A DISTRIBUTED OCSP FRAMEWORK FOR AD-HOC NETWORKS

Giannis F. Marias
Dept. of Informatics and Telecommunications
University of Athens, Greece, TK15784

Konstantinos Papapanagiotou
Dept. of Informatics and Telecommunications
University of Athens, Greece, TK15784

Panagiotis Georgiadis
Dept. of Informatics and Telecommunications
University of Athens, Greece, TK15784

ABSTRACT

Many solutions for establishing trust in mobile ad hoc networks (MANETs) involve public key cryptography. Most of these solutions, including proposals for routing protocols, suggest the deployment of public key certificates. An efficient mechanism for certificate revocation and validation is essential in every system that uses certificates. Consequently, such a scheme is required for MANETs, too. However, their infrastructure-less nature poses additional issues which are not apparent in traditional networking. In this paper, we propose an on-demand, distributed scheme based on Online Certificate Status Protocol (OCSP). This scheme, called Ad-hoc Distributed OCSP for Trust (ADOPT), utilises cached OCSP responses which are distributed and stored on intermediate nodes. Our main goal is to avoid exchanging large certificate status lists among the ad hoc nodes, avoiding needless consumption of scarce bandwidth and the energy of autonomous nodes. We also discuss alternative design approaches concerning the distribution of cached OCSP responses, to maximize the efficiency of ADOPT, taking into account network connectivity and topology.

KEYWORDS

MANET, certificates, revocation, OCSP

1. INTRODUCTION

Mobile ad hoc networks are dynamically configured peer-to-peer and multi-hop networks, where the network topology varies with time due to the ataxic locomotion of the participating nodes. They can be materialised as completely autonomous, self-organised, spontaneous networks, where the nodes are responsible for their creation, operation and maintenance in an absence of a central coordination entity, such as base stations. This type of MANETs will be referred to as AANs (Autonomous Ad hoc Network). Applicability examples include the ad hoc networks that serve wireless communication between entities of rescue teams or military troops in the front line, such as the Tactical Internet [0]. Capkun et al. propose the use of the term “fully self-organised” ad hoc networks [0], whilst the IEEE nomenclature uses the term Independent Basic Service Set (IBSS), which is the formal name of an ad hoc network in the IEEE 802.11 standard [0] [0]. Additionally, ad hoc networks can be viewed as an extension of terrestrial, wired infrastructures, such as cellular networks or the Internet. In such architectures, one or more nodes that maintain a connection with the wired network, act as relays, offering access to nearby ad-hoc devices, extending the coverage area. Examples include collaboration environments, where nodes in the proximity of a WLAN IEEE 802.11 base station offer connectivity to distant nodes. We will refer to this category of MANETs as CANs (Connected Ad hoc
Networks). For the IEEE 802.11 standard, this type of network is referred to as Extended Service Set (ESS) [0]. Error! Reference source not found. illustrates a possible deployment of AAN and CAN MANETs.

Figure 1. MANETs of CAN and AAN type

CANs X and Y can be considered as network extensions of providers X and Y, respectively. Nodes of these CANs can be connected (i.e. associated), whilst nodes for the AAN can establish connections with the nodes of a CAN. An ad hoc network, by definition, is dynamic and open: nodes constantly join or leave the network, the locomotion of nodes modifies the network topology, whilst, the transfer medium is the electromagnetic spectrum, and, thus, should be considered as publicly available. In such a dynamic and open environment, selfish, malicious and hacker nodes can attack on all fronts, aiming to violate information messages’ confidentiality and integrity, entities’ authenticity, network’s robustness and nodes’ availability. A selfish node is reluctant to spend its resources to maximize social welfare (e.g., forward packets not destined for it). However, it may demand from others to exploit their resources for network’s tasks that maximize its profit (e.g., intends for the delivery of packets originated from or destined for it). Malicious nodes attack the robustness of the network and nodes’ availability. Flooding and sleep deprivation torture [0] are Denial of Service (DoS) techniques commonly used by malicious nodes. Hacker nodes’ immediate objective is the interception of the information that is exchanged between the ad-hoc nodes, and node impersonation. Thus, the violation of the messages’ confidentiality and the entities’ authenticity are their primary goals, with indirect threat on the nodes’ availability and the network’s robustness, since trusted nodes are isolated and packet forwarding is performed selectively. Hacker intents are materialised through passive eavesdropping, sinkhole and wormhole attacks and through active impersonation and Sybil attacks [0]. To prevent attacks on the network performance and on the nodes’ availability, to preserve entities’ authentication and to ensure confidentiality and integrity of messages, several proposals have been made, based on the usage of public key cryptography, digital signatures and certificates. A certificate specifies a binding between an entity (associated with its attributes) and a public key. Public key certificates and digital signatures require the existence of a Certification Authority (CA) for the issuance, distribution, renewal, revocation and validation of entities’ public keys. CA incorporates a public-private key pair, with the private key in duty to sign certificates that bind ad hoc entities (i.e., nodes) to public keys. Thus, every node should know the CA’s public key. The CA has to be online in order to revoke issued certificates, since a node’s private key might be compromised, or a node might be no longer trusted. Thus, certificates might not longer be valid.

In this paper we propose a scheme that is based on a distributed version of the Online Certificate Status Protocol (OCSP), applicable to MANETs. The proposed scheme utilizes caches of OCSP responses to examine certificates’ validity. OCSP caches are distributed and stored on intermediate nodes, avoiding the exchange of extended certificate status lists. These caches, stored on nodes with specific characteristics, such as energy autonomy and cellular or WLAN connectivity, are updated dynamically. An ad-hoc node uses an on-demand protocol to find the closest OCSP cache that is able to provide the status of the requested, peer’s certificate. This paper proposes alternative design approaches of the method, referred to as ADOPT (Ad-hoc Distributed OCSP for Trust) to enable the revision and distribution of up-to-date OCSP responses in AANs and CANs.

The structure of this paper is as follows: First we provide a survey of the proposed certificate-based trust architectures in MANETs. In Section 3 we discuss existing solutions for certificate revocation in MANETs that rely on CRLs. We also briefly describe OCSP and its advantages compared to CRLs. Subsequently, in Section 4, we present the proposed ADOPT scheme, and in section 5 we discuss alternative approaches for
caching to support the ADOPT framework. Finally, we provide some concluding remarks and directions for further research.

2. RECENT DEPLOYMENTS OF CERTIFICATE SCHEMES IN MANETS

In AANs, the main concern for the deployment of a centralised CA is that this approach produces a single point of failure. The accessibility of the CA entity, due to nodes’ mobility, the availability of the node’s resources (e.g., battery) that accommodate the CA, and the savage when this node is impersonated, attacked or even compromised, are some potential issues of a single CA scheme. Zhou and Haas have proposed a distributed key management scheme, based on threshold cryptography [0]. In this (n, r) threshold scheme there is not a centralised CA, but n distributed CA servers. The scheme is r-tolerant, since the CA service is distributed between n nodes in such a way that if r nodes are compromised the remaining n-r nodes can be used to sign or validate certificates. Additionally, any r-1 parties can jointly provide the CA service, but it is infeasible for r parties to do so. In [0], the CA functions are distributed through a threshold secret sharing mechanism, in which each ad-hoc node holds a secret share and multiple nodes in a local neighbourhood jointly provide complete services. An online CA service in MANET that is based on threshold cryptography, called MOCA, is described in [0]. In the MOCA framework, n MOCA nodes provide the functionality of a CA to the whole network. Using threshold cryptography, these n MOCA servers share the CA’s private key and any set of k MOCA nodes can reconstruct the full CA key. On the other hand, in a CAN type of MANET, the CA services are provided by nodes that reside on the terrestrial, wire line, network infrastructure. In [0] GSM/GPRS technologies are proposed, enabling the nodes to access these CA services. An off-line CA is considered in [0] to control an ad hoc network of mobile nodes. This CA decides which nodes can join the network, and assigns a unique identity to each one. Each mobile node holds a certificate signed by the CA, which binds its identity to its public key, and maintains a copy of the CA’s public key, so that it can validate the certificates of other nodes [0]. The ARAN secure routing protocol [0] requires the use of a trusted CA server whose public key is known to all nodes. Each node keys are generated in advance, and exchanged through an out of band relationship with the CA server. Before entering the ad hoc network, each mobile node should obtain its certificate from the CA server.

In MANETs, digital certificates and signatures are employed to protect both routing messages as well as data packet forwarding. Secure routing protocols, such as ARAN [0], SAODV [0], and forwarding modules, such as TRM [0], involve CAs. When digital certificates are used, the nodes need to hold CA’s public key to validate the signatory. In any case, the binding between the ad-hoc entity and its public key, and the validation of this binding should be provided online to the nodes, when requested. Managing and providing status information of certificates is essential and has received much attention in recent years for traditional networks [0]. On the other hand, ad-hoc nodes often have limited resources, including computational capability, storage, and power supply. Thus, the certificate’s status validation procedure should take into account these limitations, and avoid the creation, distribution and processing of computationally heavy Certificate Revocation Lists (CRLs) [0]. Moreover, due to node’s mobility, entities that provide certificate status information should be selected carefully, since peer entities require high availability of these nodes, whilst the avoidance of certificate status request flooding is an essential requirement for saving bandwidth.

3. CERTIFICATE STATUS INFORMATION IN AD-HOC NETWORKS

Many solutions that enable trust in mobile ad hoc networks and rely on public key certificates suggest a way of revoking these certificates and disseminating the revocation information in the network. In this section we examine some of these solutions, focusing on the method they propose for certificate revocation. We also present OCSP and its advantages, compared to CRLs. In MOCA [0], at least k out of n MOCA nodes have to agree in order for a certificate to be revoked. A MOCA node may generate a “revocation certificate” which contains a certificate which is revoked. This is signed with its key share and then broadcasted to the network. A node that receives at least k partially signed revocation certificates can construct the full revocation certificate. The list of revoked certificates can be stored at any node of the network, possibly at the MOCA servers, specially designated nodes or every node in the MANET. MOCA, however, does not specify a
protocol for certificate validation. Moreover, flooding the network with partially signed revocation certificates creates an unnecessary overhead, which, according to [0], is not a serious drawback as revocation should be a rather rare event. ARAN [0] is a secure routing protocol in which nodes use public keys to establish trust. These keys are provided by a trusted authority, which also has the power to revoke keys. When a key is revoked the trusted server broadcasts a message in order to notify all the nodes of the network. Each node receiving the revocation message rebroadcasts it and thus, eventually all nodes get informed. This, rather simple, approach suffers from several drawbacks. For example, a malicious node in the network may choose not to propagate the revocation message. Dormant nodes may cause network partitioning as they won’t be able to forward revocation messages. Moreover, they also need to get informed about revoked keys as soon as they become again active nodes in the network. Consequently, a significant amount of time may be needed to inform all the nodes participating in the network of the newly revoked keys. J. Cheambe et al. [0] propose a secure authentication scheme for MANETs. In their model, nodes can communicate with a CA by out of band means, for example using a GPRS or UMTS bearer. Each node has a unique, factory-installed, pair of keys which is used to communicate with other nodes. Keys are revoked by the CA which also issues and distributes CRLs [0]. CRLs are distributed using a simple authentication and communication procedure. When two nodes wish to mutually authenticate each other’s certificate they exchange, among others, the latest CRL they have. Moreover, new CRLs are flooded in the network. There are many issues concerning this scheme as the exact mechanism of CRL distribution and certificate validation is not described in detail. In addition the communication link to the CA might not always be available and thus, CRLs may become out of date. Crepeau and Davis [0] describe a certificate revocation scheme designed for wireless ad hoc networks. Nodes already have a certificate issued by a CA before entering the network. The scheme is actually a protocol based on accusations. Each node can accuse inconsistent nodes. Accusation information is broadcasted to the network. When the majority of the nodes in the network accuse a specific node, its certificate is revoked. Tables that include revoked certificates are delivered to each new node that enters the network. Although the accusations scheme is well designed there exist similar open issues as with [0], mostly concerning the dissemination and freshness of revocation information.

To the best of our knowledge, none of the so far proposed schemes uses OCSP for certificate validation [0]. OCSP is a simple protocol involving requests and responses that provide the current status of one or more certificates. A client can send a request to a server (usually called OCSP Responder) asking for information on the status of one or more certificates. This request can be digitally signed and contains a reference to the queried certificate(s) (certID). The server responds with a signed message that contains the status of the referenced certificate(s). The response message also contains time and date information. More specifically, these are: the time when the OCSP responder last updated the status information about the certificate(s) (lastUpdate), the time when the message was generated (producedAt) and optionally the time when the responder will update again the status information (nextUpdate). OCSP responses are always digitally signed either by the CA, a trusted or an authorized responder. The advantages of OCSP over CRLs or other offline methods for certificate revocation and validation are well known [0]. CRLs grow bigger with time and can become eventually very large. MANET nodes have limited network, storage, processing and memory resources. It is, therefore, inefficient to periodically broadcast to the network or download a revocation list of, probably, several megabytes, instead of propagating small OCSP responses. Consequently, with OCSP, large CRLs do not need to be distributed throughout the network and stored in every node. Instead, OCSP provides a client-server protocol which can be initiated on demand, every time a node needs to verify the validity of a certificate. Thus, there is no waste of valuable network resources and energy for distribution of large CRLs. Additionally, the deployment of a distributed OCSP ensures that there will be no single point of failure and, as a result, the service will always be available, whilst bottlenecks can be easily avoided. In a distributed OCSP, several nodes will be able to act as OCSP responders in order to provide valid and up-to-date certificate status information. Even if one of these nodes gets compromised, out of range, or becomes dormant, the service will still be available by other nodes. Moreover, each node that wishes to check the status of a certificate has the ability to contact the nearest OCSP responder instead of having to rely on a single centralised server. This way, bottlenecks are avoided and network resources are consumed efficiently.

Ad hoc wireless networks form highly dynamic environments. Nodes may enter or leave the network at random times, while other nodes may become dormant. Apparently, certain nodes are not always available. On the other hand, OCSP responses and the OCSP service in general must always be available to any node of the network. Consequently, we suggest that OCSP responses should be cached and kept in several nodes.
OCSP response caching was originally proposed from the PKIX working group of IETF [0] but was later abandoned. MANETs revised the original thoughts, and formulate OCSP caching a currently open issue. The advantages from the use of cached responses are not only limited to the increased availability. Cached OCSP responses are pre-signed by the OCSP responder. As a result, nodes that keep and transmit cached responses don’t have to sign them. Thus, not only they don’t need to spend additional computation power to digitally sign messages, but, also, they do not have to keep additional private keys for this purpose. Evidently, if such a node gets compromised, the overall OCSP service is not affected, as the keys used to sign responses are still safe. However, additional issues come about, as cached OCSP responses need to be fresh and up-to-date. Efficient mechanisms for updating cached responses should be provided in order to make sure that the information given in OCSP responses is accurate and current. The Open Mobile Alliance has already published a candidate version of an OCSP profile for mobile environments [0]. Its goal is to enable the use of OCSP in mobile devices with limited resources that use the Wireless Application Protocol (WAP). This profile sets requirements and constraints on OCSP in order to have smaller, simpler and more easily processed messages. Nonetheless, it is not within its scope to describe a scheme for OCSP deployment in a distributed environment.

4. THE PROPOSED ADOPT SCHEME

ADOPT is a scheme for a distributed deployment of OCSP in MANETs using cached OCSP responses. In this section we describe in detail the proposed, on-demand, scheme. In our framework we distinguish three different kinds of nodes: Server-nodes, Caching-nodes and Client-nodes. Server-nodes are nodes that announce the revocation status of the certificates. For instance, such nodes may be authorised OCSP responders directly connected to a CA (in CANs) or to a group of CAs (in AANs). They may be part of the MANET or they may be accessible through out of band means (e.g. GPRS, UMTS, etc., as in [0]). They issue OCSP responses which are then transmitted and stored at the Caching-nodes. Caching-nodes are nodes that cache and forward OCSP responses, acting as OCSP responders. First of all, they receive pre-issued and pre-signed responses from the Server-nodes. These responses are cached in them. When a Client-node requests the status of a specific certificate, Caching-nodes receive this request, and search their cache for a pre-issued response corresponding to that certificate. If a cached response is found it is sent back to the Client-node. Caching-nodes parse OCSP requests in order to identify the queried certificate and then locate a corresponding cached response. On the other hand, they don’t need to create or sign OCSP responses as these are pre-signed by the Server-nodes. The cached response is forwarded to the Client-node as is and there is no requirement for further processing. As a result, Caching-nodes do not need to use additional computational power to digitally sign responses. In order to receive cached responses, Caching-nodes need to maintain a path to Server-nodes. As we already mentioned, out of band means may be used, when Server-nodes reside within fixed networks. Alternatively, if Server-nodes are part of the MANET, the network itself could be used to transfer OCSP responses. The authenticity and integrity of the responses is not compromised as OCSP responses are digitally signed by the Server-nodes or by an authorised and trusted responder. In addition, Caching-nodes need to receive updated cached responses, either periodically or on demand. Updating cached OCSP responses in Caching-nodes is one of the most critical parts of our scheme. Each caching node should ideally provide the most recent cached response. However, due to the nature of wireless ad hoc networks this may not be always possible. For example, the link between a Caching-node and a Server-node might be unavailable for a large period of time. Thus, a mechanism should be established for updating cached OCSP responses. For instance, caching-nodes could communicate periodically with server-nodes in order to get the updated responses. Alternatively, updated responses could be retrieved on demand. We analyse how OCSP response caching operates in the next section.

Client-nodes have to be able to construct OCSP requests and also parse and verify OCSP responses. When a Client-node needs to check the status of a certificate, it constructs an OCSP request which is then sent to an OCSP responder. The location of OCSP Responders must be known in advance to the client-nodes. In traditional OCSP this is achieved by using the authorityInfoAccess extension [0] of the X.509 certificate. Nevertheless, the use of this extension may not be applicable to MANETs, due to the frequent changes of the network topology. For instance, the node specified in this extension may be out of range or in dormant state. Therefore, we propose an alternative, ad hoc oriented, method which discovers on demand available OCSP responses. This method is based on the use of Caching-nodes and relies on the discovery of OCSP responders on demand.
responder. This approach is very similar to the route discovery mechanism of the Dynamic Source Routing protocol (DSR) [0]. The OCSP request is broadcasted to the network by the Client-node. Intermediate nodes that receive the OCSP request and are not Caching-nodes, re-broadcast the request until a Caching-node is found. If this node has a cached response that corresponds to the specific request, it forwards it back to the Client-node, using the reverse path. Otherwise, it tries to acquire a new response from a Server-node. If a link to a Server-node is not available at that time, the request is again broadcasted until a Caching-node with the corresponding response or a connection to a Server-node is found. Error! Reference source not found. illustrates the sequence of actions that a Client-node and the Caching or Server-nodes perform in order to request and locate an OCSP response in a cache, respectively.

![Figure 2. ADOPT flowchart for locating an OCSP response](image)

Evidently, an OCSP request can circulate within the ad hoc network without ever getting a response back. This problem is quite similar to a route request that cannot be completed and to other routing problems, such as routing loops. In order to resolve this kind of issues we suggest a mechanism that is similar to the originally proposed for DSR [0]. More specifically, a Client-node can specify in the OCSP request the maximum number of hops (maxHops) over which the request may be propagated. Each intermediate host that receives the request reduces this number by one. If the maximum number of hops is reached without finding a corresponding OCSP response, either cached or from a Server-node, the request message is dropped. This issue is further discussed in the next section. The freshness of the OCSP response delivered to a Client-node may be of great importance to the latter. Some critical applications may require very fresh responses, while others may be more tolerant concerning the age of the cached responses. We suggest that a Client-node should be able to specify how fresh the expected response should be. Thus, an OCSP request message should include a field (updateTime) similar to the lastUpdate field in [0]. Only a Caching-node that has a cached response which was updated recently enough will be able to answer this specific request. Otherwise, the request should be re-broadcasted until a Caching-node which has a recent enough cached response is found. If no such response can be found in the specified number of hops, the request message is dropped.

Finally, it is worth mentioning that intermediate nodes that are Caching-nodes could update their caches according to OCSP responses that are forwarded through them. Intermediate Caching-nodes can eavesdrop on the messages that they forward to detect OCSP responses designated to other nodes. These responses can be added to their cache if they are not available or out of date.

### 5. CACHE MANAGEMENT ISSUES

As we mentioned previously, the proposed ADOPT scheme uses an on-demand search approach, where a Client-node asks Caching or Server-nodes for the status of a certificate when needed. This on-demand method should take into account several characteristics of autonomous networking. It should avoid communication overheads, choose Caching or Server-nodes taking into account energy thresholds, and distribute OCSP caches based on the dynamic network topology, so that certificate status information is constantly available to the network entities. Furthermore, access latency is a critical performance parameter that should be considered. A cache management scheme in an ad hoc network is characterised by the following policies:
- Node selection and cache placement policy, which determine the criteria that force nodes to act as Caching or Server-nodes.
- Cache update policy, which identifies the rules concerning writing a new OCSP response into the existing cache.
- Cache deletion discipline, for erasing OCSP responses from the cache.

It is worth mentioning that the cache management scheme should be fully distributed. Each ad-hoc node, based on local observation and measures, or in coordination with nearby nodes, should identify its role (Client, Caching or Server Node), and determine its cache updating policy.

5.1 Cache placement policy

The problem of cache placement in an ad hoc network has been studied in [0]. The objective is to find an optimal cache placement that minimizes both access latency and energy expenditure costs. In [0] a greedy algorithm was proposed, called POACH, which provides a sub-optimal solution to the cache placement problem. POACH considers stationary nodes, and assumes that the network contains an “always available” server that originally stores the information. Here, we need to address a different scenario with mobile, instead of stationary, nodes. Obviously, nodes that are energy-autonomous (i.e. plugged) can be considered as candidate Caching-nodes. Additionally, Caching-nodes candidates are those illustrating adequate resources (CPU, storage and memory).

5.2 Cache update and deletion policy

This paragraph discusses the rules for deciding when to store a new OCSP response entry to a cache, or when to delete an aged OCSP response entry from the cache. We assume that the initial OCSP status information is stored into the CAs or Authenticated Responders. In the ADOPT scheme, a Caching-node that senses OCSP responses with a thisUpdate value greater than the thisUpdate value of an identical, already cached, response, refreshes the corresponding cache entry. Additionally, if a Caching-node that keeps an entry illustrating an exceeded nextUpdate value, then the node maintains this entry, unless there is no storage space to cache new responses. The candidate states of a Caching-node are as follows:
- Greedy: each node caches every OCSP response that passes through it.
- Selective: each node caches k out of K OCSP responses that pass through it. An alternative approach is to cache an OCSP response after m appearances, where m is considered as the popularity index.
- No-Caching: a node does not perform any caching. Instead, the node may delete from its cache expired OCSP responses, i.e. cache entries illustrating an exceeded nextUpdate value.

6. REMARKS

In this paper we introduced a novel certificate validation scheme, applicable in MANETs. This scheme, called ADOPT, is based on the OCSP protocol and suggests the use of caches that contain certificate status information. ADOPT uses cached OCSP responses which are distributed to the ad hoc nodes. OCSP, as a lightweight protocol, prevents the flooding of extended revocation lists and thus, conserves the scarce bandwidth and avoids energy consumption that takes place during complex manipulation of revocation lists. Recent simulation results [0] show that on an ad hoc network with a threshold-based CA scheme, certification, authentication and routing messages use about 80% of the network’s capacity. Thus, certificate validation should not increase overheads. Until now, no other certificate revocation and validation approach for MANETs using OCSP and cached responses has been proposed. However, recently a new IETF draft on OCSP was submitted, describing a lightweight implementation of OSCP, but it is not specifically designed for MANETs [0].

ADOPT, furthermore, requires efficient cache distribution policies among the nodes, and cache management strategies that take into account mobility and dynamic network topology. The goal of our current research is to propose caching policies that enhance performance metrics, such as the delay in the location of the certificate’s status and the minimization of communication overheads. Moreover, simulation scenarios are under design to evaluate the framework. Evaluations will take into account communication

111
overheads, load balancing, energy consumption and OCSP cache location delays. We further plan to design and validate the ADOPT protocol, applicable for MANETs, which will encapsulate the OCSP and append ad hoc semantics, such as transaction ids.

REFERENCES

Gast, M., 2002. 802.11 Wireless Networks: The Definitive Guide. O'Reilly, USA.