SUPERMAN: Security Using Pre-Existing Routing for Mobile Ad hoc Networks

Darren Hurley-Smith, Jodie Wetherall, and Andrew Adekunle

Abstract—The flexibility and mobility of Mobile Ad hoc Networks (MANETs) have made them increasingly popular in a wide range of use cases. To protect these networks, security protocols have been developed to protect routing and application data. However, these protocols only protect routes or communication, not both. Both secure routing and communication security protocols must be implemented to provide full protection. The use of communication security protocols originally developed for wireline and WiFi networks can also place a heavy burden on the limited network resources of a MANET. To address these issues, a novel secure framework (SUPERMAN) is proposed. The framework is designed to allow existing network and routing protocols to perform their functions, whilst providing node authentication, access control, and communication security mechanisms. This paper presents a novel security framework for MANETs, SUPERMAN. Simulation results comparing SUPERMAN with IPsec, SAODV, and SOLSR are provided to demonstrate the proposed frameworks suitability for wireless communication security.

Index Terms—Access control, authentication, communication system security, mobile ad hoc network

1 Introduction

Mobile autonomous networked systems have seen increased usage by the military and commercial sectors for tasks deemed too monotonous or hazardous for humans. An example of an autonomous networked system is the Unmanned Aerial Vehicle (UAV). These can be small-scale, networked platforms. Quadricopter swarms are a noteworthy example of such UAVs. Networked UAVs have particularly demanding communication requirements, as data exchange is vital for the on-going operation of the network. UAV swarms require regular network control communication, resulting in frequent route changes due to their mobility. This topology generation service is offered by a variety of Mobile Ad hoc Network (MANET) routing protocols [1].

MANETs are dynamic, self-configuring, and infrastructure-less groups of mobile devices. They are usually created for a specific purpose. Each device within a MANET is known as a node and must take the role of a client and a router. Communication across the network is achieved by forwarding packets to a destination node; when a direct source-destination link is unavailable intermediate nodes are used as routers.

MANET communication is commonly wireless. Wireless communication can be trivially intercepted by any node in range of the transmitter. This can leave MANETs open to a range of attacks, such as the Sybil attack and route manipulation attacks that can compromise the integrity of the network [2].

Eavesdropped communication may equip attackers with the means to compromise the trustworthiness of a network. This is achieved by manipulating routing tables, injecting false route data or modifying routes. Man in the middle (MitM) attacks can be lauched by manipulating routing data to pass traffic through malicious nodes [3]. Secure routing protocols have been proposed to mitigate attacks against MANETs, but these do not extend protection to other data.

Autonomous systems require a significant amount of communication [4]. Problem solving algorithms, such as Distributed Task Allocation (DTA), are required to solve task planning problems without human intervention [4]. As a result, these algorithms are vulnerable to packet loss and false messages; partial data will lead to sub-optimal or failed task assignments.

This paper proposes a novel security protocol, Security Using Pre-Existing Routing for Mobile Ad hoc Networks (SUPERMAN). The protocol is designed to address node authentication, network access control, and secure communication for MANETs using existing routing protocols. SUPERMAN combines routing and communication security at the network layer. This is in contrast to existing approaches, which provide only routing or communication security, requiring multiple protocols to protect the network.

The remainder of this paper is organised as follows: Section 2 analyses the problem in the context of previously published work. Section 3 introduces SUPERMAN, providing a technical discussion of the protocol. Section 4 outlines the characteristics chosen for modelling, and the results of simulating SUPERMAN compared against selected secure

- *D. Hurley-Smith is with the School of Computer Science, University of Kent, Canterbury, Kent, UK. E-mail: d.hurleysmith@kent.ac.uk.*
- *J. Wetherall and A. Adekunle are with the Faculty of Engineering and Science, University of Greenwich, Chatham, Kent, ME4 4TB, UK. E-mail: {j.c.wetherall, a.a.adekunle}@greenwich.ac.uk.*

routing and data security protocols. Section 5 draws conclusions from the research findings.

2 Related Work and Problem Analysis

2.1 MANET Routing

MANETs rely on intermediate nodes to route messages between distant nodes. Lacking infrastructure to administrate the manner in which packets are routed to their destinations, MANET routing protocols instead make use of routing tables on every node in the network, containing either full or partial topology information. Reactive protocols, such as Ad hoc On-demand Distance Vector (AODV) [5], plan routes when messages need to be sent, polling nearby nodes in an attempt to find the shortest route to the destination node.

Optimised Link State Routing (OLSR) [6] takes a proactive approach, periodically flooding the network to generate routing table entries that persist until the next update. Both approaches are motion-tolerant and have been implemented in UAV MANETs [7], [8]. Motion-tolerance and cooperative communication characteristics make these protocols ideal for use in UAVs.

The basic versions of AODV and OLSR lack security mechanisms, allowing malicious nodes to interfere with the network in a variety of ways [9], [10], [11]. The key contributing factor to this problem is an inability to distinguish legitimate nodes from malicious nodes.

2.2 Security Threats

The ITU-T Rec., through X.805 [12], defines wireless end-to-end security in seven classifications, which are called dimensions. This system of classification allows for clear and convenient identification of security threats in a networks and potential solutions to those problems. The following security dimensions are identified:

- *Access control* is required to ensure that malicious nodes are kept out of the network.
- *Authentication* confirms the identity of communicating nodes.
- *Non-repudiation* prevents nodes from broadcasting false information about previous transmissions, mitigating replay and related attacks.
- *Confidentiality* prevents unauthorised nodes from deriving meaning from captured packet payloads.
- *Communication security* ensures that information only flows between source and destination without being diverted or intercepted.
- *Integrity* checking allows nodes to ensure packets received are in the same form they were sent, without modification or corruption.
- *Availability* ensures that network assets are accessible. Periodic checking of node status or reports from a node to its neighbours are a common means of checking the availability of a resource.
- *Privacy* prevents outside observers from deriving valuable information through passive observation.

Many MANET routing protocols assume trust between nodes, which can be a critical weakness in terms of security [9], as such an assumption may allow malicious nodes to interfere with routing mechanisms. Routing attacks can abuse the route discovery and topology generation mechanisms of routing protocols. An attacker could, for example, advertise routes with hop counts higher or lower than real routes [13]. This could be used to attract traffic to malicious nodes to the benefit of the attacker. Malicious activity may result in; the appropriation of data, sinking of packets and modification of packets. All such outcomes impair the networks ability to guarantee safe, private and reliable communication.

Unsecured pro-active routing protocols exhibit vulnerability to packet replay and manipulation attacks [14]. Due to a lack of source authentication, topology control messages can be broadcast frequently, which other nodes will treat as legitimate and use to update global topology information. Pro-active routing protocols detect neighbours through HELLO messages, allowing tunnelling attacks if a malicious intermediate node reports a route between two out of range nodes [15]. This results in the construction of a false topology, causing failure of the network when attempting to use incorrectly advertised routes.

Packet forwarding attacks may be used for Denial of Service (DoS). These attacks do not target the routing protocol, instead forcing the node in the network to act in a manner inconsistent with the routes established, generating an excess of traffic or sinking packets maliciously [16]. X.805 describes five key threats [12]:

- *Destruction:* Completely removing a packet from the network and deleting it locally, preventing it from reaching destination and destroying the packet
- *Corruption and modification:* Making a packet unreadable, or changing the content of the packet
- *Theft, loss or removal:* Stealing packets from the network for further analysis, causing packets to drop or removing them from the network
- *Disclosure:* Revealing network information by rebroadcasting received packets to untrusted nodes
- *Interruption of services:* Disruption of any service the network offers, resulting in loss of service or unacceptable completion time.

Yang et al. notes that malicious attacks may easily disrupt MANET operations [9]. An attacker can take advantage of MANETs that assume, but not enforce, trust between nodes. Closing the network by forcing legitimate nodes to authenticate can resolve the assumption of trust, by ensuring that only legitimate nodes can become members of the network [17]. In a closed network, participation is restricted to authorised nodes, and communication is encrypted to prevent third-party comprehension of the contents of network communication. Authentication is required to allow new nodes to join and be seen as legitimate by existing network members [18].

The amount of time an individual UAV node may remain operational is limited by its battery life (energy), which may be shorter than the expected duration of the network's deployment [19]. A replacement may be required if a node runs out of energy. Malicious nodes may masquerade as legitimate nodes, attempting to gain trusted status in the network by posing as a recently departed or newly arriving node [10].

Subversion of the replacement procedure may be mitigated by requiring the successful authentication of a node with the network. This approach would authenticate nodes using cerfitificates provided at initialisation by a trusted authority. This authority is central to the network security scheme, but need not be present in the field [18].

2.3 MANET Routing Security

To tackle the problems that assumed legitimacy can cause, secure MANET routing protocols have been proposed. Secure Ad hoc On-demand Distance Vector (SAODV) and Secure Optimised Link State Routing (SOLSR) are secure implementations of AODV and OLSR respectively. SAODV secures the routing mechanism by including random numbers in Route Request packets (RREQs) [20]. If a routing packet arrives that re-uses an old packet number, that packet is invalid. Nodes observed sending re-played packets may be flagged as malicious. SAODV requires that at least two Secure RREQs (SRREQs) arrive at the destination node by different routes with identical random numbers to identify the source node.

SOLSR aims to allow detection of wormhole attacks during its neighbour detection phase [14]. Nodes should be authenticated prior to establishing neighbour status to prevent malicious nodes from asserting themselves as neighbours. Verification of a source node's identity must be performed. Each node is assumed to have an asymetric key pair, managed by a coalition of nodes using threshold cryptography. A distributed Certificate Authority (CA) system is required to manage this process if certificates are replaced in the field.

Each packet sent by SOLSR is digitally signed using a shared secret. If an incoming packet's signature is unreadable, the packet is discarded as being unauthentic. This is a point-to-point process and does not provide source authentication. To prevent replay attacks, SOLSR uses time-stamped packets. If a time-stamp is seen twice by a legitimate node, the packet will be discarded [14], [15].

Due to the lower hardware specifications and resource restrictions on UAV-based MANETs, the use of individual nodes as authentication servers is not ideal. If a node is compromised, it may deny legitimate nodes access to the network. If a compromised node has authentication privileges, it may authenticate additional malicious nodes and possibly blacklist legitimate nodes.

Centralised approaches rely on a single node taking control of key management and trust systems [21]. This puts additional strain on that node due to repeated call for authentication from other nodes. It also presents a single vector of attack against network security mechanisms; if the central authority is compromised, the entire network may also be compromised.

The primary objective of SAODV and SOLSR is to prevent malicious nodes from gaining control of the topology generation mechanisms of the routing protocol, and to protect against black hole and wormhole attacks. Routing is secured and malicious node detection is employed in both cases.

2.4 Secure Communication

Securing routes is only one aspect of a full security solution. X.805 highlights many security threats including identity, data manipulation, corruption and theft [12]. There are three requirements to securing communication; authentication, confidentiality and integrity.

X.509 sets the standard for certificate-based approaches to security [22]. Certificates provide a suite of data that can be used to represent the identity of a given node, and its relationship with a trusted authority.

Internet Protocol Security (IPsec) is a secure communication framework extending confidentiality, integrity and authentication services. It is comprised of three key protocols: Authentication Headers (AH), Encapsulating Security Payloads (ESP) and Security Associations (SA) [23].

AH provides connectionless integrity and source authentication services. It does not provide route authentication, as IPsec does not account for the route taken to destination.

ESP provides confidentiality, integrity and authentication services. ESP does not extend protection to the IP header of a packet. This is useful if the IP header must be swapped, for example during multi-hop operations. ESP encapsulates an AH packet which provides source authentication, once IP headers have been removed.

SA is a collection of security features used by AH and ESP. All nodes in the network share an SA to provide a common basis for encryption, authentication and integrity checking.

Ghosh et al. discuss the modification of a certificate-based application of IPsec supporting dynamic key-generation for MANETs [24]. They state that their approach secures mobility, application and management traffic. Increased latency and bandwidth use were observed as a cost to their approach.

MANIPsec provides a model entirely focused on MANET security using IPsec [25]. They propose a modified IPsec focused on lightweight security, while retaining authentication and confidentiality features. Their proposal seeks to extend security to all control traffic, including routing traffic. A key observation in their work is that network control traffic, such as routing operations, demand significant resources when compared with most application driven traffic.

The approaches discussed to this point have used certificates for providing security services. From the certificate, symmetric keys can be derived for secure communication, allowing confidentiality, integrity and authentication services to be extended to any packets that require it.

The Diffie-Hellman key generation algorithm is an example of a means of generating symmetric keys without the need to explicitly communicate any sensitive key information [26]. Nodes exchange locally generated data using globally known primes and local secret data. The resulting variable (referred to as a key-share) is then communicated by both nodes, facilitating the calculation of a symmetric key that is identical at both ends, without the need to communicate sensitive data at any point. This allows the discreet and secure establishment of node-to-node confidentiality between specific node pairs [27].

Key derivation functions (KDF) allow the generation of multiple keys from a combination of a source key and metadata [28]. This is useful when a single shared secret needs to be used in multiple different contexts.

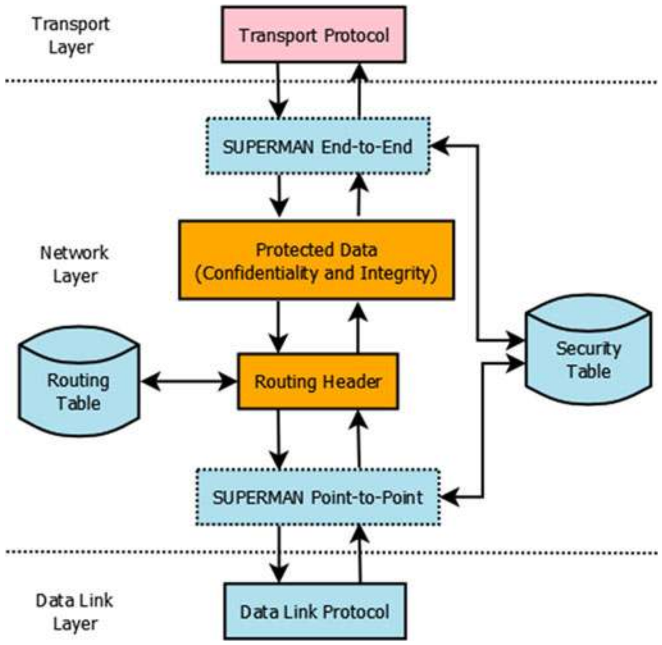


Fig. 1. Diagram illustrating the SUPERMAN confidentiality, integrity, and authentication services for data packets.

2.5 Summary

Access control has been identified as a security dimension that might address the issue of implicit trust within a MANET. By closing the network to outsiders, the issue of assumed co-operation is circumvented. Closing the network requires a means of allowing nodes to join and leave the closed network.

Authentication provides a means by which a node may be identified as trustworthy. By using a certificate to confirm that they share a trusted authority, two nodes may authenticate one-another based on their shared Trusted Authority (TA).

Wormhole and Sybil attacks have been analysed and addressed by protocols such as SAODV and SOLSR. The protection that these protocols offer is aimed at the protection of network routing services. These protocols do not protect data sent over the secured routes.

IPsec and the proposed MANET modifications (MANIPsec) protect data sent over networks. They do not protect the route, leaving the network vulnerable to attacks on the topology (e.g., MitM).

SUPERMAN, the protocol proposed in this paper, addresses the problem of unified MANET communication security. It implements a Virtual Closed Network [18] architecture to protect both network and application data. This is in contrast with the approaches proposed in previous work, which focus on protecting specific communication-based services.

3 The SUPERMAN Framework

SUPERMAN is a framework that operates at the network layer (layer 3) of the OSI model. It is designed to provide a fully secured communication framework for MANETs, without requiring modification of the routing protocol. Fig. 1 shows the flow of data from transport, through the network layer (including SUPERMAN) to the data link layer. The dashed boxes represent elements of SUPERMAN that process packets and provide confidentiality and integrity. SUPERMAN also provides node authentication.

Octets	0	1	2	3	4
0	Type	Timestamp		Protocol Identifier	

Fig. 2. SUPERMAN packet header (SH) structure.

3.1 Terminology

Key terms used when describing SUPERMAN include:

- Trusted Authority
  - A static node responsible for node initialisation and provision of certificates; it is a prerequisite to SUPERMAN.
- Certificate (*CKp*)
  - Required per node and shared with other nodes to join the network
- Public Diffie-Hellman Key Share (*DKSp*)
  - A public value communicated between nodes
- Private Diffie-Hellman Key Share (*DKSpriv*)
  - A private value, held by all nodes in the network and never communicated. Used as the shared secret for Diffie-Hellman key exchange
- Identifier (I)
  - A per node unique identifier, such as an IP address in an IP-based network
- Encrypted Payload (EP)
  - Payload data encrypted using an encryption scheme such as AEAD
- Tag (T)
  - A tag, appended as a footer to all SUPERMAN packets to provide point-to-point integrity services
- Symmetric key (SK)
  - *SKe(s, d)* is a security key used for encryption of end-to-end communication between a source and destination node, derived locally via KDF from the product of the *DKSp* and *DKSpriv*
  - *SKp(s, d)* shared by two nodes; used to authenticate traffic as it moves along the network, derived locally via KDF from the product of the *DKSp* and *DKSpriv*
- Key Derivation Function (*KDF(SK, func)*)
  - A function used to provide multiple different keys from a common private source
- Symmetric broadcast key (*SKb*), shared with newcomer nodes by the node that allows them to join the network, generated by the first node to initialise the network. Differentiated into two application specific keys by a network-wide KDF stored locally on each node
  - Symmetric end-to-end broadcast key (*SKbe*)
  - Symmetric point-to-point broadcast key (*Skbp*)

3.2 SUPERMAN Framework Overview

Every SUPERMAN packet shares a common SUPERMAN packet header (SH), shown in Fig. 2. The data contained in the header can be broken down as follows:

TABLE 1
SUPERMAN Security Table

Node ID	SKe	SKp	DKSp
I(X)	SKe(A,X)	SKp(A,X)	DKSp(X)
I(Y)	SKe(A,Y)	SKp(A,Y)	DKSp(Y)
I(*)	SKbe	SKbp	SKb

- Packet Type denotes the function of the packet
- Timestamps provide uniqueness, allowing detection of replayed packets and providing a basis for non-repudiation of previously sent packets
- The protocol identifier indicates the layer 4 type of the encapsulated data. This would be the IP protocol number in an IP based network.

3.2.1 *Key Management*

SUPERMAN relies on the dynamic generation of keys to provide secure communication.

The Diffie-Hellman key-exchange algorithm provides a means of generating symmetric keys dynamically and is used to generate the SK keys. *SKb* keys can simply be generated by means of random number generation or an equivalent secure key generation service.

3.2.2 *Secure Node-to-Node Keys*

SKe keys are used to secure end-to-end communication with other nodes, with one *SKe* key generated per node, for every other node also authenticated with the network. *SKp* keys are used for point-to-point security and generated in the same manner as *SKe* keys.

It is important that *SKe* and *SKp* keys are different, as the network needs to secure both the content of a packet and the route taken.

A KDF can be used to generate these two keys in conjunction with the result of the Diffie-Hellman algorithm, requiring a *DKSp*/*DKSpriv* pair, to minimise the cost of security on the network and reduce the key re-use and, in turn the lifetime of each key.

These keys are generated when nodes receive DKSp's from other SUPERMAN nodes.

3.2.3 *Secure Point-to-Point Footers*

Secure footers are appended to all communication packets sent between SUPERMAN nodes. *SKbp* and *SKp(x)* keys are used in broadcast and unicast integrity service provision respectively.

An example tag generation algorithm is the Hashed-Message Authentication Code (HMAC) which provides integrity and authenticity services to a packet. A digest of the packet is generated, encrypted with the appropriate key (SKbp or SKp(x)), and appended to the packet. This tag is removed, checked and regenerated at each hop.

3.2.4 *Secure Broadcast Keys*

At initialisation of the network, the first node to be contacted about joining the network will generate a symmetric network key (*SKb*). This key is sent to all nodes that authenticate with the network. This key provides the basis for all broadcast communication security in a SUPERMAN network.

TABLE 2
The SUPERMAN Packet Sizes

ID	Type ID	Packet Type	Size(Bytes)
01	DReq	Discovery Request	SH+DKSp(s)
02	CReq	Certificate Request	SH+DKSp(s)
03	CEx	Certificate Exchange	SH+CKp+T
04	CExB	Certificate Exchange with Broadcast Key	SH+T
05	DSKp Req	DSKp Request	SH+T
06	DSKp Rep	DSKp Reply	SH+DKSp(s)+T
07	SKI	SK Invalidation	SH+I+T
08	BEx	Broadcast Key Exchange	SH+SKb+T
09	DP	Data Packet	SH+EP+T

The *SKb* is processed by the function *KDF(SKb, type)* into two broadcast keys (*SKbe* and *SKbp*).

A node will use these keys to encrypt and sign packets sent to the broadcast address of the network. This key is used for broadcast and multicast communication, such as MANET route updates. It is not used for communication between individual end-points.

Upon deriving a broadcast key that will be tied to the network, the receiving node will add the resulting keys to its security table. *SKbe* keys are used to provide confidentiality to end-to-end broadcast communication. *SKbp* keys are used to generate tags, generated using an algorithm such as HMAC, appended as a footer to SUPERMAN protected packets, providing broadcast packet integrity.

Broadcast keys are generated by the first node to participate in a network joining process as the authenticator (the responding partner). They are then shared as the final stage of all network joining processes that result in a new node becoming a part of that network.

3.2.5 *Storage*

SUPERMAN stores keys in each node's security table. The security table contains the security credentials of nodes with which the node has previously directly communicated, as shown in Table 1. This table has *n* entries, where *n* is the number of nodes that the node in question has directly communicated with. Table 1 shows an example of a security table belonging to node A. It has exchanged credentials with two other nodes, X and Y.

The shared symmetric broadcast key (*SKb*) has two derived forms, the *SKbe* and *SKbp*. These are stored in the local security table as a separate broadcast address, denoted by I(*). These keys are not associated with any one network, but represent security credentials held by the whole network. A node's ID would be its address.

3.2.6 *SUPERMAN Packet Types*

Table 2 shows the packet types used by SUPERMAN, including their default packet sizes before the addition of any network layer headers such as IP or data link layer headers such as 802.11.

3.3 Network Access Control and Node Authentication

A certificate-based method, such as X.509, is used to control access to the network [22]. Every legitimate node in the

Octets	0	1	2	3	4
0	SH				
5	CKp				
1028		HMAC Tag			
1044					

Fig. 3. Example of a SUPERMAN certificate exchange packet.

network is provided with a certificate by the associated Trusted Authority.

This allows nodes from different TAs to communicate securely within the same network, establishing a hierarchical structure among TAs. This allows multiple controllers, each with their own TA, to share MANET resources if they share a hierarchy.

3.3.1 Certificates

Fig. 3 shows an example of the format for a SUPERMAN Certificate Exchange packet. This example demonstrates an implementation using a 1,024-byte certificate and 20-byte tag.

3.3.2 Certificate Exchange

A sequence diagram outlining the certificate exchange process is shown in Fig. 4.

0. Each node is provided with a certificate from a TA, in order for it to join SUPERMAN networks.
1. The joining node (*A*) seeks to join a network by periodically broadcasting Discovery Request (*DReq*) packets containing its *DKSp*. This continues until it receives a Certificate Request (*CReq*) from a networkable node (*B*).
2. Having received a *DReq* from node A, node B sends a *CReq* packet containing its *DKSp* to A. Both nodes perform Diffie-Hellman using the shared *DKSps* they now hold, to generate *SKe* and *SKp* keys which are used to encrypt and provide integrity to the rest of the access control process
3. Upon receiving a *CReq* from B:
   a. *A* sends its certificate in a Certificate Exchange (*CEx*) packet to *B*.
   b. *B* checks the integrity and authenticity of the *CEx* packet, using the shared SKp.
   c. *B* checks the certificate's authenticity against the TA hierarchy of its own certificate and the certificate contains the DKSp shared previously by A. If the certificate is deemed authentic *A* is added to *B's* security table. If the certificate fails this check, the *DKSp*, *SKe* and *SKp* credentials generated for node A by B are dropped and B and the process ends.
4. *B* responds to *A's CEx* with its own Certificate Exchange with Broadcast Key (CExB). *A* repeats steps *a* to *d* in 2. The CExB also provides A with the *SKb*, from which it derives *SKbe* and *SKbp* for broadcast communication, using the KDF. *B* and *A* both invalidate any prior security associations they have with each other when receiving *DReq or CReq* packets with new information. This involves purging all previous information from their local security table entries for each other.

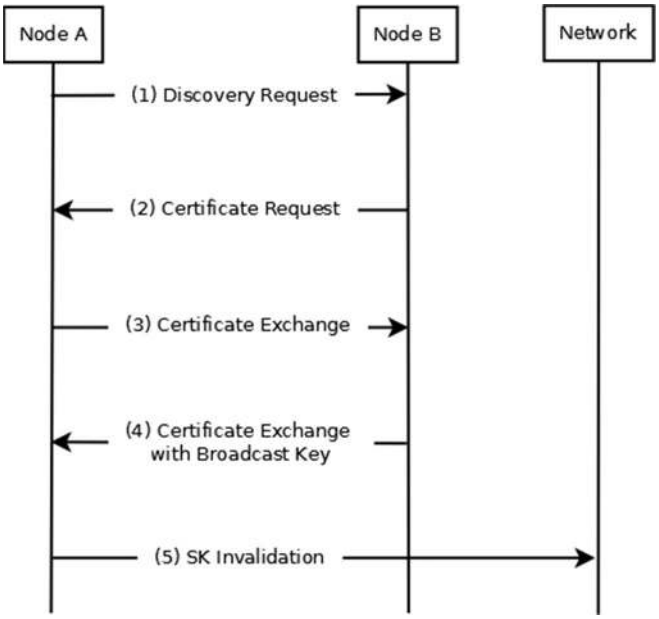


Fig. 4. Sequence diagram to demonstrate the certificate exchange process.

   a. If B has not yet authenticated any other nodes, it will generate an *SKb*, prior to sending it to the joining node (A in this case), otherwise it will send the current *SKb* to the joining node
5. If *A* has a broadcast key, it transmits a Broadcast Key Exchange (*BEx*) packet containing the new key, secured with the original key before committing the new key to its security table.
6. *B* broadcasts an SK Invalidation (*SKI)* packet, invalidating any previous credentials *A* may have had with nodes within the network. This prevents the accumulation of expired security data on nodes that may be isolated from a previous invalidation event.

After authentication has been completed, both nodes will possess the following data:

- Each other's certificate
- The network share (*SKb*) to allow the derivation of broadcast keys via the function *KDF(SKb, type)* to allow secure broadcast communication
- Each other's Diffie-Hellman Key Share (*DKSp*), resulting in the calculation of *SK*, which is used in the function *KDF(SK, meta-data)* meta-data being a variable indicating whether the key is required for encryption or other security operations:
  - ○ *SKe* and *SKp* for end-to-end and point-to-point secure communication

Following this method, it is possible for a node to build up a collection of symmetric keys representing its links with every other node it has exchanged security details with.

3.3.3 DKSp Referral Mechanism

Fig. 5 shows the four distinct conditions under which DKSp referral may take place.

All nodes that have authenticated with the network will have a valid broadcast key, and so can perform routing operations, even with nodes that they do not share *SKs* with. However, situations may arise where nodes that may

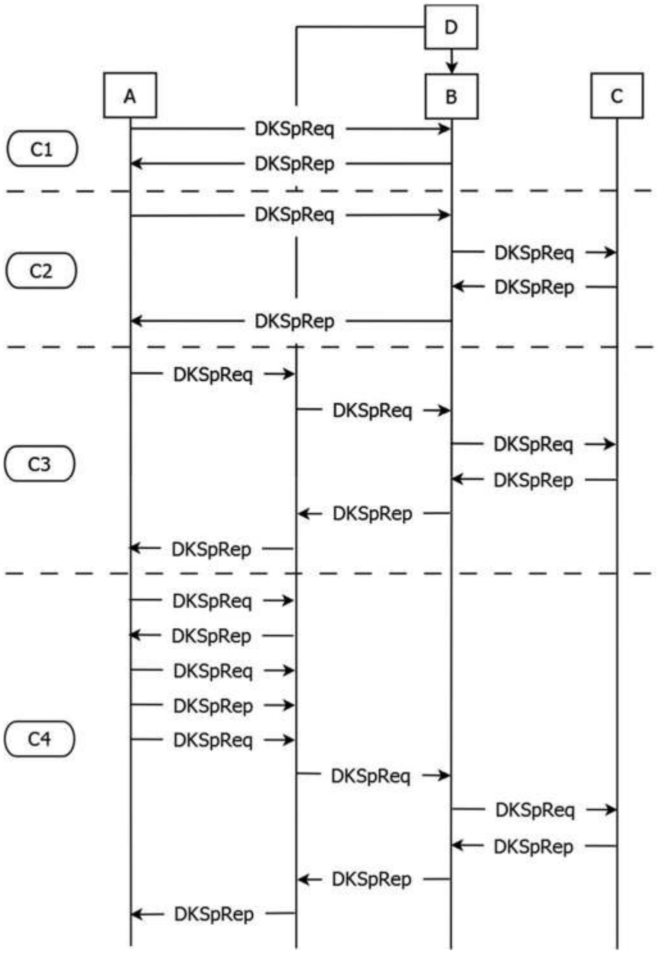


Fig. 5. Sequence diagram to show DKSp referral in four use cases.

need to communicate with individual nodes, requiring the exchange of DKSp data.

The DKSp referral mechanism is presented as a means of exchanging DKSp's in an efficient manner. If the nodes involved are separated by intermediate nodes, the intermediaries may respond on behalf of the destination node, if they hold the appropriate DKSp in their security table.

DKSp Request (*DKSpReq*) packets are used to request a nodes DKSp. DKSp Response (*DKSpRep*) packets are used to respond to *DKSpReq* packets. These packets contain the *DKSp* of the destination node. They may be communicated by the target destination or an intermediate node in possession of the sought DKSp. The purpose of this process is to reduce the overhead incurred when forming secure links between nodes that are both already within the network.

Both types of packet have a tag generated using the network *SKbp* key appended as a secure footer to provide integrity checking services.

In three of the four scenarios shown in Fig. 5, *A*, *B* and *C* have joined the network by authenticating with *D*. The fourth scenario assumes that *A* has roamed from its initial position, and does not possess the DKSp of any of the local nodes. Node A is requesting C's DKSp. All nodes are assumed to have authenticated with the network.

C1. *A* needs to communicate with *B* and A's adjacent to *B*. *A* lacks *B's DKSp*.
  a. A *DKSpReq* is sent to *B* by *A*.
  b. *B* responds with a *DSKpRep* containing its DKSp. *A* adds *B's DKSp* to its security table.

C2. *A* needs to communicate with *C* and requires an intermediate node *B* to relay communication. *A* and *B* do not know *C*, but know each other.
  a. *A* sends a *DSKpReq* to *C* via *B*.
  b. *C* is not known to *B*. *B* forwards the *DSKpReq* to *C*.
  c. *C* replies to *B* with a *DSKpRep*.
  d. *B* adds *C's DKSp* to its security table the forwards it on to *A*.
  e. *A* receives *B's* forwarded *DSKpRep*, then adds *C's* security details to its security table.

C3. *A* needs to communicate with *C* and requires a route through *D* and *B* to reach *C*. *A* knows *D* but not *B* or *C*.
  a. If nodes *D* or *B* hold the *DKSp* for *C*, they may respond on *C's* behalf and pass *C's* details on to *A* without ever contacting *C*. The dotted lines in Fig. 5 represent optional communication that will not occur if a previous node holds *C's* security details.

C4. *A* needs to communicate with *C* but does not know *D or B*.
  a. To send messages securely, *A* needs to know *D* and *C*. *A* will send a *DSKpReq* to *D*. *D* and *A* will associate with each other as per case 1.
  b. When associated with *D*, *A* will send a *DSKpReq* addressed to *C*. *D* will relay the *DSKpReq* unless it has *C* in its security table.
  c. If *D* does not have *C* in its security table, the procedure outlined in case 3 will be followed.

The above process reduces network communication by allowing nodes to respond on behalf of other authenticated nodes if possible. SUPERMAN provides a closed network of trusted nodes, allowing the trusted exchange of security credentials by third parties that are also members of that closed-network. This is an on-demand process, in which security credentials are not communicated to nodes which never need to directly communicate.

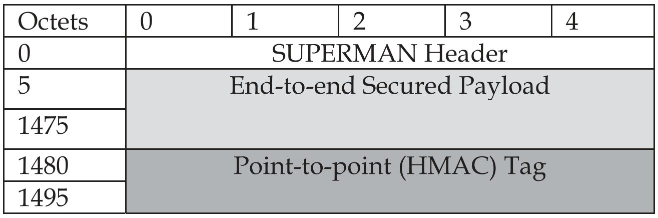
3.4 Communication Security

Once a node has joined the network, it may engage in secure communication with other nodes. Secure communication under SUPERMAN provides two types of security; end-to-end and point-to-point.

3.4.1 End-to-end Communication

End-to-end security provides security services between source and destination nodes by using their shared *SKe*.

Confidentiality and integrity are provided using an appropriate cryptographic algorithm, which is used to generate an encrypted payload (EP). Authenticated Encryption with Associated Data (AEAD) is an example of such an algorithm [29]. AEAD and related cryptographic algorithms provide confidentiality, authenticity and integrity services. The end-to-end element of a SUPERMAN packet is not modified at any point along a route. Its purpose is to provide confidentiality and source authentication services.



Octets	0	1	2	3	4
0	SUPERMAN Header				
5	End-to-end Secured Payload				
1475					
1480	Point-to-point (HMAC) Tag				
1495					

Fig. 6. Example of a SUPERMAN packet using AEAD and HMAC.

3.4.2 Point-to-point Communication

When protected, data is propagated over multiple hops, it is authenticated at each hop. This is achieved using a hashing algorithm, such as HMAC. This is applied to the entire packet to provide point-to-point integrity. A tag is generated using the shared *SKp* of the transmitting node and next hop, which is unique to the direct link in question. The tag is replaced at each intermediate hop, until the destination node is reached. Thus, the authenticity of a route is maintained, as each node on the route must prove their authenticity to the next hop. This tag can also be used for integrity checking.

Fig. 6. shows the structure of a SUPERMAN packet with end-to-end and point-to-point security services. The tag is assumed to be 20 bytes in length, but may be truncated depending on the scenario. A maximum size payload is used in this example.

3.4.3 Broadcast

When a node initiates a broadcast, it uses the broadcast address for the network. Instead of using a *SKe* or *SKp*, which would only function between two nodes, *SKbe* and *SKbp* are used. The packet is secured using the end-to-end and point-to-point methods previously described.

MANET routing protocols require broadcast capabilities. Both OLSR and AODV require broadcast communication for routes discovery. SUPERMAN provides broadcast communication security services to allow it to service the specific needs of MANET routing protocols.

3.5 Summary

SUPERMAN addresses the eight security dimensions detailed by X.805 by providing a closed-MANET, with end-to-end and point-to-point security features. The eight security dimensions are addressed as follows:

- *Access control* is provided by SUPERMAN's network joining method
- *Authentication* is provided by certificates, which allow the relationship between the node and TA to be confirmed
- *Non-repudiation* is provided by timestamps in each SUPERMAN packet header
- *Confidentiality* is provided end-to-end by payload encryption using AEAD
- *Communication* security is maintained by encrypting and performing source authentication end-to-end, and checking authenticity and integrity at each hop
- *Integrity* checking is provided by using a tag for packet integrity
- *Availability* is maintained using each nodes security table, which stores valid authentication credentials. This is combined with the DSKpReq/DSKpRep referral mechanisms to increase availability.
- *Privacy* is provided by end-to-end encryption, with keys that are specific to the link between two nodes or a node and the network.

The next section will present and analyse the results of modelling performed to determine the characteristics of SUPERMAN and its cost in terms of bandwidth, service time and throughput.

TABLE 3
MATLAB Simulation Parameters

Number of Nodes:	10-100
Routing Algorithm:	Dijkstrka [30] (shortest path)
Number of Iterations:	100
Simulation Area:	100 m x 100 m
Communication Range:	100 m
Max Hop Count:	5
Random Seed:	11
Pseudo-random Number Generation Algorithm:	Mersenne Twister [31]
Key Share Size	128 and 256 bytes
Certificate Size	1,013 and 1,275 bytes

4 Methodology and Results

To analyse SUPERMAN, the following key areas were investigated:

- Comparison of security dimension coverage
- Number of communication events required to secure communications between all nodes
- Number of bytes required to secure communications between all nodes
- Overhead of securing communication required for route generation
- Overhead of securing communication required by Consensus Based Bundle Algorithm (CBBA) and Cluster Form CBBA (CF-CBBA)

The eight key security dimensions, outlined in X.805 are evaluated by comparison between SUPERMAN, SAODV, SOLSR, and IPsec/MANIPsec. These are compared in terms of the services provided. This is important because it contextualizes the comparisons of the respective security and communication costs.

These costs represent the additional data or packets (based on the number of communication events) required to provide the security services, referred to from this point as the security overhead.

Overheads are calculated for the network layer of the OSI model. The Datalink and Physical layers of the network stack are not considered as this paper focuses on the network layer (OSI layer 3) specifically.

4.1 Simulation Parameters

All simulation is performed using MATLAB. Table 3 shows the parameters for the simulation environment.

It is assumed that all packets arrive intact without bit-error or loss, and that nodes are stationary during the initialisation and association phases.

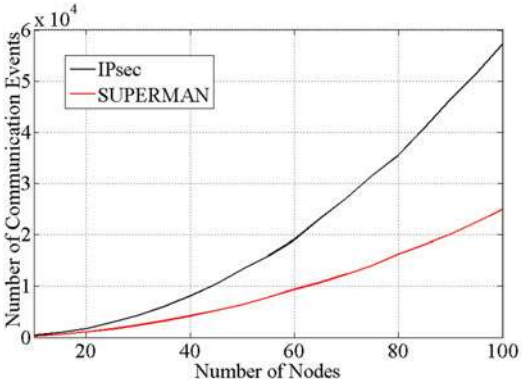


Fig. 7. Graph comparing the number of communication events to secure connections between all nodes under SUPERMAN and IPsec.

4.2 Initialisation cost of SUPERMAN and IPsec

4.2.1 Method

Comparison of the control overhead required by SUPERMAN and IPsec to initialise a secure network environment allows for the identification of the initialisation costs associated with each approach. These costs may occur throughout the lifetime of the network, but are incurred only when nodes join the network. Two metrics are considered:

- The number of communication events
- The number of bytes transmitted

Both metrics are measured until all nodes in a static set have joined the network.

4.2.2 Results

Fig. 7 compares the number of communication events required to secure all end-to-end connections in a MANET, using SUPERMAN or IPsec. All SUPERMAN nodes have authenticated with the network at this stage, and all IPsec nodes have performed IKE.

The number of communication events represents the total number of messages sent, regardless of packet size. This metric allows one to compare the verbosity of protocols, and comparisons regarding scalability may be made. It also provides data regarding the length of routes, as each relay of a given message will increment the communication event count.

MANETs of 15 nodes require 1,407 events for SUPERMAN and 1,609 for IPsec to form security associations between all nodes. SUPERMAN requires 87 percent of the communication events needed by IPsec, showing immediate gains in security association overhead.

SUPERMAN quickly demonstrates the effectiveness of its referral mechanism, showing itself to be far more scalable than IPsec. A clear trend is shown, in which SUPERMAN more slowly increases in security overhead compared to IPsec. In 100 node simulations, SUPERMAN requires only 42.1 percent of the communication events needed compared with IPsec. This is the result of SUPERMAN node being able to authenticate each other, without reference to a central trusted authority in the field. Pre-initialisation of nodes by a TA implies trusted status when unable to contact the TA directly.

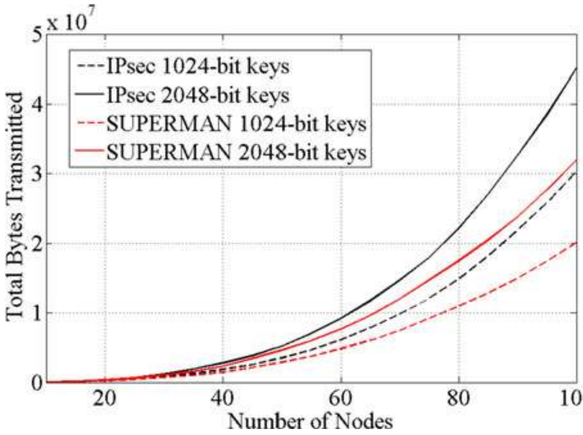


Fig. 8. Graph comparing the number of bytes required to secure connections between all nodes under SUPERMAN and IPsec.

Fig. 8 compares the number of bytes required to secure connections between all nodes in a MANET, using SUPERMAN and IPsec.

SUPERMAN consistently outperforms IPsec in terms of the number of bytes required to secure all nodes in a MANET. For smaller networks, this difference is less pronounced, but for 100 node MANETs, SUPERMAN requires 20.3 megabytes compared to IPsec's requirement of 30.5 megabytes, when using 1,024-bit symmetric keys. SUPERMAN requires only 60 percent of the data required by IPsec to achieve the same outcome, secure communications between all nodes. This trend continues for 2,048-bit keys. SUPERMAN benefits from the cooperative nature of MANETs in both experiments, whereas IPsec requires each node to check in with a coordinator during the authentication process. By allowing nodes to vouch for other nodes that they have already formed secure links with, SUPERMAN reduces the length of routes by not requiring DKSpReq and DKSpRep packets to propagate the full length of the route between source and destination.

4.3 Data Communication Cost of SUPERMAN and IPsec

4.3.1 Method

The MATLAB simulation allows the size of the added communication overhead (number of additional bytes) to be determined. Two scenarios were simulated supported by the parameters outlined in previously in Table 3:

- CBBA task allocation involving 18 nodes
- CF-CBBA task allocation involving 6 clusters of 3 nodes (18 nodes in total)
- Both DTA processes have a task list of between 1 and 50 tasks all of which must be assigned
- In both scenarios is it assumed that all nodes may communicate with each other, over routes that are no longer than the maximum hop count defined for the simulation

4.3.2 Results

Fig. 9 compares the security overhead of SUPERMAN and IPSEC performing CBBA.

The lines in this graph incorporate the noise inherent in the CBBA algorithm. The combination of network size

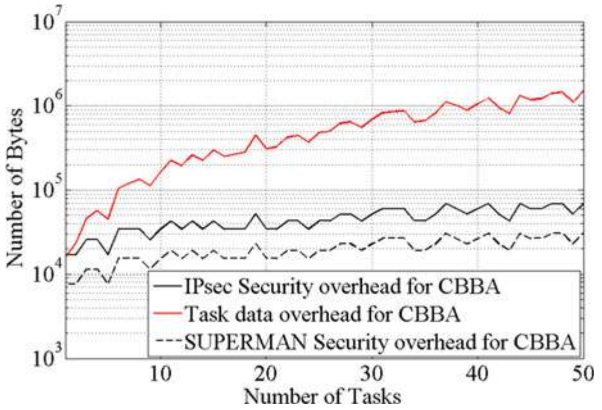


Fig. 9. Chart to compare the number of additional bytes required for the security overhead of IPsec and SUPERMAN when performing CBBA.

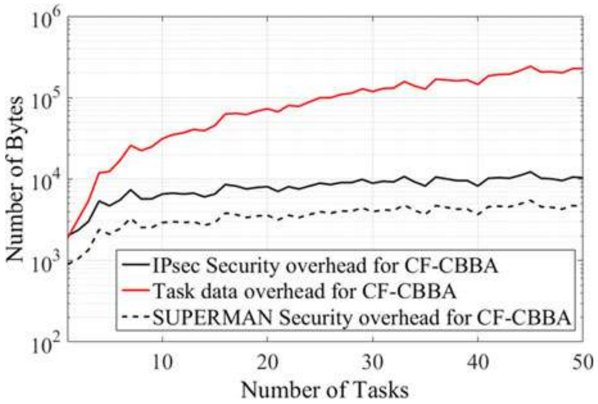


Fig. 10. Chart to compare the number of additional bytes required for the security overhead of IPsec and SUPERMAN when performing CF-CBBA.

and number of tasks can result in the number of CBBA runs required to achieve consensus varying greatly. The value is usually constrained to between 2 and 5 runs of CBBA to reach a solution, and the irregularities in Fig. 9 are a result of higher or lower numbers of runs being required by a given node/task combination. It is not trivial to calculate the number of CBBA rounds, as the number required depends on the size of the network, positions of individual nodes rleatives to tasks and the number of tasks.

The number of bytes required by CBBA is shown to grow rapidly with the size of the CBBA problem domain (the number of nodes and tasks involved). As more nodes are added to the network, the complexity of CBBA communication increases at a cubic rate. IPsec and SUPERMAN add additional security data, requiring that all outbound packets are encapsulated with appropriate headers and tags.

IPsec's overhead is larger than the size (17.1 KB compared with a payload of 15.9 KB). SUPERMAN requires only 7.6 KB of additional data, but this is still 47.7 percent of the size of the payload being protected. This is a result of having assumed the worst case for tag size (20 bytes). Both IPsec and SUPERMAN security overheads reduce in relative size for larger problem domains. For 50 task problems, SUPERMAN requires 30.6 KB and IPsec requires 68.5 KB, to protect a payload of 1.5 MB. SUPERMAN adds approximately 2 percent more data to provide security for this size of problem domain, with IPsec adding 4.5 percent. SUPERMAN requires half of the overhead generated by IPsec to provide the same level of protection to the task allocation process.

SUPERMAN does not require the two IP headers that IPsec needs. As SUPERMAN is integrated at the network layer, it does not re-encapsulate the packet. IPsec encapsulates a payload packet in an IPsec security layer, both of which must have IP headers. By avoiding this redundancy and stripping settings data from its header, SUPERMAN reduces its security overhead by a minimum of 32 bytes per packet.

(1) provides a mathematical expression for the security overhead of CBBA, under a given security framework.

$$x = \frac{(f(c) * (n(n-1))) * (h+t)}{p} \tag{1}$$

The function of c represents the number of rounds required by a given consensus based distributed task allocation algorithm. The number of nodes is represented by n. The header and tag size (are represented by h and t respectively. It is assumed that the payload of a packet will not exceed the Maximum Transmission Unit (MTU) of the network interface. Therefore, header and tag size is only counted once per bundle transmission. Header size includes the IP header when considering protocols that are not integrated into the network stack (e.g., IPsec).

The probability of a packet being delivered is represented by the variable p, which is set to the value of 1 for this investigation, assuming no packet loss in all experiments reported on in this paper. This equation holds true for any non-clustered method of distributing tasks throughout a MANET.

Fig. 10 shows the comparison of SUPERMAN and IPSEC performing CF-CBBA in terms of the number of additional bytes needed to secure data transfer during the DTA process.

IPsec requires 1.8 KB for CF-CBBA communicating a one task problem, compared with 2 KB of data for CBBA. SUPERMAN generates an overhead of 900 bytes for one task CF-CBBA problems. For 50 task problems, SUPERMAN generates security overheads 45 percent the size of IPsec's, while adding only 1.9 percent more data to the bundle exchange process for 50 task CF-CBBA problems. This is driven by the smaller packet size of SUPERMAN.

Equation (2) expands on the previously shown (1), to describe how the security overhead of a given protocol can be derived for CF-CBBA task allocation.

$$y = \left(\sum\nolimits_{1 \leq i \leq L} x(i)\right) + x(p) \tag{2}$$

The total number of bytes, y, is the product of the sum of all cluster allocation (represented as instances of x). The variable p of x represents the cluster head allocation of CF-CBBA, which is performed prior to pushing the resulting task lists to the cluster level for final allocation among cluster members.

For both CBBA and CF-CBBA, SUPERMAN's smaller packet size reduces the security overhead required. It is notable that security overheads are relatively large for smaller task allocation problems, with larger problems becoming more efficient in terms of the data being protected

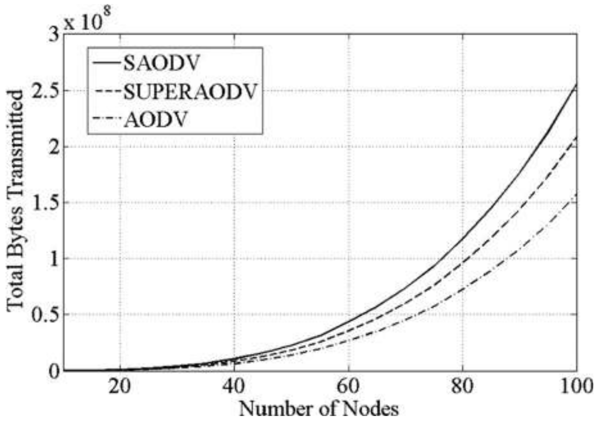


Fig. 11. Chart comparing the number of additional bytes required to secure routing packets using SUPERAODV, SAODV, and AODV.

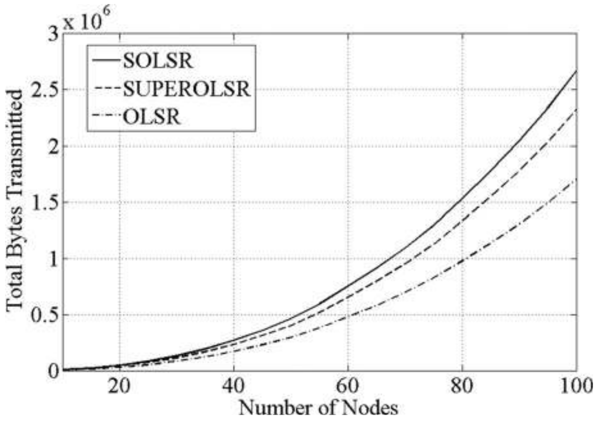


Fig. 12. Chart comparing the number of additional bytes required to secure routing packets using SUPEROLSR, SOLSR, and OLSR.

relative to the data required to provide that protection. This may be mitigated by reducing the size of the tag appended to each packet from 20 bytes to a more manageable size, such as 4 bytes. For this research, the maximum tag size has been chosen to reflect a worst-case scenario and maintain parity with the tag sizes observed for SAODV and SOLSR.

A potential limitation of the lightweight SUPERMAN header is the lack of configuration data. SUPERMAN is not multi-mode, supporting only one mode of security. It is intended as a MANET only security protocol. This means that it lacks the flexibility of VPN protocols, such as IPsec, but provides more efficient, targeted security to MANETs.

4.4 Comparison of Security Overhead in Routing

4.4.1 Method

The additional cost of secure routing is analysed to determine the impact of SUPERMAN on a proactive and reactive MANET protocol. AODV and OLSR, along with their secure implementations, are compared against SUPERMAN secured routing using each protocol. Results have been obtained using a series of MATLAB simulations under the following conditions:

- Simulation parameters outlined in Table 3
- SUPERMAN is applied to OLSR and AODV routing packets
- SOLSR and SAODV are used for comparative analysis
- It is assumed that any pre-routing authentication or first contact handshakes have been performed prior to sending routing packets

The results of these simulations show the number of bytes transmitted during the routing process. Unsecured routing protocols have no security overhead, providing a baseline cost for the routing process. SUPERMAN and secured routing protocols incur this baseline cost, plsu security overhead. The outcome of these simulations will focus on the cost of additional security. Cost of security, in this context, is measured by subtracting the bytes transferred by the secure protocol(s) from the baseline values shown.

4.4.2 Results

Due to the nature of the experiments undertaken in this subsection, a large difference may be perceived between OLSR and AODV. This is due to the experiments focusing on a single instance of routing, in which all nodes in the network form routes with each other.

A single instance of network-wide routing is more demanding for AODV than OLSR (in terms of bytes required to complete the routing operation), but it must be noted that routes will be maintained under AODV until they time out. OLSR, however, will regenerate routes periodically. These results are therefore representative of the total cost for a network wide instance of routing, not the ongoing costs associated with routing on-demand or periodically.

Fig. 11 shows the number of bytes required by AODV, SAODV and SUPERAODV to generate a fully connected set of routes for a network comprised of between 10 and 100 nodes.

AODV provides the cheapest routing with no additional security data or behavioural requirements. In networks of 100 nodes, it requires an average of 73 percent of the bytes required by SUPERMAN protected AODV (SUPERAODV). AODV requires 60.2 percent of the communication required by SAODV.

SUPERAODV does not change the behaviour of AODV, but encapsulates all packets in a SUPERMAN header and tag to provide authentication, confidentiality and integrity to the routing process. SUPERAODV adds a security overhead of 36.9 percent more bytes to AODV, in networks of 100 nodes.

SOADV requires more complex routing behaviour than AODV and SUPERAODV, as well as the addition of header data and a tag to provide security services to the routing process. SAODV generates a security overhead of 66.6 percent more bytes, when compared to AODV in networks of 100 nodes.

Fig. 12 shows the number of bytes required by OLSR, SOLSR and SUPEROLSR to generate a fully connected set of routes for a network comprised of between 10 and 100 nodes.

SUPEROLSR, does not change the behaviour of OLSR but, like SUPEROLSR, it encapsulates routing packets in a secure header and footer (tag). SUPEROLSR requires an additional 40.8 percent of OLSR's byte requirement to provide security to an instance of routing operation performed between 100 nodes.

SOLSR requires 62.3 percent more bytes than OLSR to securely route between 100 nodes. SOLSR requires the

TABLE 4
Security Feature Comparison

Dimensions	Security Protocol			
	SUPERMAN	SOLSR	SAODV	IPsec/MANIPsec
Access Control	X			X
Authentication	X			X
Non-repudiation	X			X
Confidentiality	X			X
Communication Security	X	X	X	X
Data Integrity	X	X	X	X
Availability	X	X	X	
Privacy	X			X

addition of a tag and timestamp to each routing packet, incurring a significant overhead. This does, however, provide critical security services not offered by OLSR, as shown in Table 4 (OLSR provides none of the listed services).

For both AODV and OLSR, SUPERMAN is shown to generate lower overheads by preserving the behaviour of the routing algorithms and providing only the required security features needed to provide authentication, confidentiality and integrity services to the routing process. Mode selection variables and multiple digital signatures are avoided. To provide integrity and authentication services, SUPERMAN only requires a HMAC tag and SUPERMAN header.

The relatively low-cost of SUPERMAN can be ascribed to its use of a closed-network philosophy. By harnessing the control that the owner of a MANET has over the nodes, and the dual end-point/router nature of each node, it is possible to protect routing and application data using a network-stack integrated solution.

SAODV and SOLSR assume a potentially hostile network environment, due to the persistent open-medium problem they are assumed to have to deal with. By closing the network, SUPERMAN can reduce the cost of security by enforcing trustworthiness within the network.

4.5 Security Feature Comparison

SUPERMAN offers a full suite of security services, addressing all eight of the security dimensions outlined in the ITU Rec X.805 document. Table 4 compares the security services of SUPERMAN with SAODV, SOLSR and IPsec. This comparison provides context for the costs seen in the previous results, showing the services provided in return for the additional communication overheads incurred when using SUPERMAN, IPsec or secure routing protocols in a MANET.

IPsec extends seven of eight security services. It does not provide node checking availability services to determine the status of routes and current online members of a network. IPsec does not generally provide route monitoring or point-to-point security service, instead being primarily focused on end-to-end security.

Virtual private Network (VPN) protocols such as IPsec are designed to be adaptable to a variety of networks. They consider the medium itself to be unreliable, and thus focus on the protection of data transmitted over the network, rather than the protection of topology generation and maintenance traffic. This internet-centric design becomes apparent when applied to MANETs, where the vulnerability of the routing protocol can remain a significant threat even when communication security assplied to application data is being provided by a VPN protocol.

SAODV and SOLSR are designed to secure the routes between nodes, providing protection for end-to-end and point-to-point communication for topology regeneration and route finding only. Data sent along such routes is not secured. The integrity of the route can be enforced, but confidentiality of data packets sent along the route is not.

SUPERMAN provides all eight security services. It is integrated at the network layer, providing lightweight security by avoiding the re-encapsulation process required by IPsec. It protects routing packets, as all packets passing through layer 3 of the network stack are protected. In this way, SUPERMAN provides protection for all data, safeguarding the network and data communicated over it.

In addition to protecting data end-to-end (like IPsec), protection is extended point-to-point, to ensure that the route between source and destination can be trusted. This is achievable due to the small size and direct ownership of MANETs compared to the scale of the Internet, which IPsec is designed to operate on.

MANETs could have thousands of nodes, but they will likely be owned by a single authority. Internet-like networks lack this concept of sole-ownership, making it difficult to implement integrated security solutions. This difficulty when attempting to implement an all-encompassing security solution encourages the use of IPsec and other network-agnostic VPN protocols).

By focusing specifically on securing communication in the context of MANETs, SUPERMAN avoids some of the higher costs associated with VPN approaches which target Internet-like networks. It protects all communication in the network, including routing traffic, protecting against man-in-the-middle attacks. It compares favourably with IPsec and secure routing protocols in terms of security overheads, due to its integration into layer 3 of the network stack.

5 Conclusion

SUPERMAN is a novel security framework that protects the network and communication in MANETs. The primary focus is to secure access to a virtual closed network (VCN) that allows expedient, reliable communication with confidentiality, integrity and authenticity services.

SUPERMAN addresses all eight security dimensions outlined in X.805. Thus, SUPERMAN can be said to implement a full suite of security services for autonomous MANETs. It fulfils more of the core services outlined in X.805 than IPsec, due to being network focused instead of end-to-end oriented.

IPsec is intended to provide a secure environment between two end-points regardless of route, and has been suggested by some researchers to be a viable candidate for MANET security. However, it does not extend protection to routing services. Nor does it provide low-cost security, requiring a lengthy set-up and teardown process, usually on a session basis.

Simulation has been undertaken and the results are reported and analysed to determine the relative cost of security for SUPERMAN, compared against IPsec, SAODV and SOLSR where relevant.

SUPERMAN provides a VCN, in which the foundation-block of security is provided by authenticating nodes with the network. This enables further benefits, such as the security association referral and network merging. It also provides a relatively light-weight encapsulation packet and variable length tag.

Under both CBBA and CF-CBBA, the security overheads of SUPERMAN have been demonstrated to be lower than those of IPsec. Both DTA algorithms represent how a MANET can be made autonomous, by allowing problem solving without human intervention to occur on the network. Securing the communication required to facilitate this functionality is a critical consideration when providing a fully secured network. By providing lower cost security than existing alternatives, while providing security across all eight security dimensions, SUPERMAN proves it is a viable and competitive approach to securing the communication required by autonomous MANETs.

SUPERMAN has been shown to provide lower-cost security than SAODV and SOLSR for their respective routing protocols. By establishing a secure, closed network; one can assume a certain level of trust within that network. This reduces the need for costly secure routing behaviours designed to mitigate the effects of an untrusted environment (and untrusted nodes) on the routing process. By preventing the entry of potentially untrustworthy nodes to the network, and thus the routing process, a MANET may be protected from subversion of its routing services at a lower cost, as malicious nodes are barred from the process entirely.

SUPERMAN provides security to all data communicated over a MANET. It specifically targets the attributes of MANETs, it is not suitable for use in other types of network at this time. It sacrifices adaptability to a range of networks, to ensure that MANET communication is protected completely and efficiently. A single efficient method protects routing and application data, ensuring that the MANET provides reliable, confidential and trustworthy communication to all legitimate nodes.

Future work includes the implementation of SUPERMAN [32] on a simple mobile node platform to allow experimental observation and profiling of its performance, the proposal of network bridging solutions capable of providing SUPERMAN services between two closed networks over an insecure intermediate network, and investigating the effects of variable network topology on SUPERMAN to better understand the role of the credential referral mechanism on overhead mitigation in SUPERMAN networks.

References

[1] P. S. Kiran, "Protocol architecture for mobile ad hoc networks," in *Proc. IEEE Int. Ad. Comput. Conf.*, 2009, pp. 2112–2117.

[2] A. Chandra, "Ontology for manet security threats," in *Proc. 2nd Nat. Conf. Netw. Eng.*, 2005, pp. 171–117.

[3] A. K. Rai, R. R. Tewari, and S. K. Upadhyay, "Different types of attacks on integrated manet-internet communication," *Int. J. Comput. Sci. Secur.*, vol. 4, no. 3, pp. 265–274, 2010.

[4] D. Smith, J. Wetherall, S. Woodhead, and A. Adekunle, "A cluster-based approach to consensus based distributed task allocation," in *Proc. 22nd Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process.*, 2014, pp. 428–431.

[5] I. D. Chakeres and E. M. Belding-Royer, "Aodv routing protocol implementation design," in *Proc. 24th Int. Conf. Distrib. Comput. Syst. Workshops*, 2004, pp. 698–703.

[6] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626, Oct. 2003, Doi: 10.17487/RFC3626.

[7] M. Hyland, B. E. Mullins, R. O. Baldwin, and M. A. Temple, "Simulation-based performance evaluation of mobile ad hoc routing protocols in a swarm of unmanned aerial vehicles," in *Proc. 21st Int. Conf. Adv. Inf. Netw. Appl. Workshops*, 2007, vol. 2, pp. 249–256.

[8] J. Pojda, A. Wolff, M. Sbeiti, and C. Wietfeld, "Performance analysis of mesh routing protocols for uav swarming applications," in *Proc. 8th Int. Symp. Wireless Commun. Syst.*, 2011, pp. 317–321.

[9] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: Challenges and solutions," *IEEE Wireless Commun.*, vol. 11, no. 1, pp. 38–47, Feb. 2004.

[10] N. Garg and R. Mahapatra, "Manet security issues," *Int. J. Comput. Sci. Netw. Secur.*, vol. 9, no. 8, pp. 241–246, 2009.

[11] W. Ivancic, D. Stewart, D. Sullivan, and P. Finch, "An evaluation of protocols for uav science applications," NASA Technical Reports (NTRS), 2011 Earth Science Technology Forum (ESTF2011, Jun. 2011.

[12] A. R. McGee, U. Chandrashekhar, and S. H. Richman, "Using itu-t x. 805 for comprehensive network security assessment and planning," in *Proc. 11th Int. Telecommun. Netw. Strategy Planning Symp.*, 2004, pp. 273–278.

[13] M. G. Zapata, "Secure ad hoc on-demand distance vector routing," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 6, no. 3, pp. 106–107, 2002.

[14] F. Hong, L. Hong, and C. Fu, "Secure OLSR," in *Proc. 19th Int. Conf. Adv. Inf. Netw. Appl.*, 2005, vol. 1, pp. 713–718.

[15] A. Hafslund, A. Tønnesen, R. B. Rotvik, J. Andersson, and Ø. Kure, "Secure extension to the OLSR protocol," presented at the OLSR Interop Workshop, San Diego, CA, USA, 2004.

[16] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "Dos attacks in mobile ad hoc networks: A survey," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Technol.*, 2012, pp. 535–541.

[17] S. Maity and S. K. Ghosh, "Enforcement of access control policy for mobile ad hoc networks," in *Proc. 5th Int. Conf. Secur. Inf. Netw.*, 2012, pp. 47–52.

[18] D. Hurley-Smith, J. Wetherall, and A. Adekunle, "Virtual closed networks: A secure approach to autonomous mobile ad hoc networks," in *Proc. 10th Int. Conf. Internet Technol. Secured Trans.*, 2015, pp. 391–398.

[19] S. Bhattacharya and T. Basar, "Game-theoretic analysis of an aerial jamming attack on a uav communication network," in *Proc. Amer. Control Conf.*, 2010, pp. 818–823.

[20] S. Lu, L. Li, K.-Y. Lam, and L. Jia, "Saodv: A manet routing protocol that can withstand black hole attack," in *Proc. Int. Conf. Comput. Intell. Secur.*, 2009, vol. 2, pp. 421–425.

[21] S. Zhao, R. Kent, and A. Aggarwal, "A key management and secure routing integrated framework for mobile ad-hoc networks," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1046–1061, 2013.

[22] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X. 509 internet public key infrastructure online certificate status protocol-OCSP," *RFC 2560*, Jun. 1999, Doi: 10.17487/RFC2560.

[23] N. Doraswamy and D. Harkins, *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks*. Upper Saddle River, NJ, USA: Prentice Hall Professional, 2003.

[24] A. Ghosh, R. Talpade, M. Elaoud, and M. Bereschinsky, "Securing ad-hoc networks using IPSEC," in *Proc. IEEE Mil. Commun. Conf.*, 2005, pp. 2948–2953.

[25] K. N. Ali, M. Basheeruddin, S. K. Moinuddin, and R. Lakkars, "Manipsec-ipsec in mobile ad-hoc networks," in *3rd IEEE Int. Conf. Comput. Sci. Inf. Technol.*, 2010, vol. 1, pp. 635–639.

[26] E. Rescorla, "Diffie-hellman key agreement method," *RFC 2631*, Jun. 1999, Doi: 10.17487/RFC2631.

[27] L. Harn, M. Mehta, and W.-J. Hsin, "Integrating diffie-hellman key exchange into the digital signature algorithm (DSA)," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 198–200, Mar. 2004.

[28] H. Krawczyk and P. Eronen, "Hmac-based extract-and-expand key derivation function (HKDF)," *RFC 5869*, May 2010, Doi: 10.17487/RFC5869.

[29] A. Adekunle and S. Woodhead, “An aead cryptographic framework and tinyaead construct for secure wsn communication,” in *Proc. Wireless Adv.*, 2012, pp. 1–5.

[30] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[31] M. Matsumoto and T. Nishimura, “Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator,” *ACM Trans. Modeling Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.

[32] An open-source implementation of SUPERMAN is in development consisting of a Linux Kernel Module and Daemon. (2016). [Online]. Available: https://bitbucket.org/wj88/superman/

Darren Hurley-Smith received the BEng (Hons) degree in computer systems and software engineering from the University of Greenwich, in 2012, and the PhD degree from the University of Greenwich, in 2015. Currently, he is a post-doctoral research associate with the University of Kent’s School of Computing.

Jodie Wetherall received the BEng (Hons) degree in 2001, and the PhD degree in 2010. He is currently a principal lecturer in the Faculty of Engineering and Science, University of Greenwich. His research interests include MANETs, security, scheduling, and automation.

Andrew Adekunle received the BENg (Hons.) and PhD degrees in network security. He is currently a lecturer in the Faculty of Engineering and Science, University of Greenwich. His research interests include network security and embedded networked systems.