revocation queries from the other mobile nodes. These new repositories first verify the signatures of CRL issuing CA by online mechanism and then become available to respond to revocation queries by other nodes. In the third phase, these new repositories can also update the CRL and download updated D-CRLs. The issuing CA also releases the information about how many number of MHTs have been released to keep all nodes informed for freshness. In the fourth stage, the new repositories start responding to revocation queries. The nodes can verify either querying to online available these new repositories or can download the CRL from them. This BECSI is similar to distributed revocation model where multiple repositories can respond to queries. However, still it is request/response model where link presence is very important and is not in compliance with DTN design assumptions. Also, the designed MHT-based BECSI trees contain hashes of each revoked certificates and root node in the tree act as digest for all nodes. This hashes based mechanism is fast as compared with digital signatures verification, but to verify the root node still digest verification of root node is required which will be very costly operation for DTN where each intermediate node can hold the CRL and will have to verify the signatures of root for each MHT tree. Our proposed mechanism verifies the issued CRL offline based on NOVOMODO mechanism and does not require the presence of link to CA. Also, the formation and updation of BECSI tree can be very costly as two nodes in leaves are combined to calculate hash of upper layer parent node and this process is applied repeatedly until there is only one node at highest level which is termed as root node. Our proposed new CRL mechanism consists of hash table where each node can be directly mapped and similarly can be verified in just single operation. Also, the revocation proof verification in our proposed mechanism is a just one time hash calculation as compared with multiple hashes calculations in BECSI.

A similar idea to BECSI is presented in [19] called collaborative certificate status checking mechanism for VANETs (COACH). COACH is a small variation of BECSI and all design and working concepts are same for CRL design and distribution to BECSI. The paper in [19] proposes a new extension to COACH as well called EvCOACH that is more efficient than COACH for scenarios where revocation rates per CRL validity period are low. However, the design assumptions of BECSI and COACH are same and are not in lined with DTN design assumptions as discussed above.

Now in this section, we discuss the concepts which are building blocks of our proposed scheme to resolve above-described issues.

2.1 Public-key infrastructure

PKI is an infrastructure to manage the digital certificates. PKI provides mechanisms to generate, store, distribute and revoke the digital certificates which are means to associate the public key to a specific entity signed by a CA. A certificate authority is a trusted entity in the network which is responsible for issuing certificates. There can be different levels of CAs to issue certificates to different entities. There is one root CA to issue certificates to itself and other sub-ordinate CAs who can issue certificates to other CAs or end entities [14].

2.2 NOVOMODO: scalable certificate validation and simplified PKI management

NOVOMODO is a validation mechanism based on X.509 V3 certificates. NOVOMODO introduces the concept of time granularity for which the certificate is considered to be valid and is different from certificate expiry date. NOVOMODO appends two 20 bytes hashes with the certificate. It also breaks the total time span between certificate issuance and expiry time into smaller units. That smaller unit is the time period for which the certificate must remain valid. It is proposed in NOVOMODO to generate the chain of hash functions equal to the selected time units for certificate expiry date. Let us say a certificate is valid for 1 year (365 days) and time granularity is selected to be 1 day. According to NOVOMODO, a hash chain of 365 hashes ($X_0, X_1, \ldots, X_{364}$) is needed to be calculated from a randomly selected initial seed of 20 bytes known only to server. The last calculated hash X_{365} is appended in the V3 certificate before issuing to any entity. NOVOMODO also adds one more hash Y_1 in the X.509 V3 certificate which is calculated from another randomly selected value Y_0. Now suppose an entity on the ith day sends a request to validate the certificate. The CA has to make decision based on the data from its local database. If the certificate is valid, CA will select the ith hash value X_i and append it with the certificate serial number and sends this as a response to certificate status checking request. If the certificate was revoked, then the issuing CA would have selected the Y_0 and would have sent it along with the serial number. The receiving entity can use the X_i and produce the X_{i+1} to validate the certificate [20, 21].

2.3 Hash chain

A hash is a fixed length, unique value which is produced by using a special function called cryptographic hash function $h(x)$, where x is any value. A hash function is called one-way function as it is computationally infeasible to generate the original value from the hash value. A hash chain is a list of many hashes by applying the hash function $h(x)$ multiple times [22].

2.4 Hash table

A hash table is a data structure used to implement an array which can store values against some keys. It uses a hash function to compute the index of array. Ideally, each key should be mapped to a unique memory location but practically it is not possible due to hash collisions. In hash collision, different keys can be hashed to same memory location. An example of hash table is shown in Fig. 1.

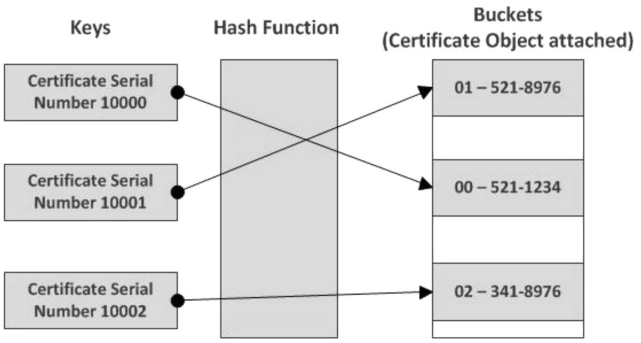

Fig. 1 *Example of hash table*

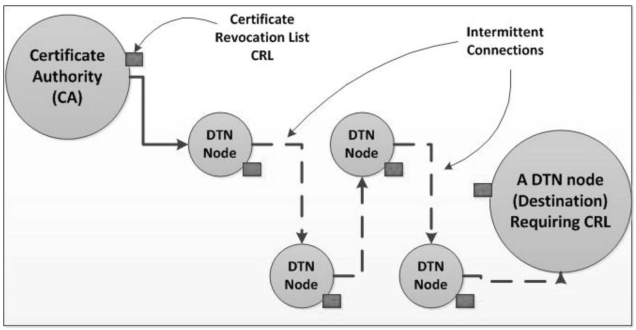

Fig. 2 *DTN network showing the 'New Proposed CRL' distribution mechanism*

In many situations, hash tables are more efficient in searching rather than using search trees or table lookup. So, these are widely used to represent sets or lists. Table 1 shows the complexity analysis of hash table, where $O(x)$ is the upper bound symbol to represent complexity analysis, n represents the number of items in the hash table (list) and k represents the number of items at one location in hash table in case of collision [23].

3 Proposed new PKI validation and revocation mechanism

3.1 Design and security goals

The design goals are: to design a CRL of reduced size, efficiently searchable and to distribute CRL in DTN network in a way that each intermediate receiving node in DTN can validate the CRL offline thus making it compliant with BSP. The compliance with BSP means that CRL should be authenticated at each intermediate node that it is sent from a valid CA. The security goals are to ensure that CRL should not be forged and it should be always ensured that CRL is issued from a valid CA before being forwarded by any DTN node. As CRL information is public so it is not required to encrypt it.

3.2 Network model

We consider a DTN network formed by a set of mobile nodes with unique name N_i for $i = 0, 1, 2, 3, \ldots, m$. Each node in the network has a public-key certificate based on NOVOMODO associated with its unique name N_i issued by a CA. Each node also knows the algorithm used by CA to compute the hashes for validity and revocation proofs. In a particular communication, only the public-key certificate of CA is known by all nodes in the network and is to be validated at each node when required. We also assume that each node receives its validity proof hash for its certificate from its issuer according to selected granularity period. It is always assumed that CA cannot be compromised for the whole life of description of the system. We also assume that nodes in the network have limited communication range so two nodes outside of the communication range of each other can only communicate by the participation of other nodes in a multi-hop manner. The end-to-end connectivity is not always present so routing is done in an opportunistic way.

Table 1 Complexity analysis of hash table (map)

Operation	Average complexity	Worst case complexity
insert	$O(1)$	$O(n)$
search	$O(1 + n/k)$	$O(n)$
delete	$O(1 + n/k)$	$O(n)$

3.3 System initialisation

In initialisation phase the CAs and end entities are issued with the certificates. The issued certificate is X.509 version 3.0 which provides the facility to include any other information as extension. In DTN, different applications may want the certificate to be valid for different minimal periods of time depending on security requirements. Suppose the certificate issued are valid for 100 days and application specific selected minimal period T for which a certificate must be valid. This time granularity concept is very suitable for DTN environment, as depending on the delay for a selected scenario, the granularity period for the validity of certificate can be set up from seconds to months and for our selected scenario, it is 1 day. Here, we produce a hash chain of 100 length for selected arbitrary value X_0 which corresponds to validation just like NOVOMODO. The hash chain generated will be $X_0, X_1, X_2, \ldots, X_{99}$ by applying the hash function $h(x)$ repeatedly.

The last item X_{99} of the hash chain is appended in the issued certificate as an extension. Another hash value called Y_1 is also appended in the issued certificate which is generated from second arbitrary selected value Y_0. This second hash is used to verify that a certificate is revoked.

So, during system initialisation all the issued certificates will contain two extra hash values X_i and Y_1 as extension fields in the certificate. These hashes X_i and Y_1 will help to validate the status of certificate later on [20–23].

3.4 CRL generation and distribution

We propose to build a CRL of revoked certificates using hash table data structure. The CRL will contain a hash table built on the serial number of revoked certificates along with validity or revocation proofs hashes. Fig. 2 shows an example of new proposed CRL distribution in DTN networks.

When any node in the network receives the CRL, first it will validate the CA certificate offline by computing the hash in the original CA certificate from the hash X_i received along with CRL according to granularity period. After successful verification of CA certificate, it is ensured that CRL is coming from right CA.

Once CRL is verified that it is coming from actual CA, the receiving node stores CRL in its persistent storage and then forwards it to other node according to routing algorithm. The stored CRL can be kept for validity period of CRL.

3.5 Design of new proposed CRL

In this section, we describe the new CRL design in detail and compare our new CRL design with X.509 CRL. The new CRL is designed to provide same information as X.509 CRL provides but with fewer fields.

Fig. 3*a* shows the fields which are present in a usual X.509 CRL. The X.509 CRL is constructed by setting signature algorithm Id, the issuer distinguished name (DN), this update field as well as next update field together with list of all revoked certificates and some mandatory extensions. The sequence up to extensions field is then signed and signature is appended to the CRL. The field, 'Revoked Certificates', is a complex field and contains list of revoked certificates. As CRL grows, the huge size of CRL is dependent on this 'Revoked Certificates' field. Each entry in 'Revoked Certificates list consists of 39 bytes including a 6 bytes long serial number of revoked certificate, a 13 byte long entry for revocation date and a 12 bytes long entry for reason code [24, 25].

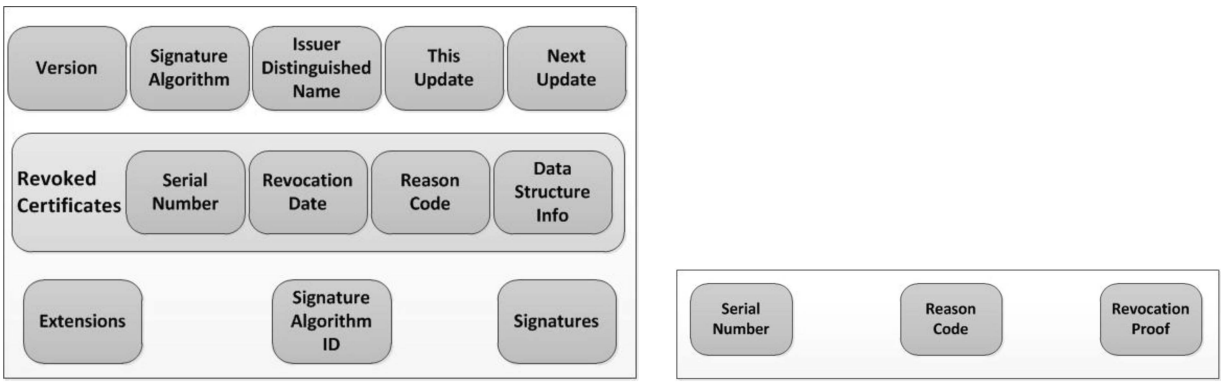

Fig. 3 *Design of new proposed CRL and comparison with standard X.509 CRL*
a Original X.509 CRL record showing all fields
b New proposed CRL record showing all fields

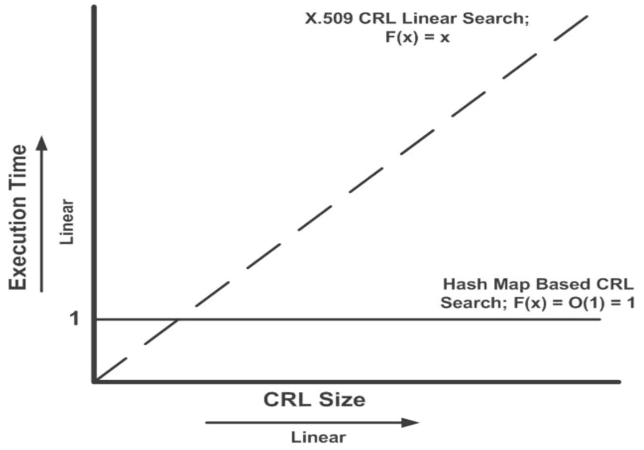

Fig. 4 *Complexity analysis of hash table-based new CRL*

The size of mandatory fixed fields in X.509 CRL has been measured to be 400 bytes. So, a total size of X.509 certificate can be calculated as follows:

$$\text{size of X.509 CRL} = (\text{number of revoked certificates} \times 39 + 400)\ \text{bytes}$$

In parallel to this X.509 CRL, we propose a new CRL design as shown in Fig. 3*b* based on NOVOMODO and is compliant with X.509 data structures. Each entry in this new CRL is of 38 bytes compared with X.509 39 bytes entry, but there are no other fixed entry fields. Each entry in the new proposed CRL is formed by a 6 bytes serial number of revoked certificate, a 12 bytes long reason code and a 20 bytes long field of revocation proof hash. The new CRL can also include information about hash generation algorithm as an option. Thus a new CRL built for same number of revoked certificates will be of smaller size as compared with X.509 CRL. The size of new CRL can be calculated as follows:

$$\text{size of new CRL} = (\text{number of revoked certificates} \times 38)\ \text{bytes}$$

Though new CRL consists of fewer fields as compared with X.509 CRL, yet same information can be deduced from it as X.509 CRL provides. The signature algorithm and issuer DN are not required to be included in new CRL since if the hash contained in the certificate can be computed from the hash available in the revocation proof field, then it is guaranteed that the proof is coming from the actual issuer and there is no need to sign the hash to prove the identity of CA. In X.509 CRL, 'This Update' and 'Next Update' fields provide the dates on which the current CRL was issued and what is the next issuance date of new CRL, whereas in new CRL there is no need to provide this information. The 'This Update' and 'Next Update' information can be deduced from the granularity period chosen at the time of certificate generation. We have seen that design of new CRL is smaller as compared with X.509 CRL, but still it provides equivalent information as provided by X.509 CRL.

4 Evaluation of the proposed mechanism

This section focuses on evaluating the proposed new CRL from various aspects according to the selected design and security goals.

4.1 Complexity analysis of searching in hash table (map)-based new CRL

This section analyses the complexity of hash table-based CRL. We build the new CRL by inserting the serial number of revoked certificates as keys for hash table and revocation or validation proof hash. We know that insertion operation average complexity of hash table is $O(1)$, searching operation average complexity in hash map is $O(1)$ if no collision occurs while searching operation average complexity through a simple sorted list is linear as used in X.509 CRL. If we can define functions for different operations complexities as, for building hash table-based CRL average complexity of $f(x) = O(1)$, searching for a revoked certificate in hash-based CRL average complexity of $f(x) = O\ (1 + n/k)$ and for linear search in X.509 CRL as $f(x) = x$, then we can draw these functions behaviour on a graph whose the x-axis represents CRL size and the y-axis represents execution time as shown in Fig. 4. From figure, it is clear that as x value increases, the graph for $f(x) = x$ will always increase in a constant fashion along with x while the $f(x) = 1$ will always be constant.

From the above analysis, it is obvious that new CRL searching operation is very fast and always constant independent of the size of CRL while the searching operation through a basic sorted list is always linear and will take longer as size of list will increase.

4.2 Compliance with X.509 PKI

The new CRL is compliant with X.509 PKI standard thus should be easy to implement in real scenarios. Here, we review all the modifications proposed with reference to X.509 PKI standard.

During the system initialisation, two hashes each of 20 bytes are appended into the X.509 V3 certificates to get certificate validated and revoked later on. The addition of these hashes into X.509 V3 certificate is compliant with the PKI X.509 certificate specifications which allow adding any extensions to the certificate.

We have reduced the fields in the new CRL design as compared with X.509 CRL design. However, all the entries in the CRL are in compliance with X.509 specifications. So, new CRL can easily replace the old X.509 CRL without so many modifications in the implementation of the system.

4.3 Security analysis of the proposed mechanism

In security design considerations, we emphasise that the proof of revocation cannot be forged when transmitted in the network and also CRL is validated on each DTN node before being forwarded in DTN network to filter bogus messages. It was also emphasised that new CRL design helps to achieve compliance with the BSP specifications. Here, we analyse the new CRL with respect to set goals.

4.3.1 Proof of revocation and validation cannot be forged: The invocation proof present in the new CRL cannot be forged. Since, invocation proof hash is reverse of certificate revocation field Y_1. Since hash is practically impossible to invert, so once a verifier checks that a given 20 byte revocation proof hash can compute Y_1, it knows that revocation proof must have been released from the actual CA. In fact, only the actual CA can compute the inverse of Y_1, not due to the reason that CA can invert the Y_1 but because it computed Y_1 from a hash value Y_0 when issuing the certificate. Since, the certificate issuing CA never releases the proof of revocation as long as the certificate remains valid, no malicious entity can fake a revocation proof. So a revocation can be publically distributed on an unsecure channel without worrying about that it can be forged [20, 21]. On the same idea, validation proof can also not be forged. Since, validation proof hash is computed as part of hash chain for X_0 to X_n. Since hash is practically impossible to invert, so once a verifier checks that a given 20 byte validation proof hash can be computed by iterative hash computation of X_k depending on the granularity and time passed since certificate was issued, it knows that validation proof must have been released from the actual CA.

4.3.2 Offline CA certificate validation and compliance with BSP: The new CRL distribution in the DTN network is compliant with the BSP specifications as on each hop the original CA certificate is validated and is authenticated which are the security design goals of the BSP. As described above, the certificate of CA can be validated by re-computing the hash contained in the X.509 certificate according to granularity set by the system with the hash received along with CRL from the CA. If the hash can be re-computed, then certificate is valid and as hash cannot be forged, so it is assumed that it is coming from the claimed CA; otherwise, the CA certificate is considered invalid.

4.4 Simulations and performance evaluations

4.4.1 Cryptographic overhead calculations: X.509 CRL-based and new proposed CRL-based revocation and validation mechanisms involve many security operations at receiving nodes when CRL is distributed, e.g. signature verification, hash computation etc. To investigate the performance of revocation mechanisms, we first study the execution time of the cryptographic operations involved in its operation. For the evaluation of delay of cryptographic operations, the used machine specifications are given below in Table 2.

On the selected machine, we implemented all the security operations involved in revocation mechanisms and calculated their execution time. We run the programmes for ten times and average the results. The result of execution time for selected operations and algorithms are presented below in Table 3.

4.4.2 Simulations: The validation and revocation mechanism, based of new CRL design, was implemented on opportunistic networking environment simulator [26] specially designed for DTN and evaluated its performance on a selected scenario based on mobile nodes. The details of simulations parameters are summarised in Table 4.

Certificate validation and revocation method was implemented on 'Spray and Wait (SW)' binary mode routing protocol. The SW routing protocol is a deterministic and multi-copy protocol which achieves high delivery ratio while utilises very low resources. Since, it is deterministic protocol and number of forwarding copies can be controlled which is why we have selected this for simulations. The revocation mechanism can be implemented on any other routing protocol for DTN as well [4, 26].

For evaluation of performance for selected scenario, we compared the simulation performance between X.509-based revocation mechanism as benchmark and new proposed CRL-based revocation mechanism. The network metrics selected for evaluation of system performance are given below (see equation below) To evaluate the practicability of new proposed revocation method, we examined the system performance under different traffic loads by varying CRL size. For this analysis, we fixed the number of deployed nodes to 250 for the selected city map. The SW routing protocol number of forwarding copies were 10% of total deployed nodes. Figs. 5*a*–*c* show the performance comparison between X.509-based CRL and new proposed CRL-based revocation methods.

Fig. 5*a* shows the relationship between delivery ratios and different CRL sizes. It is observed that successful delivery ratio degrades with increase in CRL size. The decrease in the delivery ratio is due to increase in traffic load a network can handle and computational processing to verify signatures and re-computing

Table 2 Summary of specifications of machine

laptop	X86, 32 bit
processor	Intel Core Duo central processing unit, at 2.40 GHz, at 2.40 GHz
random access memory	2 GB
operating system	Windows 7
JDK version	1.6

Table 3 Details of execution time of cryptographic operations

Algorithm	Key size, bits	Operation	Average execution time, s	Standard deviation of execution time, s
MD5 with RSA	1024	verify	0.0503	0.0022
MD5	128 (hash length)	hash	0.0109	0.0033

Table 4 Parameters for new CRL-based revocation mechanism simulation

Parameter	Value range
duration	12 h
number of nodes	250
speed of nodes	10–70 km/h
transmission coverage	100 m
mobility model	map-based mobility
message size	variable
message generation interval	25–35 s
routing protocol	SW protocol
number of forwarding copies	25 (10% of deployed nodes)

$$\text{delivery ratio} = (\text{messages delivered}/\text{messages relayed})$$

$$\text{average latency} = \text{average}(\text{time at messages delivered} - \text{time at messages created})$$

$$\text{overhead ratio} = (\text{messages relayed in network} - \text{messages delivered to destinations}) / (\text{messages delivered to destinations})$$

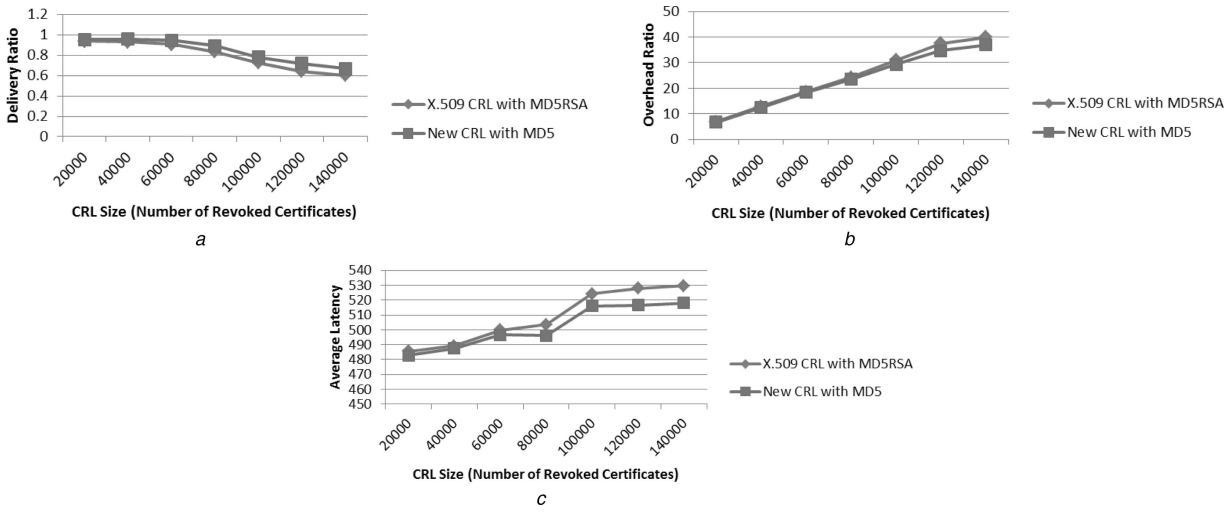

Fig. 5 *Performance comparison between X.509-based CRL and new proposed CRL-based revocation methods*
a Impact of CRL size: delivery ratio
b Impact of CRL size: overhead ratio
c Impact of CRL size: average latency

hashes. However, we can see that new CRL-based revocation mechanism gives slightly better performance as compared with X.509 CRL-based revocation method. The delivery ratio decreases faster as the traffic load increases above a specific threshold (for example, after 60,000, 80,000 number of revoked certificates in Fig. 5*a*) and before that threshold there is a gradual decrease in the delivery ratio when number of revoked certificates are <60,000. This shows that delivery ratio is highest at some optimal number of revoked certificates (60,000–80,000) CRL is distributed and once number of revoked certificates increase that threshold the delivery ratio starts decreasing fast as traffic becomes higher than the optimal capacity of network system.

However, it can also be observed that security computation overhead and a bit high size of X.509 CRL degrades performance compared with new proposed CRL. Fig. 5*a* clearly shows that when number of revoked certificates is low, the difference between both schemes is very close. However, when CRL becomes very high, our proposed CRL gives better performance in terms of delivery ratio as compared with X.509 CRL.

Fig. 5*b* demonstrates that X.509 CRL revocation mechanism has slightly larger overhead as compared with new proposed CRL-based revocation mechanism. The overhead ratio for large CRL size is slightly more in the X.509 CRL-based revocation mechanism as compared with new proposed CRL-based revocation mechanism. However, the overhead in both revocation mechanisms are almost identical and slightly larger overhead in X.509 CRL-based revocation mechanism.

From Fig. 5*c*, it can be observed that average latency is increased while the CRL size in the network is increased. Here the average latency for X.509 CRL-based revocation mechanism and new proposed CRL-based revocation mechanism is not significantly different and the simulations have demonstrated similar performance in terms of average latency.

5 Conclusion

This paper has described in detail a novel mechanism of certificate validation and verification for existing PKI-based infrastructures. The new NOVOMODO-based CRL mechanism enables all the intermediate nodes in the DTN network to validate the signatures of CA who issued CRL without contacting the root CA (or higher rank CA) by calculating hash thus making it in compliance with BSP working for DTN. It is also analysed in security analysis of new proposed scheme that hash-based revocation and validation mechanism cannot be forged. Standard PKI validation and revocation mechanism is also enhanced by our new scheme by enabling the applications to build CRL of reduced size and to search efficiently through the list as compared with standard PKI CRL as shown in complexity analysis of CRL. It is also observed via simulations that the new proposed revocation mechanism gives slightly better performance as compared X.509 CRL-based revocation mechanism and besides that it remains in compliance with BSP specifications, whereas X.509-based revocation mechanism will not be viable to work in compliance with BSP specifications due to the fact of verifying CA certificate via OCSP method. In summary, we can say that new proposed CRL-based revocation mechanism is indeed a viable and lightweight solution for distributing CRLs in DTN environment.

6 References

[1] Fall, K., Farrell, S.: 'DTN: an architectural retrospective', *IEEE J. Sel. Areas Commun.*, 2008, **26**, (5), pp. 828–836
[2] Fall, K.: 'A delay tolerant network architecture for challenged internets', IETF, August 2003
[3] Scott, K., Burleigh, S.: 'Bundle protocol specifications', IRTF DTN Research Group, RFC 5050, November 2007
[4] D'Souza, R.J., Jose, J.: 'Routing approaches in delay tolerant networks: a survey', *Int. J. Comput. Appl.*, 2010, **1**, (17), pp. 975–8887
[5] Huang, M., Chen, S., Zhu, Y., *et al.*: 'Topology control for time evolving and predictable delay-tolerant networks', *IEEE Trans. Comput.*, 2012, **62**, doi: 10.1007/s11276-012-0520-6, pp. 2308–2321
[6] Symington, S., Farrell, S., Weiss, H., *et al.*: 'Bundle security protocol specification', IRTF RFC 6257, May 2011
[7] Seth, A., Keshav, S.: 'Practical security for disconnected nodes'. Secure Network Protocols (NPSEC), November 2005, pp. 31–36
[8] Farrell, S., Cahill, V.: 'Security considerations in space and delay tolerant networks', *IEEE SMC-IT*, 2006, **8**, p. 38
[9] Farrell, S., Lovell, P.: 'Delay-tolerant networking security overview', draft-irtf-dtnrg-sec-overview-06, IETF draft, March 2009
[10] Farrell, S.: 'Security is the wild', *IEEE Internet Comput.*, 2011, **15**, (3), pp. 86–91
[11] Bhutta, N., Ansa, G., Johnson, E., *et al.*: 'Security analysis for delay/disruption tolerant satellite and sensor networks'. Satellite and Space Communications, IWSSC 2009, 2009, Tuscany
[12] Farrell, S.: 'DTN key management requirements', IETF drat, June 2007
[13] Cooper, D., Santesson, S., Farrell, S., *et al.*: 'Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile', IETF RFC5280, May 2008
[14] 'Public Key Infrastructure', Article available on website. Available at http://www.en.wikipedia.org/wiki/Public_key_infrastructure, accessed February 2013
[15] Myers, M., Ankney, R., Malpani, A., *et al.*: 'X.509 internet public key infrastructure online certificate status protocol – OCSP', IETF RFC2560, June 1999
[16] Santesson, S., Hallam-Baker, P.: 'Online certificate status protocol algorithm agility', IETF RFC6277, June 2011
[17] Munoz, J.L., Forne, J., Esparza, O., *et al.*: 'Certificate revocation system implementation based on the merkle hash tree', *Int. J. Inf. Sec.*, 2004, **2**, pp. 110–124, doi: 10.1007/s10207-003-0026-4
[18] Ganan, C., Munoz, J.L., Esparza, O., *et al.*: 'BECSI: bandwidth efficient certificate status information distribution mechanism for VANETs', *Mob. Inf. Syst.*, 2013, pp. 347–370, doi: 10.3233/MIS-130167, IOS Press
[19] Ganan, C., Munoz, J.L., Esparza, O., *et al.*: 'COACH: collaborative certificate status checking mechanism for VANETs', *J. Netw. Comput. Appl.*, 2013, **36**, pp. 1337–1351

[20] Micali, S.: 'NOVOMODO: scalable certificate validation and simplified PKI management', US Patent Application Number EP20020719315, Publication Date, 12/17/2003
[21] Micali, S.: 'NOVOMODO: scalable certificate validation and simplified PKI management'. First Annual PKI Research Workshop, October 2003
[22] Perrig, A., Canetti, R., Tygar, J.D., *et al.*: 'Timed efficient stream loss-tolerant authentication protocol', IETF RFC 4082, June 2005
[23] 'Hash Table', article available on http://www.en.wikipedia.org/wiki/Hash_table, accessed 18 February 2013
[24] Park, K.-W., Lim, S.-S., Park, K.-H.: 'Computationally efficient PKI based single sign-on protocol PKASSO for mobile devices', *IEEE Trans. Comput.*, 2008, **57**, (6), pp. 821–834
[25] Perlines-Homann, T., Wrona, K., Holtmanns, S.: 'Evaluation of certificate validation mechanisms', *Comput. Commun.*, 2005, **29**, pp. 291–305
[26] Keränen, A., Kärkkäinen, T., Ott, J.: 'Simulating mobility and DTNs with the ONE', *JCM*, 2010, **5**, (2), pp. 92–105